Universität Rostock Traditio et Innovatio

Modellbasierte, sichere Service- und Netzwerkkonfiguration in der Gebäudeautomation

Der Fakultät für Elektrotechnik und Informatik der Universität Rostock
zur Erlangung des akademischen Grades eines

Doktor-Ingenieur (Dr.-Ing.)

vorgelegte Dissertation von
Arne Wall

Tag der Einreichung: 3.5.2023
Tag der Verteidigung: 19.10.2023

Gutachter:
Prof. Dr.-Ing. Dirk Timmermann, Universität Rostock, Institut für Angewandte Mikroelektronik und Datentechnik
Prof. Dr.-Ing. Christian Haubelt, Universität Rostock, Institut für Angewandte Mikroelektronik und Datentechnik
Univ. Prof. Dipl.-Ing. Dr. techn. Wolfgang Kastner, Technische Universität Wien, Institut für Computer Engineering

https://doi.org/10.18453/rosdok_id00004483

Dissertation
Fakultät für Informatik und Elektrotechnik

Arne Wall
Institut für Angewandte Mikroelektronik und Datentechnik

Danksagung

Während meiner Doktorandenzeit am Institut für Angewandte Mikroelektronik und Datentechnik durfte ich wertvolle Erfahrungen sammeln. Ich möchte mich hiermit bei den Professoren und Kollegen für die tolle und prägende Zeit bedanken. Besonders dankbar bin ich für die vielen anregenden Diskussionen. Außerdem konnte ich auf meinen Dienstreisen viele spannende Menschen und Projekte kennenlernen und mich weiter entwickeln.

Ich blicke zudem auf sechs Jahre an der Universität Rostock zurück, in denen ich tiefe Freundschaften geschlossen habe.

Ich möchte mich außerdem bei meiner Familie und meinen Freunden bedanken, die mir während dieser Zeit den Rücken gestärkt haben.

Vielen Dank für diese tolle Zeit!

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die von mir vorgelegte Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen, Internet-Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit, einschließlich Tabellen und Abbildungen, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Rostock, den 24. April 2023

_____

Arne Wall

Zusammenfassung

Die moderne Maschine-zu-Maschine-Kommunikation (M2M) wird zunehmend durch offene Standards und Webtechnologien geprägt. Dadurch ist es möglich, Geräte verschiedener Hersteller miteinander IP-basiert zu vernetzen und an das Internet anzubinden. Dabei verschmelzen das Web und die lokalen Netzwerke der Gebäudeautomation (GA).

Dieselben Angriffstechniken auf Server und Clients lassen sich auf eingebettete Systeme übertragen. Während Webserver und Endgeräte durch regelmäßige Sicherheitsupdates gegen Angriffe gehärtet werden, bleiben eingebettete Systeme oftmals auf der Strecke. Falls ein Endgerät kompromittiert sein sollte, um weitergehende Angriffe innerhalb eines Netzwerks auszuführen, lassen sich die Folgen durch eine Abschottung der Geräte voneinander drastisch reduzieren. So kommunizieren z.B. in Unternehmensnetzwerken die einzelnen Teilnehmer gemäß einer firmeninternen Sicherheitsrichtlinie miteinander.

Solche Mechanismen zur Absicherung der Gerätekommunikation existieren auf Systemebene einer GA bislang nicht. Daher wurde in dieser Arbeit eine Sicherheitsarchitektur entwickelt, die speziell an die Anforderungen einer modernen M2M-Kommunikation in einem GA-System zugeschnitten ist. Es wird ein Konzept vorgestellt, wie die einzelnen Endgeräte einer GA über ein sicheres Verfahren mit Zugangs- und Konfigurationsdaten versorgt werden und über abgeschottete Domänen miteinander kommunizieren. Dabei wird die Implementierung der Endgeräte zur Designzeit beschrieben. Des Weiteren wird ein Protokollstack vorgeschlagen und experimentell untersucht, um verschiedene Anwendungsanforderungen innerhalb der GA zu erfüllen. Sämtliche Implementierungsvorschläge für Gerätehersteller werden auf Basis von offenen Standards und Protokollen gemacht. Offene Standards sind essentiell, damit eine herstellerübergreifende Gerätekommunikation gelingen kann. Angewandte Security-Verfahren und Protokolle, die offengelegt sind, bieten den besten Schutz gegenüber Angriffen, da ihre Sicherheitseigenschaften stetig durch die Forschungsgemeinschaft untersucht werden. Der gesamte Lebenszyklus eines Gerätes, von seiner Entwicklung bis hin zur Demontage, wird durch offene Gebäudeinformationsmodelle begleitet. Ein Digital-Twin, der die gesamte GA von der Kommissionierung bis hin zu Umbaumaßnahmen modelliert, dient als Grundlage für eine automatisierte Berechnung von Konfigurationsdaten. Es gilt, den Einfluss-

faktor ”Mensch” als potentielle Security-Schwachstelle durch automatisierte Abläufe zu unterstützen, jedoch zu jeder Zeit einen manuellen Eingriff zuzulassen.

Abstract

Modern machine-to-machine communication is increasingly characterized by open standards and web technologies. This makes it possible to network devices from different manufacturers with each other on an IP basis and to connect them to the Internet. In the process, the web and local networks as used in building automation (BA) are merging. The same attack techniques on servers and clients can be applied to embedded systems. While web servers and endpoints running client applications are hardened against attacks through regular security updates, embedded systems often are not provided with urgent software fixes. If an end device is compromised to carry out more extensive attacks within a network, the consequences can be drastically reduced by separation of the devices from one another. In corporate networks, for example, the participants communicate with each other in accordance with a company-internal security guideline. Such mechanisms for securing device communication do not yet exist at the system level of a BA. Therefore, a security architecture has been developed that is specifically tailored to the requirements of modern machine-to-machine communication in a BA system. A concept is presented how each individual end device of a BA is supplied with access and configuration data via a secure procedure and communicates with other EDs within partitioned domains. The implementation of the end devices at design time is described. Furthermore, implementation proposals based on open standards and protocols are made for device manufacturers. Open standards are essential for cross-vendor device communication to succeed. Applied security procedures and protocols, that are open, offer the best protection against attacks, as their security properties are constantly being investigated by the research community. The entire life cycle of a device up to disassembly is accompanied by building information modeling. A digital twin, which models the entire BA from commissioning phase until building conversions, serves as the basis for an automated calculation of configuration data. The aim is to support the human factor as a potential security vulnerability through automated processes, but to allow manual intervention at any time.

Inhaltsverzeichnis

XIV Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ACK	Acknowledgement
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
ALM	Airtime Link Metric
ANTs	Application-Driven Network Trust Zones
AODV	Ad-hoc On-demand Distance Vector
BACnet	Building Automation and Control Networks
BAS	Building Automation System
BSI	Bundesamt für Sicherheit in der Informationstechnik
CBC	Cipher Block Chaining
CBOR	Concise Binary Object Representation
CCM	Counter with Cipher Block Chaining Message Authentication Code
CoAP	Constrained Application Protocol
CRUD	Create Read Update Delete
CSRF	Cross Site Request Forgery
DASCo	Dynamic Adaptive Streaming over CoAP
DDR3	Double Data Rate 3
DHCP	Dynamic Host Configuration Protocol
DPWS	Devices Profile for Web Services
DTLS	Datagram Transport Layer Security
ED	End Device
FEC	Forward Error Correction
GA	Gebäudeautomation
GC	Garbage Collector
GCM	Galois/Counter Mode
GLT	Gebäudeleittechnik
GUI	Graphical User Interface

HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HWMP	Hybrid Wireless Mesh Protocol
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ITTT	If-this-then-that
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
LDAP	Leightweight Directory Access Protocol
LWM2M	Leightweight Machine-to-Machine
M2M	Machine-to-Machine
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
MJPEG	Motion Joint Photographic Experts Group
mPCIe	mini Peripheral Component Interconnect express
MPEG	Moving Picture Experts Group
MQTT	Message Queuing Telemetry Transport
MTU	Maximum Transmission Unit
NAC	Network Access Control
NFC	Near Field Communication
NIC	Network Interface Card

OBDD Ordered Binary Decision Diagram
OSCORE Object Security for Constrained Representational State Transfer Environments
OWASP Open Web Application Security Project

PSK Pre-shared Key

QAM Quadrature Amplitude Modulation
QoS Quality-of-Service

RAM Random Access Memory
RESTful Representational State Transfer
RFC Request for Comments
RP-SMA Reverse Polarity Sub-Miniature-A
RSA Rivest–Shamir–Adleman
RSS Rich Site Summary
RTT Round-Trip-Time

SAE Simultaneous Authentication of Equals
SAT Satisfiability
SC Security Controller
SDN Software-defined Networking
SNMP Simple Network Management Protocol
SPS Speicherprogrammierbare Steuerung
SQL Structured Query Language
SSID Service Set Identifier

TCP Transmission Control Protocol
TLS Transport Layer Security

UDP User Datagram Protocol
URI Uniform Resource Identifier
URL Uniform Resource Locator

VLAN Virtual Local Area Network

VPN Virtual Private Network

WLAN Wireless Local Area Network

XSS Cross Site Scripting

1 Einleitung

Die Gebäudeautomation (GA) ist aus modernen Gebäuden kaum wegzudenken. Sie schafft Komfort durch klimatisierte Räume und intelligente Lichtsteuerung , die sich dynamisch an die Umgebungsbedingungen anpasst [A 1], [A 2], [A 3]. Betreiber können durch eine optimal eingestellte GA Energie sparen und die einzelnen Gebäude warten [A 4]. Bislang war die Planung einer GA aufwendig, da für jedes einzelne Gerät Stromleitungen und Steuerleitungen eingeplant und verlegt werden mussten. Der entstandene Kabelbaum ist teuer und schlecht wartbar. Es ist nicht ohne Weiteres möglich, Änderungen an bestehenden Installationen vorzunehmen. Daher haben im Laufe der Entwicklung zunehmend Webtechnologien für eingebettete Systeme an Bedeutung gewonnen. Sie ermöglichen es, IP-basierte Nachrichten zwischen Geräten auszutauschen, um Sensordaten und Steuerbefehle zu übermitteln.

Die IT-Sicherheit hat in den vergangenen Jahren zunehmend an Bedeutung gewonnen. Durch den stetig steigenden Grad der Gerätevernetzung eröffnen sich für Angreifer neue Angriffsmöglichkeiten, um unbefugt IT-Systeme zu manipulieren, sensible Daten auszulesen und wirtschaftlichen Schaden anzurichten [A 5], [A 6], [A 7]. Der Trend, eingebettete Systeme mit Web-Technologien zu vernetzen, ist bereits seit einigen Jahren Gegenstand von internationalen Forschungsarbeiten [A 8], [A 9]. Dabei wurden bestehende Web-Technologien auf ihre Eignung für den Einsatz auf eingebetteten Systemen untersucht und durch neue Protokolle weiterentwickelt. Geräte, die IP-basiert kommunizieren, sind in der Industrie in Form von Produkten im Consumer-Bereich und in professionellen Anwendungen omnipräsent. Haushaltsgeräte, Lampen, Türschlossanlagen und Heizungsanlagen lassen sich mit Cloud-Anwendungen koppeln und durch andere Geräte steuern. Dieselbe Entwicklung, wenn auch etwas langsamer, findet ebenso in der Gebäudeautomatisierung statt. Die besonderen Anforderungen an die Geräteinteroperabilität und an eine anwenderfreundliche Installation können durch Web-Technologien erfüllt werden [A 10]. Die steigende Anzahl an Geräten in einem Netzwerk erhöht jedoch die Gefahr, dass sich mindestens ein verwundbares Gerät im System befindet. Angreifer, die bislang Angriffe auf Web-Dienste des Internets ausführen, sind in der Lage, ihr Repertoire an Angriffstechniken auf eingebettete Systeme anzupassen und auszuführen. Ein einzelnes Gerät, das sich in einem Gebäudeautomationsnetzwerk befindet,

kann missbraucht werden, um den Angriff auf weitere Geräte auszuweiten. In dieser Arbeit wird eine allgemeingültige Sicherheitsarchitektur für Gebäudeautomationssysteme vorgestellt, um dieser Bedrohung zu begegnen. Dazu wird der Stand der Technik zu bestehenden M2M-Protokollen im Gebäudeautomationsumfeld analysiert und hinsichtlich der Sicherheitseigenschaften bewertet. Ziel ist es, ein Gesamtsystem zu erstellen, das aus möglichst wenigen Teilkomponenten besteht, um die Sicherheit zu verbessern. Das Sicherheitskonzept berücksichtigt alle Phasen des Produktlebenszyklus. Damit gehen Implementierungsvorschläge für Geräteentwickler einher. Während der Inbetriebnahme und während der Laufzeit der Geräte wird besonderer Fokus auf eine gute Anwendbarkeit durch Techniker und Nutzer gelegt. Dabei werden sicherheitskritische Gerätekonfigurationen durch Modelle beschrieben und Kommissionierungsaufgaben mithilfe von Algorithmen automatisiert, so dass die ”Fehlerquelle Mensch” minimiert wird. Eine zentrale Instanz soll ähnlich wie ein DHCP-Server (Dynamic Host Configuration Protocol) [A 11] jedes einzelne Gerät mit individuell berechneten Konfigurationsdaten versorgen. Trotz der Automatisierung soll es dem Anwender jedoch möglich sein, in das System einzugreifen und Änderungen vorzunehmen. Ziel der Arbeit ist es, ein Gebäudeautomationssystem ohne Verlust von Funktionalitäten zu partitionieren, um die einzelnen Geräte in abgeschotteten Trust Zones kommunizieren zu lassen. Somit soll Angriffen vorgebeugt bzw. sollen Angriffe zur Laufzeit eingedämmt werden. Weiterhin soll ein sicheres Verlassen von Geräten aus einem Netzwerk gewährleistet sein.

1.1 Problemstellung und Zielsetzungen der Arbeit

Während bislang Server und Rechner des Internets von Security-Risiken bedroht worden sind, geraten zunehmend eingebettete Systeme in den Fokus von Angreifern. Durch ein physisch weit ausgedehntes LAN mit unzähligen Geräten vergrößert sich die Angriffsfläche immens. Es ist eine Frage der Zeit, bis ein Gerät des GA-Netzes von einem Angreifer kontrolliert wird, um den Einflussbereich auf weitere Geräte auszudehnen und weiteren Schaden im Netzwerk anzurichten. Begünstigt wird ein solches Vorgehen durch die Tatsache, dass Geräte über IP-basierte Netze miteinander verbunden und verschiedene Protokolle der Anwendungsschicht durch Technologie-Gateways überbrückt werden [A 10], [A 12], [A 13], [A 14], [A 15]. Die Vollvernetzung steigert die unnötige Gefahr, dass Geräte, die auf logischer Ebene nicht miteinander kommunizieren müssen, durch Angreifer missbraucht werden.

Ziel dieser Arbeit ist es, eine Sicherheitsarchitektur zu entwickeln, die speziell auf die Bedürfnisse eingebetteter Systeme in GA-Netzwerken zugeschnitten ist. Dabei werden u.a. das Gebäude samt aller Geräte und Anwendungen modelliert, um basierend auf diesen Informationen automatisiert Geräte in abgeschottete Netzwerkbereiche einzuordnen. Dadurch lässt sich die Angriffsfläche massiv verringern. Das Sicherheitskonzept erstreckt sich über alle Phasen des Produktlebenszyklus. Es werden Anforderungen an Endgeräte hinsichtlich der implementierten Protokolle und offenen Schnittstellen abgeleitet, sodass eine Integration in die GA möglich ist. Auf der Grundlage von anerkannten kryptographischen Verfahren und Protokollen wird ein sicherer Kommissionierungsprozess vorgestellt. Die ”Fehlerquelle Mensch“ soll bei der Planung des Netzwerks und der Konfiguration der einzelnen Geräte minimiert werden. Dazu werden eigens entwickelte Algorithmen auf einer vertrauenswürdigen Hardwareplattform ausgeführt. Das Sicherheitskonzept ist in der Lage, dynamisch zur Laufzeit auf sicherheitsrelevante Ereignisse zu reagieren. Schließlich werden Sicherheitsrisiken beim Umbau eines Gebäudes und bei der Dekommissionierung von Geräten berücksichtigt.

1.2 Aufbau der Arbeit

In Kapitel 2 werden die Grundbegriffe und -konzepte der IT-Sicherheit erläutert. Des Weiteren wird die Struktur einer klassischen GA beschrieben und erläutert. Neben den typischen GA-Potokollen werden in Kapitel 3 moderne Verfahren der M2M-Kommunikation vorgestellt und ihre Eignung für die GA bewertet. Die Performance-Eigenschaften des ausgearbeiteten Protokollstacks werden anhand einer Streaming-Anwendung in Kapitel 4 experimentell evaluiert. In Kapitel 5 erfolgt eine Bedrohungs- und Anforderungsanalyse. Die entwickelte BIM-basierte Sicherheitsarchitektur wird in Kapitel 6 vorgestellt. Die technische Realisierung mit Endgeräten, die den in Kapitel 3 gefundenen Protokollstack nutzen, wird in Kapitel 7 behandelt. Das Kapitel 8 fasst alle Ergebnisse dieser Arbeit zusammen. In Abbildung 1.1 werden die eigenen wissenschaftlichen Beiträge den Ebenen der entworfenen Security Architektur zugeordnet.

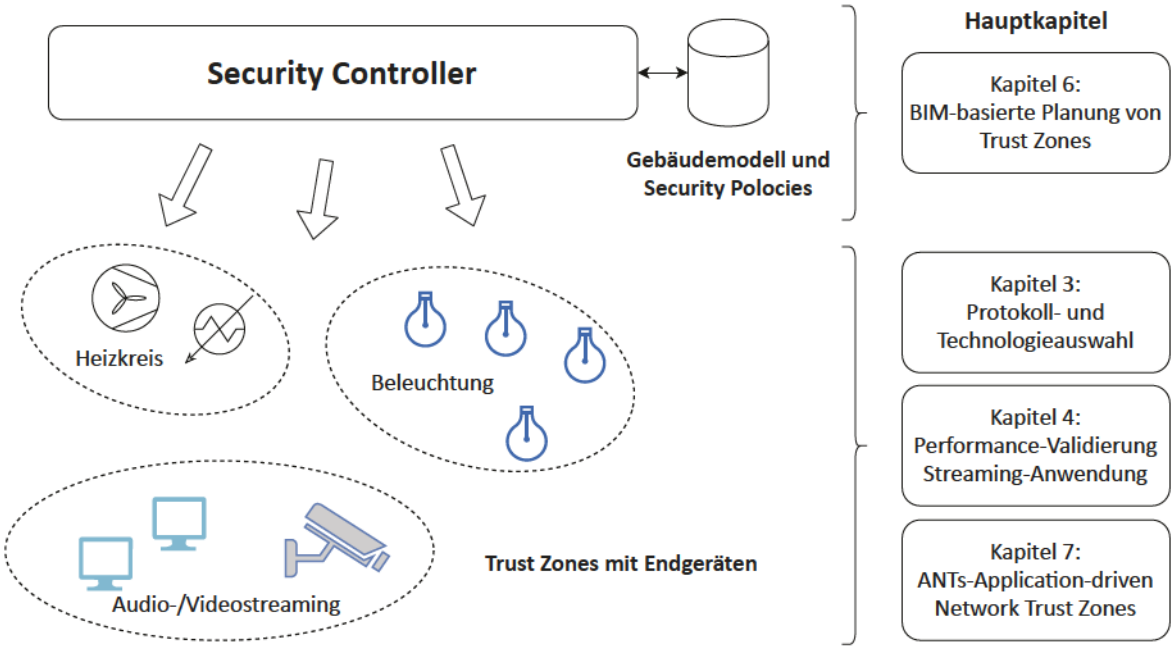

Abbildung 1.1: Übersicht der Ebenen der Security Architektur und Zuordnung der Hauptkapitel

2 Grundlagen

Das Grundlagenkapitel umfasst alle relevanten Begriffe, Standards und Protokolle nach [A 16] und [A 17]. Es werden die Grundbegriffe und Verfahren der IT-Security erläutert. Die grundlegenden Konzepte der IT-Security werden durch verschiedene Kommunikationsprotokolle (Kapitel 3) angewendet. Anschließend wird die allgemeine Struktur eines (Gebäude-)Automationssystems vorgestellt.

2.1 Grundbegriffe und Verfahren der IT-Security

Der englische Begriff “Security“ umfasst drei Haupt-Schutzziele: Vertraulichkeit, Authentizität und Integrität. In dieser Arbeit wird der deutsche Begriff ”Sicherheit” als Übersetzung verwendet. Hierbei sei angemerkt, dass der Begriff ”Sicherheit” nicht mit dem englischen Wort ”Safety” gleichzusetzen ist. ”Safety” beschreibt vielmehr die Ausfallsicherheit bzw. Betriebssicherheit eines Systems.

2.1.1 Vertraulichkeit

Vertraulichkeit ist die Eigenschaft, dass Informationen, die zwischen zwei Entitäten (Alice und Bob) ausgetauscht werden, nicht durch Dritte (Mallory) gelesen werden können. Die Vertraulichkeit einer Nachricht lässt sich durch die Verwendung eines (symmetrischen oder asymmetrischen) Verschlüsselungsverfahrens realisieren.

2.1.2 Authentizität

Eine Nachricht gilt als authentisch, wenn ihr Absender zweifelsfrei durch den Empfänger validiert werden kann. Erhält Bob vorgeblich eine Nachricht von Alice, ist Bob in der Lage, durch Anwendung von mathematischen Verfahren die Korrektheit des Absenders zu prüfen. Dabei kommen häufig digitale Signaturen zum Einsatz.

2.1.3 Integrität

Die Nachrichtenintegrität stellt sicher, dass eine Nachricht auf dem Übertragungsweg zwischen zwei Kommunikationspartnern nicht durch Dritte verändert werden kann oder dass eine Manipulation der Nachricht auf Empfängerseite detektiert werden kann.

2.1.4 Symmetrische Verschlüsselungsverfahren

Bei einer symmetrischen Verschlüsselung verwenden die Kommunikationspartner denselben Schlüssel, um eine Klartextnachricht durch einen Verschlüsselungsalgorithmus zu ver- und entschlüsseln (Abbildung 2.1).

Dazu muss der gemeinsame Schlüssel zwischen zwei Teilnehmern über einen sicheren Kommunikationsweg übertragen oder über ein Schlüsselaushandlungsverfahren, wie z.B. Diffie-Hellman [A 18], berechnet werden. Der AES-Algorithmus (Advanced Encryption Standard) [A 19] ist einer der bekanntesten Vertreter symmetrischer Verschlüsselungsverfahren. AES bietet verschiedene Betriebsmodi, um Datenblöcke zu verschlüsseln. Die Betriebsmodi CCM (Counter with Cipher Block Chaining Message Authentication Code (CBC-MAC)) und GCM (Galois/Counter Mode) stellen AES-Varianten dar, die eine authentifizierte Verschlüsselung ermöglichen. Dabei werden nach Verschlüsselung sowohl Vertraulichkeit und Integrität als

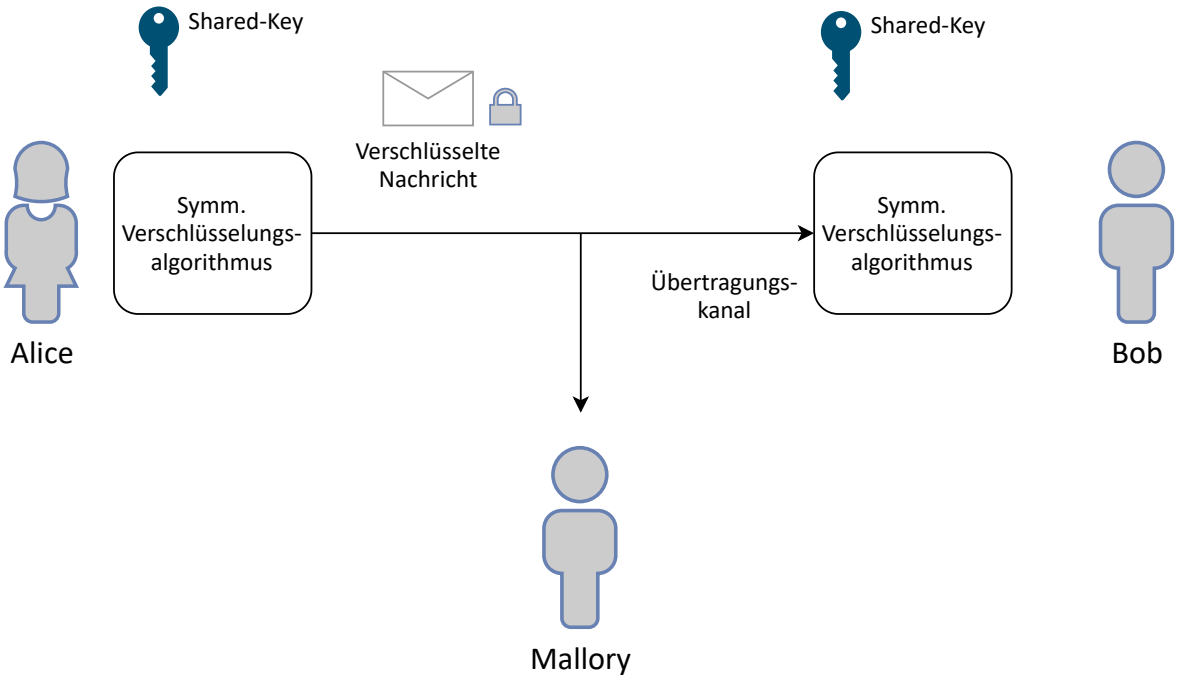

Abbildung 2.1: Symmetrische Verschlüsselung zur Gewährleistung der Nachrichtenvertraulichkeit zwischen Sender (Alice) und Empfänger (Bob) [A 16]

auch Authentizität sichergestellt, ohne dass ein Message Authentication Code gebildet und geprüft werden muss. Die Klasse der Verschlüsselungsverfahren, die alle drei Schutzziele erfüllen, werden als AEAD-Verfahren (Authenticated Encryption with Associated Data) bezeichnet. Die Ende-zu-Ende-Verschlüsselung auf Transportebene des ISO/OSI-Modells nach der aktuellen Version TLS 1.3 [A 20] (Stand 2021) lässt ausschließlich AEAD-Verfahren zu.

2.1.5 Asymmetrische Verschlüsselungsverfahren

Bei asymmetrischen Verschlüsselungsverfahren verfügt jeder Kommunikationsteilnehmer über ein Schlüsselpaar (Abbildung 2.2).

Das Schlüsselpaar besteht aus einem öffentlichen Schlüssel, der für jeden einsehbar ist und aus einem privaten Schlüssel, der nur dem Schlüsselbesitzer bekannt ist. Möchte Alice eine verschlüsselte Nachricht an Bob übermitteln, so verschlüsselt sie die Nachricht mit dem öffentlichen Schlüssel von Bob. Für niemanden außer Bob ist es möglich, ohne Kenntnis über den korrespondierenden privaten Schlüssel

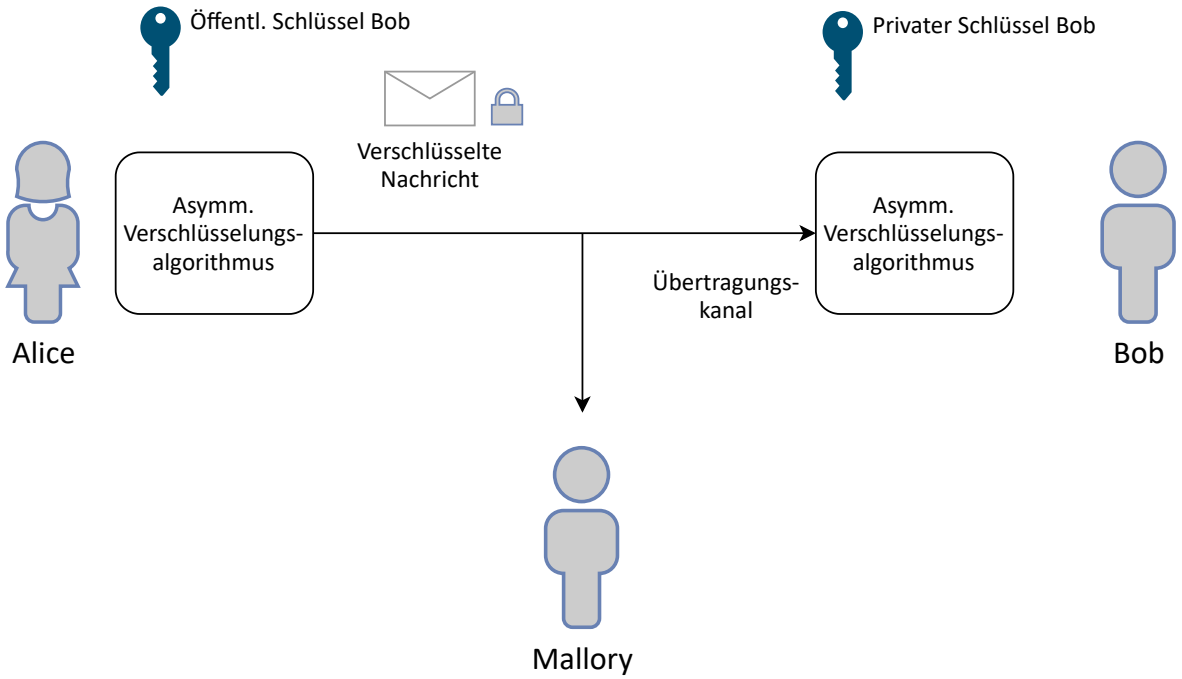

Abbildung 2.2: Asymmetrische Verschlüsselung zur Gewährleistung der Nachrichtenvertraulichkeit zwischen Sender (Alice) und Empfänger (Bob) [A 16]

die Nachricht zu entschlüsseln. Ein anderer Anwendungsfall ist die Generierung von Signaturen (Abbildung 2.3).

Dazu wird der Hashwert einer Nachricht mit dem privaten Schlüssel des Absenders (Alice) verschlüsselt. Der verschlüsselte Hashwert wird als HMAC (Hash Message Authentication Code) oder Signatur [A 21] bezeichnet. Der Empfänger der Nachricht (Bob) dechiffriert die Signatur mit dem öffentlichen Schlüssel des Absenders. Stimmt der entschlüsselte Hashwert mit dem Hashwert der Nachricht überein, gilt die Nachricht als authentisch, da nur der Absender mit der exklusiven Kenntnis über den privaten Schlüssel in der Lage ist, eine gültige Signatur zu berechnen. Weiterhin ist die Nachrichtenintegrität sichergestellt, da es unwahrscheinlich ist, dass eine weitere sinnvolle Nachricht mit demselben Hashwert existiert. Ein bedeutender Vertreter asymmetrischer Verschlüsselungsverfahren ist RSA (Rivest–Shamir–Adleman) [A 22]. Da RSA jedoch nur mit sehr großen Schlüssellängen von 2048 bzw. 4096 Bit sicher verwendet werden kann, wurden asymmetrische Verfahren, die auf elliptischen Kurven [A 23] basieren, entwickelt. Jedoch gelten asymmetrische Verfahren als deutlich weniger performant als symmetrische Verfahren, sodass sie sich nur

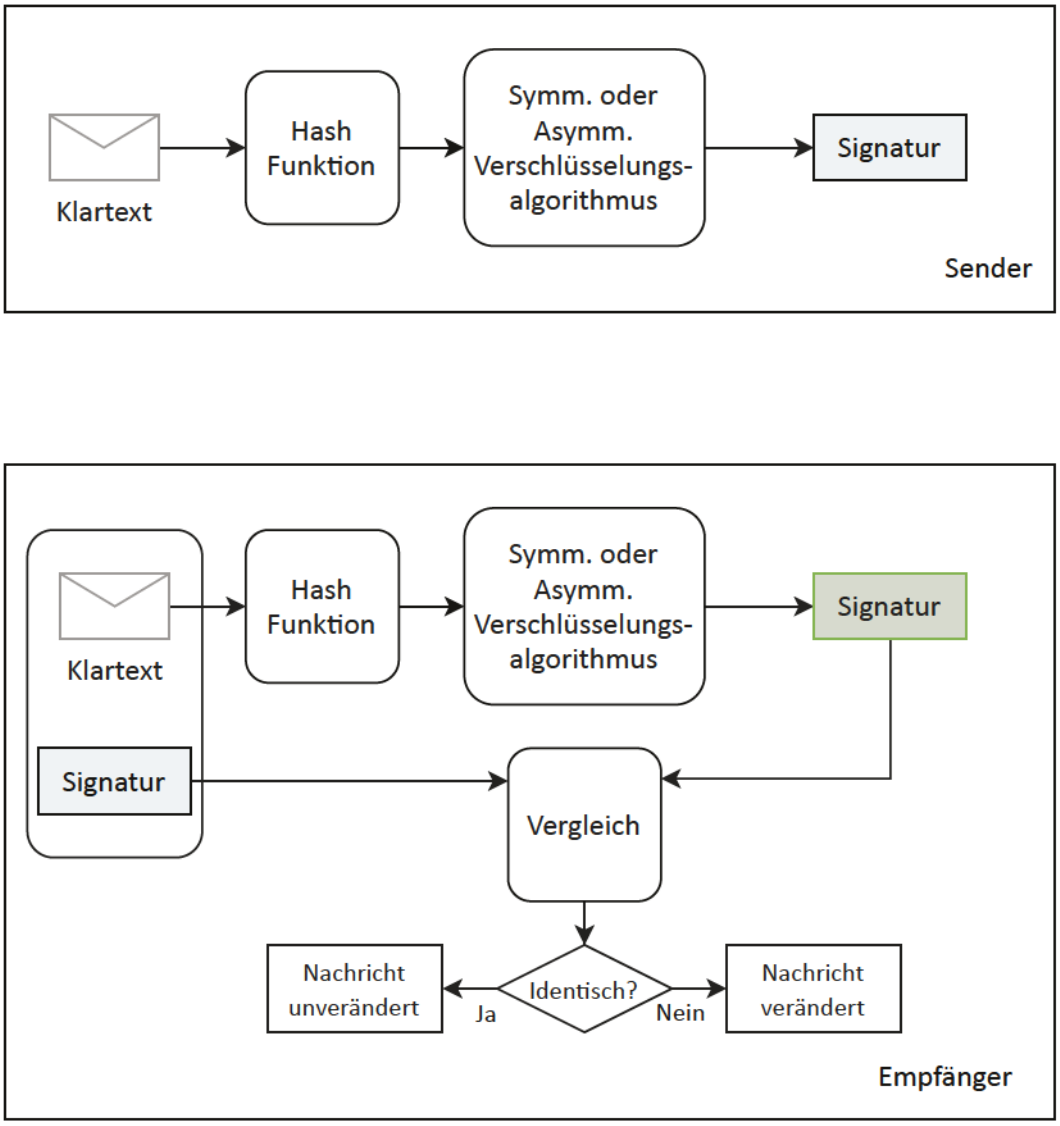

Abbildung 2.3: Generierung und Validierung einer Signatur zur Gewährleistung der Nachrichtenintegrität und -authentizität [A 16]

zur Verschlüsselung bzw. zur Signierung von kleinen Datenmengen eignen. Asymmetrische Verfahren werden häufig zur sicheren (Vertraulichkeit, Authentizität und Integrität) Übermittlung von Schlüsselmaterial, das für symmetrische Verfahren benötigt wird, angewendet.

2.2 Gebäudeautomation

In [A 24] stellen die Autoren die allgemeine Architektur der Automationstechnik dar und geben Beispiele für die konkrete Realisierung in einem Gebäudeautomationssystem. Die klassische Automationstechnik lässt sich abstrakt in drei Ebenen unterteilen: Feld-, Automations- und Bedienebene (Abbildung 2.4).

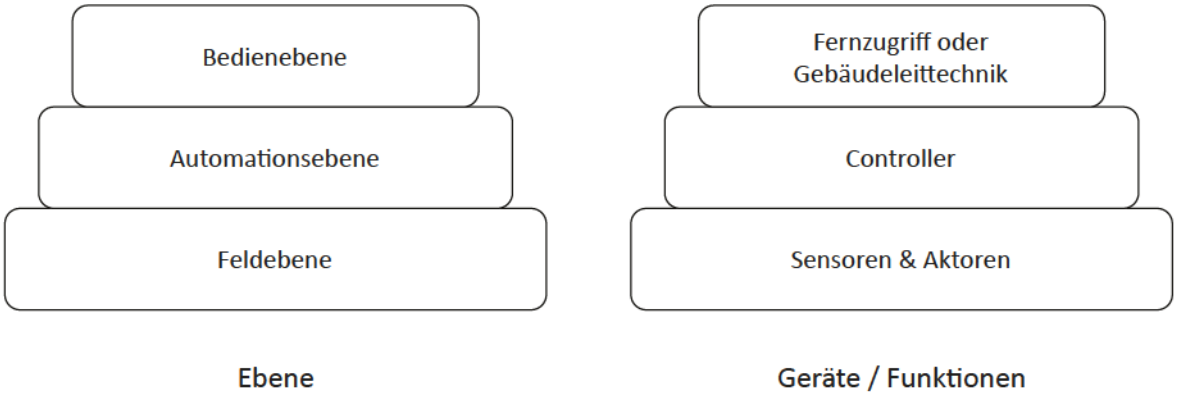

Abbildung 2.4: Automationspyramide [A 24]

Sensoren und Aktoren sind Elemente der Feldebene. Sie sind die leistungsärmste Geräteklasse und stellen die Schnittstelle zur physischen Welt dar. Ein Controller der Automationsebene kommuniziert auf direktem Weg (ohne weitere logische Instanzen) mit den einzelnen Sensoren und Aktoren. Die Sensordaten werden gemäß einer Routine ausgewertet um Aktorik anzusteuern. Somit lassen sich beispielsweise Regelkreise realisieren, um die Temperatur eines Raumes konstant zu halten. Außerdem ist es möglich, durch eine entsprechende Routine das Licht (Aktor) auf einem Flur einzuschalten, wenn der entsprechende Bewegungsmelder (Sensor) ausgelöst hat. Sensordaten können durch den Controller lediglich abgerufen und gespeichert werden. Die nächsthöhere Ebene, die Bedienebene, bündelt die einzelnen Controller der Automationsanlage. Beispielsweise lassen sich Statusinformationen aller Anwendungsbereiche (repräsentiert durch Controller) aufbereiten, um eine Gebäudeleittechnik (GLT) zu implementieren. Die Bedienebene sieht weiterhin die Remote-Bedienung über das Internet vor. Die oberste Schicht der Automationspyramide beinhaltet die leistungsstärksten Geräte des Systems.

In älteren GA-Systemen werden Sensoren und Aktoren über dedizierte Daten- und Stromleitungen angesteuert und mit Energie versorgt. Die Planung einer großen GA-Anlage gestaltet sich als aufwendig, da frühzeitig alle Geräteverknüpfungen zu

einem Controller geplant werden müssen. Dies resultiert in einem großen Kabelbaum, der zum Controller führt. Änderungen zur Laufzeit sind nicht ohne Weiteres möglich. Da das System häufig aus nur einem oder wenigen Controllern besteht, führt ein Ausfall zu einer vollständigen oder weitreichenden Beeinträchtigung der GA-Funktionen.

2.3 Modellierung von Gebäuden mit openBIM

Building Information Modeling beschreibt die moderne Methodik, die einzelnen Gewerke der Bauindustrie miteinander zu verbinden. Dabei begleiten Gebäudemodelle den gesamten Gebäudelebenszyklus. Die Digitalisierung des Baugewerbes zielt darauf ab, Gebäudedaten phasenübergreifend zwischen den einzelnen Gewerken auszutauschen. Dadurch lassen sich Abläufe automatisieren. Die Basis dafür sind Modelle, die die Gebäudestruktur samt funktionaler und physikalischer Eigenschaften umfassen. OpenBIM beschreibt die Verwendung offener Standards, um diese Daten einheitlich darzustellen und die verschiedenen Gewerke des Baugewerbes besser miteinander zu verknüpfen. Der im Baugewerbe anerkannte IFC (Industry Foundation Classes) -Standard [A 25] (ISO-16739) dient als Datenmodell für OpenBIM. Er wird von der internationalen buildingSMART-Gesellschaft [A 26] gepflegt und erweitert. Neben Raumplänen umfassen OpenBIM-Modelle auch installierte Geräte und deren Anwendungsdomäne. So finden sich Daten über Sensoren und Aktoren von Heizkreisen und Beleuchtungsinstallationen. Insbesondere die Informationen über Installationsorte und physische Erreichbarkeit lassen sich aus den Modellen extrahieren. Der OpenBIM-Prozess sieht vor, dass die Modelle während des Planungs- und Umsetzungsprozesses zwischen den einzelnen Gewerken als Informationsbasis ausgetauscht und erweitert werden. Selbst im Wartungsabschnitt des Gebäudelebenszyklus werden die Modelle aktualisiert. Umbaumaßnahmen, weitere Geräteinstallationen und die Demontage von Geräten werden in die Modelle eingetragen. Somit stellen OpenBIM-Modelle einen sogenannten Digital Twin dar, der den aktuellen Gebäudestatus repräsentiert und protokolliert.

Im nachfolgenden Abschnitt werden gängige Protokolle der Gebäudeautomation vorgestellt. Dabei wird näher auf die einzelnen Funktionalitäten und die inhärenten Security-Eigenschaften eingegangen. Abschließend wird ein zukunftssicherer Referenz-Protokoll-Stack vorgeschlagen.

3 Protokoll- und Technologieauswahl für die Gebäudeautomation

Das Ziel ist einen zukunftssicheren Protokollstack zu finden, der sämtliche Anwendungen zwischen eingebetteten Systemen einer Gebäudeautomation (GA) realisiert [B 1]. Dazu werden zunächst die klassischen Protokolle der GA erläutert. In den vergangenen Jahren wurden Webtechnologien auf eingebetteten Systemen verwendet, um eine interoperable Kommunikation zu ermöglichen. Die wichtigsten Web-Protokolle und -standards werden hinsichtlich ihrer Security-Eigenschaften und ihrer Eignung für GA-Systeme ausgewertet. Neben sporadischen Kommunikationsflüssen treten in realen Szenarien Streaminganwendungen mit erhöhter Datenrate und Anforderungen an die Latenz auf. In der eigenen Publikation [B 2] wurden die Experimentalergebnisse bezüglich eines Datenstreamings diskutiert. Der gefundene Protokollstack ermöglicht ein Gesamtsystem, das aus wenigen Teilkomponenten besteht. Dadurch, dass mithilfe eines Protokollstacks sämtliche Anwendungen und Gerätetypen implementiert werden können, entfallen heterogene Protokollimplementierungen. Die Wahrscheinlichkeit, dass sich im Netzwerk mindestens ein Gerät mit einer fehlerhaften Implementierung befindet, sinkt. Aufgrund der geringeren Komplexität des Gesamtsystems steigt das Security-Niveau. Der Referenz-Potokollstack dient als Grundlage, alle relevanten Komponenten der selbst entwickelten Security-Architiktur für das GA-System zu implementieren und experimentell zu evaluieren.

3.1 Funktionale Anforderungen

In Tabelle 3.1 werden funktionale Anforderungen an ein Protokoll für das GA-Umfeld aufgelistet. Außerdem wird die Bedeutung jeder einzelnen Anforderung begründet.

3.2 Klassische Protokolle der Gebäudeautomation

Die beiden größten Vertreter von M2M-Protokollen in der Gebäudeautomation sind KNX [A 1] und BACnet [A 2]. In [A 24] werden die wesentlichen Protokolleigenschaften und Funktionalitäten aufgeführt.

Tabelle 3.1: Funktionale Anforderungen an ein Prodokoll im GA-Umfeld

Anforderung	**Begründung**
Geringe Nachrichtengrößen	Kleine Nachrichtengrößen können den Energieverbrauch bei drahtloser Kommunikation reduzieren. Entscheidend ist die benötigte Energie pro Byte. Außerdem führen kleine Nachrichtengrößen dazu, dass Nachrichten in einzelnen Frames übertragen werden können. Somit sind weniger Kanalzugriffe erforderlich.
Asynchrone Benachrichtigungen	Eingebettete Systeme, wie Bewegungsmelder, Temperatursensoren oder Lichtschalter müssen in der Lage sein, Zustandsänderungen asynchron an eine Gegenstelle zu kommunizieren. Dies resultiert im Vergleich zu einem Polling-basierten System in einem deutlich reduzierten Nachrichten-Overhead. Dabei fragt die Gegenstelle (Client) zyklisch Informationen vom Server ab.
Geeignet für Anwendungen mit hoher Datenrate	Neben Anwendungen mit geringer Datenrate und sporadischem Nachrichtenaufkommen sollte ein GA-Protokoll Anwendungen mit hoher Datenrate und Latenz ermöglichen. Dieses Verkehrsmuster tritt bei der Übertragung von Audio-/Videostreams von Gegensprechanalagen oder Überwachungskameras auf.
Ende-zu-Ende Sicherheit	Die Ende-zu-Ende Sicherheit gewährleistet, dass Daten auf dem Übertragungsweg nicht von Dritten gelesen werden können. In einem modernen Ga-System befinden sich möglicherweise bösartige Geräte, die unter der Kontrolle eines Angreifers stehen.
IP-basierte Kommunikation zwischen Endgeräten	Ein IP-basiertes lokales Netzwerk kann einfach durch Infrastrukturgeräte erweitert werden. Außerdem ist es möglich, mehrere Gebäude über das WAN (Wide Area Network) miteinander zu verknüpfen.
Anwendungslogik in Endgeräte integrierbar	Die Anwendungslogik sollte sich in die Endgeräte integrieren lassen. Einzelne Steuergeräte, die Sensordaten auswerten und die Aktorik steuern, stellen einen sogenannten Single Point of Failure dar.
Datenmodell vorhanden	Einheitliche Datenmodelle gewährleisten die Kompatibilität zwischen Geräten verschiedener Hersteller.
Datenmodell nachrüstbar	Besitzt ein Protokoll kein inhärentes Datenmodell, so sollte es durch ein Modell erweitert werden können.

3.2.1 KNX

Ein in der GA seit Jahren verbreiteter Standard ist KNX [A 1]. Der Standard ist in die Klasse der Feldbussysteme einzuordnen. Mit der Norm ISO/IEC 14543-3 ist der Standard seit November 2006 auf internationaler Ebene bekannt. Ein charakteristisches Merkmal von KNX ist die Trennung von Gerätestromversorgung- und steuerung. Die Gebäudestromversorgung und die Verlegung der Datenleitungen zur Übermittlung von Steuerbefehlen und Daten können unabhängig voneinander organisiert sein. Dadurch ist es möglich, die Funktionalität zwischen Geräten zu verändern. So kann beispielsweise ein Schalter durch das Senden von Steuerdaten verschiedene Geräte ein- oder ausschalten.

Security

Die größten Nachteile birgt KNX hinsichtlich der Security. Es sind lediglich rudimentäre Sicherheitsvorkehrungen getroffen, die sich leicht umgehen lassen. Es ist ein optionaler Passwortmechanismus vorgesehen, um die Anwendung von Konfigurationsdaten zu autorisieren. Jedoch wird das Passwort samt Konfiguration im Klartext übertragen, sodass sich diese sensiblen Daten über die Busstruktur durch passives Sniffen durch einen Angreifer auslesen und wiederverwenden lassen. Ein Angreifer hätte in diesem Fall uneingeschränkte Möglichkeiten, sich durch das Passwort zu autorisieren und weiter vorzugehen. Dem Standard ist zu entnehmen, dass die beste Strategie eine Abschottung des KNX-Busses gegenüber der Außenwelt ist. Dies ist in vielen Installationen häufig nicht möglich, da sich Geräte und damit der Zugang zum KNX-Bus in öffentlich zugänglichen Bereichen befinden.

3.2.2 BACnet/IP

Der 2012 veröffentlichte BACnet-Standard [A 2] definiert die Kommunikation zwischen Maschinen durch standardisierte Objekte. Außerdem werden sogenannte BACnet-Dienste, die u.a. den Zugriff auf Objekte und Dateien ermöglichen sowie Geräte- und Management-Funktionalitäten realisieren, definiert. Durch den Geräte- und Netzwerkmanagement-Dienst lassen sich Geräte und Objekte eines Netzwerks auffinden. Wird eine WhoIs-Anfrage durch einen BACnet-Client an einen oder mehrere Server (Unicast bzw. Broadcast) gesendet, liefert der Server oder liefern die Server ihre Netzwerkadresse und ihre Objektbezeichner [A 27]. Durch eine WhoHas-Anfrage kann ein Client selektiv alle Geräte ansprechen, die einen

Objektbezeichner unterstützen. Wenn ein Client alle benötigten Daten über einen BACnet-Server besitzt, ist er in der Lage, über den Objektzugriff-Dienst des Servers beispielsweise Sensoren und Aktoren anzusteuern. Der Objektzugriff-Dienst stellt die Schnittstelle dar, auf Werte lesend und schreibend zuzugreifen. Dabei wird zwischen analogen und digitalen Ein- und Ausgängen unterschieden. BACnet liefert ein einheitliches Datenmodell, das verschiedene Gerätetypen eines GA-Systems beschreibt. Somit ist es möglich, dass Geräte herstellerübergreifend miteinander kommunizieren können. Neben den Request-Response-Schema definiert BACnet Alarm- und Ereignisdienste [A 27]. Dadurch ist es möglich, dass ein Client über Server-Benachrichtigungen Informationen erhält. Dieses Verfahren wird auch als Publish-Subscribe-Mechanismus bezeichnet. Der Server sendet an alle Clients, die als Subscriber (dt.: Abonnenten) registriert sind, eine Notification (dt.: Benachrichtigung), sobald sich der Status eines Wertes geändert hat. Die BACnet-Architektur trennt zwischen Geräten der Feldbusebene und der Steuerungsebene. Ein BACnet-Controller kommuniziert in Form einer Stern-Topologie mit den einzelnen Sensoren und Aktoren. Controller-Funktionalitäten werden nicht in die Endgeräte integriert. Der BACnet-Standard definiert u.a. die Kommunikation über IP-basierte Netze. Neben dem Internetprotokoll Version 4 (IPv4) kann auch IPv6 eingesetzt werden, um BACnet-Nachrichten zu übermitteln. Bereits im Jahr 2004 wurde in [A 28] ein Videodatenstreaming-Konzept für BACnet vorgestellt. Jedoch fand dieses Verfahren keine Anwendung in der Praxis.

Security

BACnet Secure Connect (BACnet/SC) [A 29] ermöglicht in IP-basierten Netzwerken BACnet-Nachrichten auf Transportebene abzusichern. Dabei wird ein geschützter Tunnel zwischen Server und Client mittels TLS 1.3 hergestellt. Somit können Nachrichtenintegrität, -authentizität und die Vertraulichkeit sichergestellt werden. Es werden keine speziellen Nachrichtenformate eingeführt.

3.3 Web-Technologien und ihre Eignung für die Gebäudeautomation

In den vergangenen Jahren wurden im Forschungsbereich verstärkt IP-basierte Web Technologien zur M2M-Kommunikation eingesetzt. Sie zeichnen sich durch ei-

ne lose Kopplung (ohne feste Kommunikationstechnologie) zwischen den einzelnen Geräten aus und ermöglichen eine nahtlose Verbindung an das Internet.

3.3.1 MQTT - Message Queuing Telemetry Transport

Die MQTT-Spezifikation [A 30] sieht eine Architektur bestehend aus einem Broker und Clients vor. Der MQTT-Broker wird als ein Server angesehen, da er Anfragen von Clients entgegennimmt, ohne von sich aus eine Verbindung zu anderen Teilnehmern aufzubauen. Das Protokoll gehört zu der Klasse der Publish-Subscriber Protokolle. Informationen werden von einem Publisher an das korrespondierende Topic (Thema) eines Brokers gesendet. Ein Topic unterteilt sich auf logischer Ebene in eine Hierarchie von einzelnen IDs. Andere Teilnehmer, auch Subscriber (Abonnenten) genannt, werden durch den Broker über Änderungen eines Topics benachrichtigt. Die Daten werden dabei über WebSockets oder eine TCP-Verbindung versendet.

Security

Die MQTT-Spezifikation sieht keine zwingend notwendigen Sicherheitsmaßnamen vor. Lediglich wird eine Client-Authentifikation von Connect-Nachrichten mittels Nutzernamen und Passwort beschrieben. Die Spezifikation lässt dem Entwickler die Freiheit zur konkreten Implementierung eines Client-Authentifikationsverfahrens. So können Nutzernamen und Passwörter beispielsweise als Hashes übertragen und wahlweise an eine zentralisierte Nutzerverwaltung wie Kerberos [A 31], LDAP (Lightweight Directory Access Protocol) [A 32] oder Radius [A 33] übertragen werden. Weiterhin bietet die MQTT-Spezifikation die Möglichkeit, einen TLS-Tunnel zwischen Publisher und Broker bzw. Broker und Subscriber aufzubauen. Die einzelnen Security-Eigenschaften von MQTT sind in Tabelle 3.2 aufgelistet.

3.3.2 HTTP(S) - Hypertext Transfer Protocol (Secure)

Das Hypertext Transfer Protocol (HTTP) [A 34] ist ein Protokoll der Anwendungsschicht und unterstützt REST (Representational State Transfer) -Methoden. HTTP ist damit ein Vertreter der zustandslosen Protokolle. Durch die zeichenbasierte Codierung kann der Nachrichtenoverhead durch HTTP als relativ groß angesehen werden. HTTP folgt dem Server-Client-Prinzip, bei dem ein Server nur nach vorheriger Anfrage eine Antwort an einen Client sendet. Der Server hostet Ressourcen, die von

Tabelle 3.2: Security-Eigenschaften von MQTT

Eigenschaft	**Umsetzung**
Anwendungsschicht-Sicherheit	Nein
Transportschicht-Sicherheit	Ja
Integrität	Ja
Authentizität	Ja
Vertraulichkeit	Ja
Serverauthentifikation	Durch Zertifikat
Clientauthentifikation	Durch Zertifikat oder durch Nutzername und Passwort

Clients beispielsweise angelegt, verändert, abgefragt oder gelöscht werden können. Ein Abonnement einer Ressource mit anschließender Benachrichtigung des Clients sind in HTTP nicht vorgesehen. Um eine entsprechende Funktionalität zu realisieren, kommen RSS (Rich Site Summary) [A 35] -Feeds bzw. WebSockets [A 36] zum Einsatz.

Security

HTTP bietet die Möglichkeit Clients zu authentifizieren. Der Standard beschreibt zwei Methoden. Ein Client kann Nutzernamen und Passwort innerhalb einer HTTP-Nachricht als Klartext an den Server übertragen. Dieses Verfahren bietet jedoch keinen Schutz vor Abfangen und Mitschneiden sowie vor Manipulation durch einen Angreifer. Diese sogenannte “Basic“-Methode ist daher ohne zusätzliche Sicherung auf Transportschicht nicht empfehlenswert. Die “Digest“-Methode hingegen kann ohne Sicherung auf unterer Schicht verwendet werden. Der Server sendet eine Nonce an den Client. Dabei handelt es sich um eine Zufallszahl, die lediglich einmal verwendet wird. Dieser berechnet aus der Nonce, dem Nutzernamen, dem Passwort, der URL und der HTTP-Methode einen Hashwert und sendet ihn mit der Anfrage mit. Der Server prüft nach Empfang der Client-Nachricht die Authentizität durch Berechnung desselben Hashwertes. Dabei benötigt der Server Kenntnis über Nutzername und Passwort, die vorher über einen sicheren Kanal übertragen werden müssen. Durch die Verwendung einer zufälligen Nonce können Replay-Angriffe unwirksam gemacht werden. Sensible Daten, wie die Nutzername-Passwort-Kombination, sind durch die Verwendung einer geeigneten Hashfunktion nicht rekonstruierbar. Jedoch

ist bei dieser Methode kein Integritätsschutz gewährleistet, da die Payload nicht mit in die Hashwertberechnung einfließt. Die Variante HTTPS (Hypertext Transfer Protocol Secure) verwendet auf Transportschicht TLS (Transport Layer Security) [A 20], [A 37]. Dadurch erzielt HTTPS (Tabelle 3.3) die selben Security-Eigenschaften von MQTT.

Tabelle 3.3: Security-Eigenschaften von HTTPS

Eigenschaft	**Umsetzung**
Anwendungsschicht-Sicherheit	Nein
Transportschicht-Sicherheit	Ja
Integrität	Ja
Authentizität	Ja
Vertraulichkeit	Ja
Serverauthentifikation	Durch Zertifikat
Clientauthentifikation	Durch Zertifikat oder durch Nutzername und Passwort

3.3.3 CoAP - Constrained Application Protocol

CoAP [A 38], das Constrained Application Protocol, findet seine Anwendung in der Implementierung von RESTful APIs (Representational State Transfer Application Programming Interface). Ein Server stellt seine Funktionalität Clients in Form von Ressourcen, die ein API darstellen, zur Verfügung. CoAP ist als ein Vertreter der zustandslosen Protokolle anzusehen. Anfragen an ein und dieselbe Ressource führen nicht zu einer Zustandsänderung der Ressource. Sie werden unabhängig von der Anfragehistorie beantwortet. Dienste oder auch Funktionalitäten in Form einer RESTful-Architektur bereitzustellen, bietet den Vorteil der Modularisierung. Ein komplexes Gesamtsystem lässt sich durch Unterteilen in die einzelnen Teilkomponenten übersichtlicher strukturieren. Der resultierende Vorteil weniger komplexer Teilkomponenten, die auf RESTful Services abgebildet werden, vereinfacht den Design-, Implementierungs- und Wartungsprozess. Beispielsweise wird die Anfrage eines Clients von einem Server entgegengenommen und verarbeitet. Der Server fragt daraufhin weitere Services an, die auf anderen Servern gehostet werden können. Die gesammelten Antworten werden vom Server wiederum verar-

beitet und an den ursprünglichen Anfragesteller (Client) zurückgesendet. Man erhält dadurch ein mehrstufiges System aus Servern, die modular organisiert einen Gesamtservice darstellen. Dieses REST-Konzept ist auch als Schichtenarchitektur bekannt. Das Code-on-demand-Prinzip ist bei CoAP möglich. So wird ein ausführbarer Code oder ein Skript von einem Server an einen Client gesendet. Dies ist mit dem Abrufen von JavaScript-Code eines Webbrowsers via HTTP von einem Server vergleichbar. Ein weiteres REST-Prinzip, das durch CoAP angeboten wird, ist das Caching, bei dem Antworten von einem Server zwischengespeichert werden. Da CoAP, genau wie HTTP, Ressourcen in RESTful APIs organisiert (Tabelle 3.4) und CRUD-Operationen GET, PUT, POST und DELETE unterstüzt, lassen sich entsprechende CoAP/HTTP-Proxies realisieren. Ein solcher Proxy wertet die Nachrichtenheader, insbesondere die Zielressource samt Query, aus und wandelt die Nachrichtenformate um. Der Proxy wandelt dabei die binär codierten Header von CoAP in das HTTP-Pendant um. Während des Designprozesses von CoAP ist die Designentscheidung auf einen binären Nachrichtenkopf gefallen, um Nachrichtengrößen zu reduzieren. Geräte mit limitierten Energieressourcen (wie beispielsweise durch Akkus und Knopfzellen) müssen durch die Verwendung eines binär codierten Protokolls weniger Daten senden. Insbesondere das Senden über einen Drahtloskommunikationsstandard gilt als teuer für das Energiebudget. Ein weiterer Faktor, der zur Energieoptimierung von CoAP beiträgt, ist die Limitierung der maximalen Nutzdatengröße, sodass die resultierende Nachrichtengröße samt UDP-Header mit einem einzelnen IP-Paket übertragen werden kann. Falls keine Informationen über den Link Layer existieren, sollten laut CoAP-Standard von einer IP Maximum Transmission Unit (MTU) von 1280 Byte ausgegangen werden und die gesamte CoAP-Nachrichtengröße von 1152 Byte, wovon 1024 Byte auf die Nutzdaten entfallen, gewählt werden. Bei einer Verwendung von Ethernet bzw. WLAN würde keine Fragmentierung auftreten. Durch das Senden mehrerer Frames, um eine einzelne CoAP-Nachricht zu übertragen, entstünde ein erhöhter Energiebedarf. CoAP nutzt auf der Transportschicht UDP. Um die QoS zu gewährleisten, können CoAP-Nachrichten optional durch ein Flag im Header als “confirmable“ gekennzeichnet werden. Die Gegenstelle einer solchen Nachricht quittiert den Empfang durch ein Acknowlege (ACK) welches “piggy-backed“ in eine andere Nachricht eingebettet sein kann. Erhält der Server oder Client kein ACK, so wird nach Vorschlag des CoAP-Standards eine Neuübertragung nach Ablauf einer exponentiell steigenden Wartezeit ausgeführt. Das Muster und die maximale Anzahl an Übertragungsversuchen können vom

Entwickler jedoch frei verändert werden. Damit sind energieoptimierte Strategien der QoS-Gewährleistung möglich.

Der bekannteste Vertreter eines solchen Schemas ist die sogenannte Mikroservicearchitektur. Dabei wird das Gesamtsystem in möglichst kleine Systeme/Services unterteilt. Dies soll am Beispiel einer Alarmanlage erläutert werden: Spezifiziert wird die Funktionalität, dass, im Falle eines Einbruchs, ein Alarm ertönt und das Licht eingeschaltet wird. Ist eine autorisierte Person anwesend, soll kein Alarm ausgelöst werden. Eine mögliche Architektur besteht aus Mikroservices für die Sensorik (Bewegungsmelder, Lichtschranken oder Kameras mit Personenerkennung), der Aktorik (Sirene und Beleuchtung) und einem oder mehreren Services zur Erkennung autorisierter Personen (PIN-Pad oder NFC-Kartenleser). Jeder Mikroservice verfügt über ein Interface (RESTful API), eine interne Verarbeitungslogik und optional über eine persistente Datenhaltung.

Tabelle 3.4: CRUD-Operationen

Operation	**RESTful Web Service**	**Bedeutung**	**Anwendung**
CREATE	PUT/POST	Anlegen einer neuen Ressource	Client legt Ressource mit URI an, falls noch keine Ressource mit derselben URI vorhanden ist
READ	GET	Anfrage an eine Ressource	Datum wird mittels URI adressiert und abgerufen; Auslösen einer Serverseitigen Aktion ist möglich; bspw.: Anfrage an eine Lichtsteuerung zum Einschalten der Beleuchtung
UPDATE	PUT	Senden von Daten an eine Ressource	Ein Client fungiert als Datenquelle und sendet Datum an eine URI; bspw.: Senden eines Sensorwertes an einen Server
DELETE	DELETE	Löschen einer Ressource	Ressource wird nach Beendigung eines technischen Prozesses nicht mehr benötigt

Security

Der CoAP-Standard sieht eine optionale verschlüsselte und signierte Kommunikation auf Transportschicht vor. Da CoAP auf UDP basiert, dient DTLS (Datagram Transport Layer Security) als Sicherungsschicht. Die Security-Schutzziele werden

in DTLS durch Anwendung von symmetrischen und asymmetrischen Verschlüsselungsverfahren erreicht. In der CoAP-Standardfamilie findet sich außerdem das Security-Protokoll OSCORE [A 39], das die Schutzziele auf Anwendungsschicht realisiert. OSCORE basiert auf speziellen Nachrichtenformaten die durch die IETF definiert werden. Nachfolgend werden die einzelnen Standards erläutert:

3.3.4 CBOR - Concise Binary Object Representation

Der IETF-Standard CBOR (Concise Binary Object Representation) existiert seit 2013 und beschreibt ein binäres Nachrichtenformat [A 40]. Durch die binäre Codierung lassen sich sehr kleine Nachrichtengrößen realisieren, weshalb sich das Nachrichtenformat für IoT-Anwendungen eignet. Weiterhin wurde während des Standardisierungsprozesses Rücksicht auf kleine Codegrößen der Nachrichtenserialisierer und -parser genommen. CBOR lässt sich direkt mit dem Nachrichtenformat JSON (JavaScript Object Notation) vergleichen. Beide Formate strukturieren die Informationen einer Nachricht in Form von sogenannten Key-Value-Paaren. Dabei adressiert ein Key den zugehörigen Wert. Diese Paare können als 2-Tupel betrachtet werden. Jedoch kommt bei JSON im Vergleich zu CBOR ein menschenlesbares Format mit dem Nachteil größerer Nachrichtengrößen zum Einsatz. Eine CBOR-Nachricht muss zur menschenlesbaren Darstellung durch einen Parser aufbereitet werden. CBOR ist ein reiner Nachrichtenformatsstandard, der keinerlei Security-Spezifikationen/-Konzepte beinhaltet.

3.3.5 COSE - CBOR Object Signing and Encryption

Der COSE-Standard ist unter dem RFC 8152 bekannt [A 41]. Er beschreibt die Ver- und Entschlüsselung von Nachrichten, die auf dem CBOR-Standard basieren. Außerdem definiert COSE die Sicherstellung der Nachrichtenintegrität mittels Message Authentication Codes (MAC). COSE stellt das Pendant zum JSON Object Signing and Encryption Standard (JOSE) dar [A 42].

3.3.6 OSCORE - Object Security for Constrained RESTful Environments

Seit dem Jahr 2019 existiert mit dem IETF Standard RFC 8613 namens OSCORE eine Lösung zur Realisierung der Anwendungsschichtsicherheit für CoAP [A 43]. Der CoAP-Standard beschreibt die Absicherung der Kommunikation zwischen zwei

CoAP-Geräten mittels DTLS. Ein wesentlicher Vorteil von CoAP besteht jedoch in der Kompatibilität zu HTTP. Ein Proxy zur Vermittlung zwischen beiden Protokollen stellt eine architekturelle Schwachstelle im Netz aus CoAP- und HTTP-Geräten dar. Der Proxy muss in der Lage sein, die Nachrichtenheader auf Anwendungsschicht zu parsen, um beispielsweise die Ziel-URI aus dem binären CoAP-Header in den HTTP-Header zu übersetzen. Dabei ist lediglich eine Ende-zu-Ende-Sicherheit auf Basis von DTLS zwischen dem CoAP-Gerät und dem Proxy bzw. auf Basis von TLS zwischen Proxy- und HTTPS-Gerät möglich. OSCORE hingegen ermöglicht die selektive Absicherung von Nachrichteninhalten durch die Verwendung von COSE. Um durch dieses Nachrichtenformat in Verbindung mit der Funktionalität einzelne Nachrichteninhalte zu schützen, lässt sich die URI als “signed-only“ übertragen. Dies ermöglicht Brokern oder Geräten innerhalb einer mehrstufigen Architektur (CoAP-Broker-HTTPS) die URI zu lesen, die Integrität und Authentizität zu validieren und in ein anderes Format umzuwandeln. Andere Nachrichteninhalte wie die Payload lassen sich optional verschlüsseln, um die Vertraulichkeit zu gewährleisten. In Tabelle 3.5 werden die einzelnen Security-Eigenschaften von OSCORE zusammengefasst.

Tabelle 3.5: Security-Eigenschaften von OSCORE

Eigenschaft	**Umsetzung**
Anwendungsschicht-Sicherheit	Ja
Transportschicht-Sicherheit	Nein
Integrität	Ja
Authentizität	Ja
Vertraulichkeit	Ja
Serverauthentifikation	Durch Zertifikat oder Pre-Shared-Key
Clientauthentifikation	Durch Zertifikat oder Pre-Shared-Key

3.4 Auswahl des Anwendungsschichtprotokolls

In Tabelle 3.6 sind die einzelnen Anforderungen an ein Anwendungsschichtprotokoll für die GA aufgelistet. CoAP/OSCORE hat sich als Protokoll herauskristallisiert, das sich für die Kommunikation am besten eignet. Durch geringe Nachrich-

tengrößen sind ressourcenschwache Geräte in der Lage, das Protokoll auszuführen. Werden die Nachrichten über ein drahtloses Netzwerk gesendet, führt dies zu kurzen Sendevorgängen, sodass Energie gespart werden kann. Des Weiteren sind asynchrone Benachrichtigungen von einem Server zu den einzelnen abonnierenden Clients möglich. Diese Funktionalität führt dazu, dass lediglich im Falle einer Zustandsänderung Nachrichten versendet werden müssen, anstatt zyklisch Anfragen an den Server senden zu müssen. Neben der Übertragung von Sensorwerten und Steuerbefehlen mit geringen Datenraten sollte ein zukunftssicheres Protokoll Live-Streaming von beispielsweise Audio-/Videostreams ermöglichen. In einer eigenen Studie (Kapitel 4) wird ein CoAP-basiertes Verfahren hinsichtlich seiner Performance-Eigenschaften näher evaluiert. Es existieren bereits Arbeiten [A 44] und [A 45], die die Eignung von CoAP für diesen Anwendungsfall bestätigen. Mit OSCORE ist es möglich, Daten von Endpunkt-zu-Endpunkt abzusichern (Vertraulichkeit, Integrität und Authentizität zu gewährleisten). Dies ist sogar im Zusammenspiel mit HTTP/CoAP-Brokern möglich. CoAP-basierte Endgeräte können mit HTTP-basierten Webanwendungen verknüpft werden. Die Nachrichtenübermittlung erfolgt IP-basiert über das Internet und lokale Netzwerke. Außerdem ist es möglich, Anwendungslogiken nicht über eine zentrale Instanz (Webanwendung oder SPS-Steuergerät), sondern direkt in den Endgeräten umzusetzen [B 3]. Diese Eigenschaft spielt eine zentrale Rolle in der in Kapitel 6 entwickelten Security-Architektur. CoAP bzw. OSCORE besitzen kein inhärentes Datenmodell, um die Kompatibilität zwischen Geräten verschiedener Hersteller zu gewährleisten. Jedoch ist es möglich, ein Datenmodell wie beispielsweise das der Open Mobile Alliance [A 46] zu verwenden. Das resultierende IoT-Protokoll trägt den Namen ”LWM2M” (Lightweight Machine-to-Machine).

Tabelle 3.6: Anwendungsschichtprotokolle für die Gerätevernetzung im GA-Umfeld

Anforderung	**KNX**	**BACnet/IP**	**HTTP**	**MQTT**	**CoAP/OSCORE**
Geringe Nachrichtengrößen	Ja	Ja	Nein	Ja	Ja
Asynchrone Benachrichtigungen	Ja	Ja	Nein	Ja	Ja
Geeignet für Anwendungen mit hoher Datenrate	Nein	Ja	Ja	Nein	Ja
Ende-zu-Ende Sicherheit	Nein	Ja	Ja	Nein	Ja
IP-basierte Kommunikation zwischen Endgeräten	Nein	Ja	Ja	Ja	Ja
Anwendungslogik in Endgeräte integrierbar	Nein	Nein	Ja	Ja	Ja
Datenmodell vorhanden	Ja	Ja	Nein	Nein	Nein
Datenmodell nachrüstbar	-	-	Ja	Ja	Ja

In einem modernen GA-System kommunizieren Geräte über eine Netzwerkinfrastruktur, um Daten und Befehle auszutauschen. Funknetzwerke bieten gegenüber kabelgebundenen Infrastrukturen den Vorteil, dass sie nicht zur Designzeit eines Gebäudes geplant werden müssen. Außerdem sind sie günstiger und flexibler bei Änderungen der GA. Im nachfolgenden Abschnitt werden verschiedene Kommunikationsstandards der Drahtloskommunikation erläutert. Es wird ein Funkstandard gesucht, der die CoAP/OSCORE-Nachrichten übermittelt.

3.5 Funkstandards

Drahtlose Kommunikationsstandards ermöglichen es, Geräte ohne zusätzliche Datenleitungen miteinander kommunizieren zu lassen. Physische Bussysteme, wie sie bei KNX vorgesehen sind, müssen während der Planungsphase eines Gebäudes berücksichtigt werden und später eingebaut werden. Eine Änderung zur Laufzeit ist mit hohem Aufwand und Kosten verbunden. Ähnliches gilt für Ethernet-basierte

Netzwerkinfrastrukturen. Die Verlegung von Netzwerkkabeln, Switches und Routern muss frühzeitig geplant werden. Aufputz-Kabelschächte, um eine Netzwerkanbindung der einzelnen Räume zu gewährleisten, sind aufwändig zu installieren und stören die Ästhetik der Inneneinrichtung. Drahtlose Kommunikationskanäle lösen diese Probleme. In Tabelle 3.7 sind die wichtigsten lizenzfreien Vertreter aufgelistet. Lizenzfreie Bänder können ohne Gebühren verwendet werden.

Tabelle 3.7: Lizenzfreie Funkstandards für die Gerätevernetzung im GA-Umfeld

Eigenschaft	**IEEE 802.11**	**IEEE 802.15.4 (+6LoWPAN)**	**ZigBee (IP)**	**Z-Wave**	**Bluetooth**	**LoRa (WAN)**
Sub-1-GHz-Band	Ja	Ja	Ja	Ja	Nein	Ja
2,4-GHz-Band	Ja	Ja	Ja	Nein	Ja	Nein
5-GHz-Band	Ja	Nein	Nein	Nein	Nein	Nein
Reichweite	Hoch	Mittel	Mittel	Mittel	Mittel	Hoch
Topologie	Stern, P2P, Mesh	Stern, P2P, Baum	Stern, P2P, Baum	Mesh	P2P, Mesh	Stern
P2P-Security	Ja	Ja	Ja	Ja	Ja	Ja
Datenrate	Hoch	Gering	Gering	Gering	Gering	Gering
IP-Kommunikation	Ja	Ja	Ja	Nein	Nein	Nein

IEEE 802.11 (WLAN)

Die IEEE 802.11-Standardfamilie [A 47] (umgangssprachlich ”WLAN” (Wireless Local Area Network)) war ursprünglich als drahtlose Ethernet-Variante konzipiert. In Anwendungsfällen, in denen Ethernetkabel nicht verlegt werden können, soll WLAN eine Konnektivität zwischen PCs und anderen Netzwerkgeräten gewährleisten. Im Zuge einer stetigen Weiterentwicklung der WLAN-Familie lassen sich immer größere Datenraten erzielen. Realisiert werden Datenraten von über 1 Gbit/s durch Nutzung breiterer Bänder von bis zu 160 MHz und effizienterer Codierungsschemata (IEEE 802.11 ac und ax) im Vergleich zu den Vorgängergenerationen IEEE 802.11 a,b,g,n. Die aktuelle IEEE 802.11 ax-Erweiterung berücksichtigt die speziellen Anforderungen an moderne IoT-Netzwerke mit großen Teilnehmerzahlen. Bis auf Streaming-Anwendungen, wie z.B. Audio-/Videostreaming von Überwachungskameras des Gebäudemanagements, haben alle weiteren Anwendungen andere Anforderungen an das Netzwerk. In einem System, das aus sehr vielen Senso-

ren und Aktoren besteht, werden sporadisch bzw. periodisch kleine Datenmengen versendet. Dabei handelt es sich überwiegend um Messdaten bzw. Steuerbefehle. Aufgrund von limitierten Energiereserven (Batterien oder Akkus) müssen eingebettete Systeme mit wenig Energie pro Sendevorgang auskommen. Ein Abschalten des WLAN-Transceivers zur Energieeinsparung ist dabei essentiell. Befindet sich der Transceiver im sogenannten Sleep-Modus, können keine Nutzdaten in Form von Daten-Frames empfangen werden. Die aktuelle IEEE 802.11 ax-Erweiterung führt einen Aufweckmechanismus ein, der es ermöglicht, Endgeräte aufzuwecken und die Übertragung von Nutzdaten zu starten. Alle WLAN-Generationen beschreiben die MAC-Layer-Kommunikation zwischen den einzelnen Endgeräten (im Standard ”Stations” genannt) untereinander in Form einer Direktverbindung (P2P) und über einen Access Point (AP). Im Falle einer AP-basierten Struktur handelt es sich um eine Sterntopologie. Möchte eine Station Daten-Frames senden, so wird die Weiterleitung durch den AP implementiert. Dies geschieht, selbst wenn der Empfänger eines Frames Teil derselben Sterntopologie ist. WLANs können logisch mit mehreren APs im Verbund existieren, um größere Bereiche abzudecken. Die Netzwerkinfrastruktur besteht aus APs, die z.B. mit Ethernetkabeln, Switches und/oder Routern verbunden sind. Wird ein Gebäude mit einem solchen Netzwerk ausgestattet, führt dies zwangsläufig zu einem großen Planungs- und Kostenaufwand für die Infrastruktur. Der Erweiterungsstandard für den MAC Layer IEEE 802.11s [A 48] löst dieses Problem durch eine vermaschte Netzwerktopologie. Er definiert eine Erweiterung der MAC-Schicht um den Mesh-Modus. Frames werden auf MAC-Ebene durch jeden Teilnehmer des Netzwerks weitergeleitet. Pfade durch das Netzwerk werden dynamisch zur Laufzeit bestimmt, sodass sich gestörte Einzelverbindungen umgehen lassen. Der Hauptvorteil der IEEE 802.11s-Erweiterung besteht in den Einsparungen der Netzwerkinfrastruktur. Außerdem ist es möglich, ein IEEE 802.11s-basiertes Netzwerk an eine Ethernetinfrastruktur anzubinden. Die Schnittstelle zwischen beiden Netzwerkstandards bilden sogenannte ”Mesh Gates”. Zusammengefasst erfüllt die WLAN-Standardfamilie Anforderungen verschiedener Anwendungsarten (hohe Datenraten und Systeme mit vielen Teilnehmern mit sporadischen Sendewünschen) und Gerätetypen (leistungsstarke Stations und batteriebetriebene IoT-Geräte). Durch die Verwendung von Funkkanälen geringer Bandbreite im Sub-1-GHz Band und/oder die Möglichkeit, vermaschte Netzwerktopologien aufzubauen, wird die Abdeckung über große Distanzen ermöglicht.

IEEE 802.15.4

Ein anderes Netzwerkprotokoll der Bitübertragungs- und MAC-Schicht ist IEEE 802.15.4 [A 49]. Dieser Standard wurde ursprünglich für Wireless Personal Area Networks entwickelt, um batteriebetriebene Sensoren und Aktoren miteinander zu vernetzen. Zur Bitübertragung werden Kanäle des Sub-1GHz-Bereichs und des ebenso lizenzfreien 2,4 GHz-Bandes verwendet. Durch die schmalen Bandbreiten lassen sich jedoch nur Bitraten von 20-40 kbit/s (je nach Region zugelassene Bandbreite) im Sub-1-GHz-Band und 250 kbit/s im 2,4 GHz-Band erzielen. Durch eine Baum-Topologie lässt sich im Vergleich zu Stern- oder Punkt-zu-Punkt-Topologien ein größerer Bereich durch Multi-Hop Übertragungen abdecken. Protokolle oberhalb der MAC-Schicht müssen jedoch das Routing zwischen den einzelnen Teilnehmern implementieren.

6LoWPAN

6LoWPAN (IPv6 over Low power Wireless Personal Area Network) ist ein IETF-Standard um IPv6-Pakete über drahtlose IEEE 802.15.4-Netzwerke zu übertragen [A 50]. Header-Informationen werden komprimiert, indem IPv6-Adressen gekürzt werden. Da die maximale Paketgröße von IEEE 802.15.4 127 Byte groß ist und die IPv6-MTU einen Minimalwert von 1280 Byte aufweist, wird durch 6LoWPAN ein Mechanismus zur Paketfragmentierung eingeführt. Außerdem umfasst der Standard verschiedene Mesh-Routing Verfahren.

ZigBee(IP)

ZigBee ist ein Protokoll, das auf IEEE 802.15.4 aufsetzt [A 51]. Es definiert oberhalb der MAC-Schicht eine Vermittlungs-, Security- und Anwendungsschicht. Auf der Vermittlungsschicht werden Nachrichten in einem eigenen Adressraum geroutet. Mit der Spezifikation ”ZigBee IP” werden spezielle (komprimierte) IPv6-Pakete nach dem 6LoWPAN-Standard eingeführt. Außerdem wird ein Schlüsselsystem beschrieben, bei dem ein Vertrauensanker andere Netzwerkteilnehmer (Router und Endgeräte) authentifiziert und mit Schlüsselmaterial versorgt. Um einem Netzwerk beizutreten und verschlüsselt kommunizieren zu können, existieren drei Möglichkeiten. Mittels Pre-shared Key kann ein Endgerät dem Netzwerk beitreten. Dies erschwert es jedoch, Geräte verschiedener Hersteller miteinander zu verbinden. Alternativ beschreibt der ZigBee-Standard eine unverschlüsselte Schlüsselübertra-

gung aus einem Netzwerk an einen neuen Teilnehmer. Die dritte Variante beschreibt einen Mechanismus, bei dem ein Gerät, das dem Netzwerk beitritt, einen Schlüssel besitzt, um eine abgesicherte Verbindung zu einem Vertrauensanker herzustellen. Der Vertrauensanker versorgt den neuen Netzwerkteilnehmer mit Schlüsselmaterial, um innerhalb des Netzwerks abgesichert kommunizieren zu können. ”Abgesichert kommunizieren zu können” bedeutet, dass Nachrichten auf Netzwerk- und Anwendungsschicht verschlüsselt werden können.

Z-Wave

Z-Wave ist ein proprietärer Drahtloskommunikationsstandard, der alle Netzwerkschichten von der physikalischen bis hin zur Anwendungsschicht definiert [A 52]. Durch die verwendeten ISM-Bänder im Sub-1-GHz-Bereich, der geringen Sendeleistung von wenigen Milliwatt und der eingesetzten Modulation werden Übertragungsraten von bis zu 100 kbit/s erzielt. Die Reichweite beträgt ca. 150 m im Freifeld und ca. 40 m im Innenbereich. Nachrichten werden auf Basis eines eigenen Adresssystems geroutet. Um die Interoperabilität zwischen verschiedenen Geräten zu gewährleisten, wird eine einheitliche Anwendungsschicht definiert. Nach dem Z-Wave-Standard zertifizierte Geräte können ohne Inkompatibilitäten miteinander kommunizieren.

Bluetooth

Bluetooth ist ein weit verbreitetes Protokoll um eingebettete Systeme miteinander zu vernetzen [A 53]. Es unterstützt in der aktuellen Version Mesh-Strukturen. Zur Kommunikation wird das freie ISM-Band von 2,402 GHz bis 2,480 GHz genutzt. Die Low Energy-Erweiterung des Standards ist speziell auf die Anforderungen von batteriebetriebenen Geräten zugeschnitten. Außerdem ist es möglich IP-Pakete über Bluetooth zu versenden [A 54].

LoRa(WAN)

LoRa (Long Range) ermöglicht es Kleinstgeräten eine Datenübertragung von mehreren Kilometern zu erzielen [A 55]. Dies wird durch eine schmalbandige Datenübertragung erreicht. Es können verschiedene Modulationsverfahren gewählt werden um zwischen Datenrate und Energieumsatz abzuwägen. Die maximalie Datenrate liegt bei 50 kbit/s. LoRaWAN beschreibt eine Infrastruktur, bei der IoT-Geräte wie Sensorknoten Daten an ein Gateway senden. Das Gateway kommuniziert IP-basiert über

ein WAN (Wide Area Network) mit einem Anwendungsserver. Der Anwendungsserver kann u.a. Messdaten visualisieren oder als Nutzer-Interface dienen, um Steuerbefehle an IoT-Geräte zu senden.

LTE-M

Der Mobilfunkstandard LTE-M [A 56] ermöglicht es IoT-Geräten energiesparend Daten in einem Wide Area Network zu übertragen. Es werden je nach Bandbreite (1,4 bis 20 MHz) bis zu 7 Mbit/s übertragen. Jedoch müssen sich die Endgeräte bei einem Provider anmelden. Die verwendeten Frequenzbereiche sind exklusiv für Netzwerkanbieter reserviert.

NB-IoT

NB-IoT (Narrowband Internet of Things) [A 57] steht im direkten Vergleich zu LTE-M. Die verwendete Bandbreite ist mit 180 kHz deutlich geringer. Daher sind geringere Übertragungsraten von bis zu 159 kbit/s möglich. NB-IoT-Netwerke müssen jedoch von einem Mobilfunkanbieter bereitgestellt werden.

3.6 Zwischenfazit

In den vergangenen Jahren wurden eingebettete Systeme leistungsstärker und Kommunikationsprotokolle effizienter. Microcontroller-Hardware wie der ESP32 sind in der Lage, über WLAN eine IP-basierte Verbindung zu anderen Geräten aufzubauen [A 58]. Während der Entwicklung des Protokolls CoAP wurden besonders kleine Nachrichtengrößen durch einen effizient codierten Header vorgesehen. CoAP vereint alle Features von HTTP und MQTT, sodass es das geeignetste Protokoll für die M2M-Kommunikation darstellt. RESTful-Webdienste stellen die beste Variante dar, Daten und Funktionen eines Gerätes bereitzustellen, da sie eine herstellerübergreifende Kommunikation zwischen Client und Server ermöglichen. CoAP lässt sich durch die Verwendung eines Brokers an HTTP-basierte Webdienste des Internets anbinden. Mit OSCORE existiert ein Protokoll, mit dem eine Ende-zu-Ende-Sicherheit über Broker hinweg realisiert werden kann. Alternativ ist eine Ende-zu-Ende-Sicherheit zwischen CoAP-basierten Geräten mittels DTLS möglich. In Abbildung 3.1 ist der Referenzstack (CoAP über WLAN Mesh) abgebildet. Er stellt die Grundlage einer modernen GA-Systemarchitektur dar. CoAP über UDP bzw.

OSCORE über DTLS zu übertragen stellt eine Variante dar. Alternativ (Variante II) lassen sich CoAP bzw. OSCORE über 6LoWPAN übertragen.

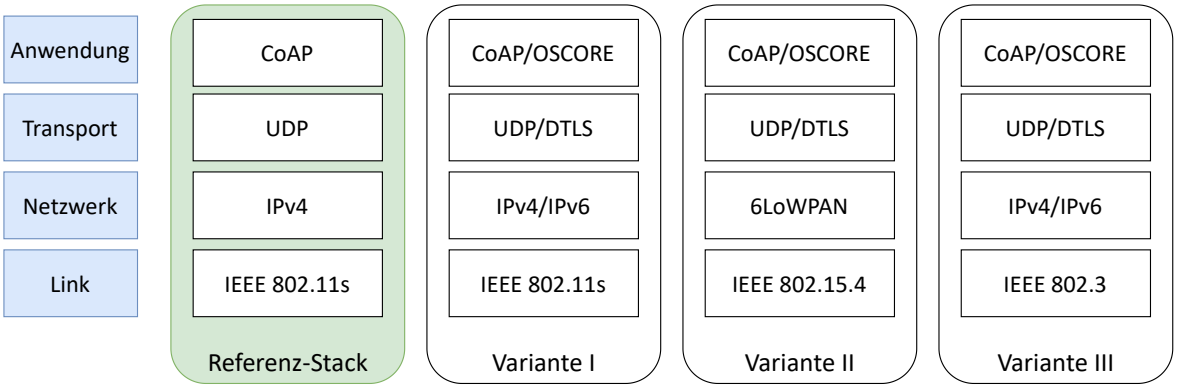

Abbildung 3.1: CoAP-Referenz-Stack

Die IEEE 802.11s Mesh-Erweiterung bietet die größten Vorteile. Dadurch, dass die einzelnen Netzwerkteilnehmer selbst als Infrastrukturknoten auftreten, entfällt der Installationsaufwand für zusätzliche Access Points. Das dynamische Routing kann zur Laufzeit auftretenden Störungen begegnen und die Frames über alternative Routen weiterleiten. Z-Wave und alle 802.15.4-basierten Varianten legen den Fokus auf Energieeffizienz. Die erzielten Datenraten sind verglichen mit WLAN sehr gering. Alternativ ist die IPv6-basierte Übertragung von CoAP und OSCORE Nachrichten über 6LoWPAN möglich. Ethernet nach IEEE 802.3 [A 59] kann ebenso genutzt werden, um CoAP-Nachrichten auszutauschen.

CoAP ist bestens geeignet, kleine Payloads bei geringem Overhead zwischen Geräten auszutauschen. In der Gebäudeautomation sind außerdem Audio-/Videostreaming-Anwendungen zu Überwachungszwecken von großer Bedeutung. Um alle Anwendungsmuster von sporadischer Sensordatenübertragung bis zu Übertragungen mit hohen Bitraten mit nur einem Protokoll zu realisieren, wurde die Eignung von CoAP für hohe Bitraten in der eigenen Publikation [B 2] untersucht. Wenn sich alle Anwendungen einer GA mit nur einem Protokoll implementieren lassen, reduziert sich die Komplexität des Gesamtsystems deutlich. Die große Angriffsfläche durch viele verschiedene Protokolle, deren Implementierungen Verwundbarkeiten aufweisen können, ließe sich reduzieren.

4 Performance-Validierung der Protokollauswahl anhand einer Streaming-Anwendung

Dieses Kapitel widmet sich der Fragestellung, ob sich CoAP nicht nur für sporadische Datenübertragungen von Messwerten und Steuerbefehlen eignet. Wenn mit CoAP auch Anwendungsmuster mit hoher Datenrate übertragen werden können, lassen sich sogar Streaming-Anwendungen realisieren. Die nachfolgenden Untersuchungen beinhalten die in [B 2] publizierten Ergebnisse.

4.1 Motivation

Innerhalb eines Gebäudeautomationsnetzwerks tauschen Geräte einzelne Nachrichten aus, um Sensordaten abzurufen bzw. asynchron zu empfangen und Steuerbefehle an Aktoren zu senden. Einzelne Kommunikationsflüsse zwischen zwei beliebigen Geräten sind sporadischer bzw. periodischer Natur. Charakteristisch für diesen Datenverkehr sind niedrige Übertragungsraten. Einen weiteren Anwendungsfall in der Gebäudeautomation stellt das Streaming von Audio/Video-Daten dar. Kameras zur Überwachung bzw. zur Personenerkennung an einer Gegensprechanlage senden Audio- und Video-Datenströme an Indoor-Bedienpaneele oder Mobilgeräte. Dabei treten hohe Datenraten auf, die durch das lokale Netzwerk zwischen Datenquelle und -senke übertragen werden müssen. Dabei ist eine niedrige Latenz im Bereich von weniger als 100 ms zu erzielen um Livedaten zu erhalten. Insbesondere im Falle einer Gegensprechanlage sind geringe Übertragungslatenzen von großer Bedeutung.

In diesem Abschnitt sollen verschiedene Protokolle und Standards vor dem Hintergrund einer WLAN-basierten Netzwerkinfrastruktur verglichen werden. Wenn ein technisches System aus möglichst wenigen Teilkomponenten besteht, lässt sich die Angriffsfläche verkleinern. Jede zusätzliche Komplexität des Systems erhöht die Anzahl potentieller Schwachstellen. Da CoAP alle Anforderungen eines Gebäudeautomationssystems für Anwendungen mit niedrigen Datenraten erfüllt und die Kompatibilität zu Web-Diensten (HTTP-basierte Cloud APIs) gewährleistet, ist der Einsatz für hochbitratige Anwendungen zu untersuchen. Ursprünglich dient CoAP zum Austausch von kleinen Datenmengen, die in eine einzelne Nachricht passen, ohne zu einer Fragmentierung auf IP- oder MAC-Schicht zu führen. Jedoch wurden

bei der Standardisierung Anwendungsfälle berücksichtigt, in denen ein Server via CoAP block-wise Transfer [A 60] große Datenmengen an einen Client sendet. Dabei wird die zu versendende Payload in einzelne Blöcke unterteilt und stückchenweise übertragen. Der Client setzt auf Basis der Header-Informationen der einzelnen Nachrichten das ursprüngliche Datum zusammen. Um einen kontinuierlichen Datenstrom zwischen zwei Geräten zu übertragen, existieren zwei Verfahren: Nach dem Request/Response-Schema ruft der Client die einzelnen Datenströme per Anfrage ab. Der Server liefert den in Blöcken segmentierten Datenstrom an den Client per block-wise-Transfer aus. Alternativ können Datenströme über den asynchronen Observe-Mechanismus angefordert und übertragen werden. Ein CoAP-Client sendet einen GET-Request mit dem sogenannten Observe-Flag und trägt sich in die Liste der Subscriber (dt.: Abonnenten) ein. Sobald genügend Daten gepuffert wurden und die maximale Payload-Größe einer einzelnen CoAP-Nachricht erreicht ist, wird von der Streaming-Ressource eine Nachricht an jeden Abonnenten per Unicast versendet. Das erste Verfahren nach dem Request/Response-Schema wurde unter dem Namen DASCo in [A 45] untersucht. Die Namensgebung geht auf die Verwandtschaft zu MPEG Dash [A 61] zurück. MPEG Dash beschreibt ein modernes Verfahren zum Videostreaming im Internet, bei dem ein Video über HTTP an einen Client gesendet wird. Das Videomaterial wird vom Server in Abschnitte von wenigen Sekunden segmentiert und über eine RESTful API zur Verfügung gestellt. Der Client fordert per HTTP-Request die einzelnen Segmente nacheinander an. Ein MPEG Dash Server bietet typischerweise den Videostream in unterschiedlichen Qualitätsstufen an. Jede Qualitätsstufe wird über eine dedizierte URL repräsentiert. Um die Nutzererfahrung zu optimieren, rufen MPEG Dash Clients während des Starts eines Streams die niedrigste Qualitätsstufe ab. Dadurch startet das Rendern des Streams für den Nutzer unmittelbar nach Aufruf, da nur eine kleinere Datenmenge bewältigt werden muss. Gemäß der Netzwerkbandbreite werden im zeitlichen Verlauf andere URLs abgerufen, die höhere Auflösungen und Bitraten liefern. Die Größe der Chunks orientiert sich an der Frequenz der Referenzframes des jeweiligen Videokompressionsverfahrens. Aktuelle Verfahren wie H.264 und H.265 codieren Videodaten durch einzelne Frames, die in hoher Qualitätsstufe mit einer relativ großen Datenmenge codiert werden. Aufgrund der Ähnlichkeit zwischen aufeinanderfolgenden Frames werden vom Kompressionsverfahren lediglich die Änderungen codiert. Dies führt zu einer Reduktion der Datenrate. Der Decoder eines entsprechenden Videostreams kann bei Fehlen des Referenzframes das Videomaterial nicht anhand

der nachfolgenden Frames korrekt decodieren und rendern. Daher ist ein Umschalten zwischen verschiedenen, mit H.264 [A 62] bzw. H.265 [A 63] codierten, Streams nur zu jedem Referenzframe möglich. DASCo [A 45] erbt dieselben Eigenschaften und Features von MPEG Dash. Durch den Einsatz von CoAP fällt die TCP-basierte Flusskontrolle von HTTP weg. Die Autoren von [A 45] haben die Flusskontrolle auf der Anwendungsschicht von CoAP mit der TCP-Standardkonfiguration verglichen. Es konnte gezeigt werden, dass die CoAP-Flusskontrolle in WLAN-Szenarien der Flusskontrolle von TCP überlegen ist. Die typische WLAN Fehlercharakteristik führt zu Nachrichtenverlusten. Das CoAP-Neuübertragungsschema kann in WLAN-basierten Netzwerken Nachrichtenverluste effizienter kompensieren.

Das CoAP-basierte Verfahren nach [A 64] verwendet den Observe-Mechanismus. Dabei bietet der Server das Datenmaterial über eine Ressource an, die als ”observable” gekennzeichnet ist. Sobald genügend Streaming-Daten für eine CoaP-Payload vorliegen, wird die Nachricht an jeden Client, der als Ressourcen-Abonnent gespeichert wurde, versendet. Das Verfahren nach [A 64] dient als Grundlage für eigene Implementierungen in den Sprachen C und Java. Innerhalb eines Testaufbaus wurde das Datenstreaming evaluiert und in der eigenen Publikation [B 2] veröffentlicht. Die Untersuchungen dienen dazu, die technische Machbarkeit auf eingebetteten Systemen zu untersuchen. Dabei wurden verschiedene State-of-the-Art CoAP-Implementierungen miteinander verglichen. Da sich in der Literatur [A 45] Experimente zum Übertragungsverhalten von CoAP-Nachrichten über WLAN finden, wurde das Laufzeitverhalten der Endgeräte durch eigene Experimente über Ethernet näher untersucht.

4.2 Szenario

Die CoAP-Observe-basierte Datenübertragung nach [A 64] (Abbildung 4.1) wurde mit verschiedenen CoAP-Implementierungen realisiert. Java-Implementierungen zeichnen sich durch eine hohe Portabilität aus, da die Java-Anwendung auf einer entsprechenden JVM (Java Virtual Machine) auf der Zielplattform ausgeführt wird. JVMs sind sowohl als quelloffene als auch proprietäre Implementierungen für verschiedene Hardware-Plattformen (ARM, x86) verfügbar. Einplatinencomputer der Leistungsklasse Raspberry Pi Zero oder Raspberry Pi 1 [A 65] sind bereits in der Lage, Java-Anwendungen im Embedded-Bereich auszuführen. Die

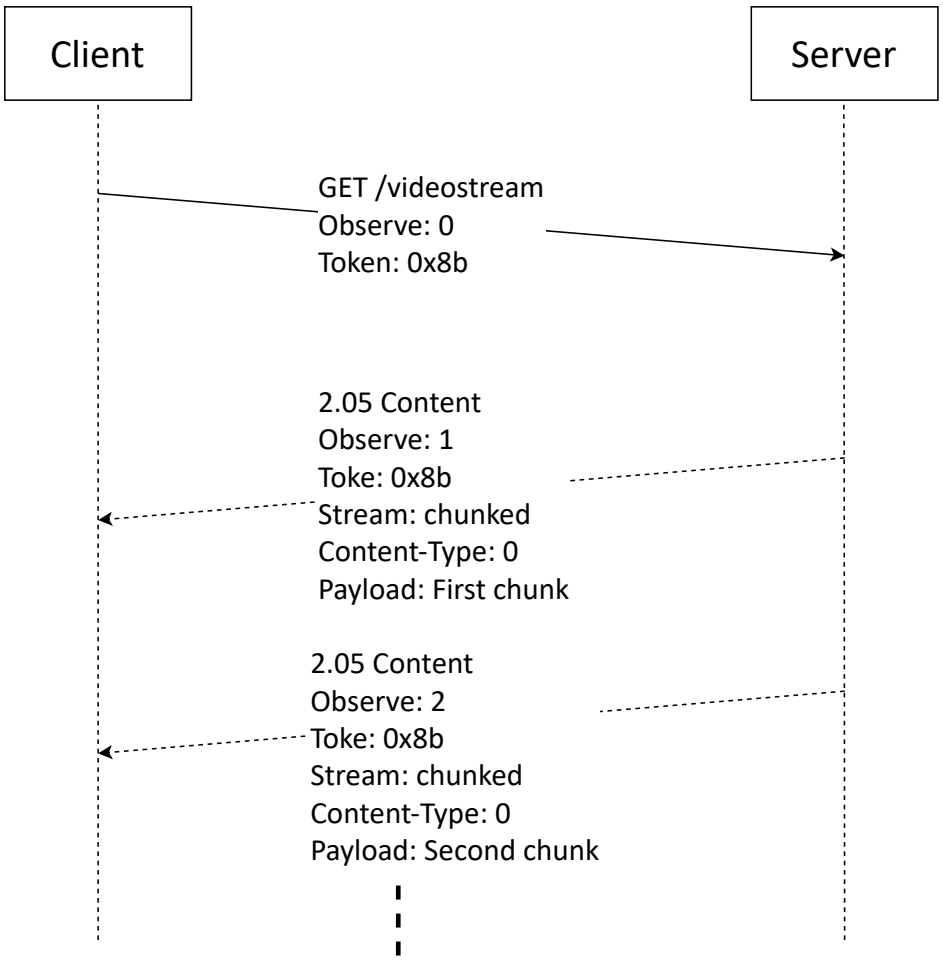

Abbildung 4.1: Abruf und Übertragung eines Videostreams mittels Observe-Mechanismus nach [A 64]

Abstraktionsschicht der JVM verringert den Portierungsaufwand, sodass Java-Implementierungen von Protokollen und Anwendungen insbesondere für Prototypenentwicklungen von großer Bedeutung sind. Californium ist eine quelloffene Implementierung der Eclipse Foundation in Java [A 66]. Sie wurde genutzt, um einen CoAP-Streaming-Server und Client zu programmieren. Zum direkten Vergleich wurden Server und Client mit jCoAP [A 67], einer weiteren CoAP-Java-Implementierung, umgesetzt. Der Speicher einer Java-Anwendung wird vom Garbage Collector (GC) der JVM verwaltet. Der GC gibt u.a. nicht mehr benötigten Heap-Speicher frei, um die Arbeitsspeicher zu bereinigen. Währenddessen wird die Ausführung der Anwendung angehalten. Da die Aktivität des GC sporadischer Natur ist, sind Java-Anwendungen, die nicht auf speziellen JVMs ausgeführt wer-

den, nicht echtzeitfähig. Jene speziellen JVMs unterstützen einen zeitlich vorhersehbaren und auch konfigurierbaren GC, um limitierte Unterbrechungszeiten zu gewährleisten. Bei Java-basierten Live-Streaming-Anwendungen wird die Latenz zwischen Datenquelle und -senke maßgeblich durch den GC beeinflusst [A 68]. Daher sollen im nachfolgenden Experiment Messwerte für die Verarbeitungsdauer von CoAP-Nachrichten aufgenommen werden, um die Eignung für Live-Streaming-Anwendungen mit einer maximalen Latenz von wenigen 10 ms zu untersuchen. Um die einzelnen Java-Anwendungen miteinander vergleichen zu können, wurde dieselbe Java-Laufzeitumgebung (JRE 1.8 build 171) verwendet. Im Vergleich zu den Java-Implementierungen wurde das CoAP-basierte Streaming als nativer Code für die Zielplattform compiliert. Als CoAP-Implementierung wurde das weit verbreitete libcoap [A 69] verwendet.

Gemäß Abbildung 4.2 dient ein in der Programmiersprache C geschriebenes Programm als Generator von Testdaten (Programmname: "Testdata"). Mit Hilfe dieses Programms ist es möglich, definierte Payload-Größen mit bestimmter Rate zu erzeugen und über den Standard-Datenstrom (stdin) in den CoAP-Server zu leiten. Der CoAP-Server nimmt die Testdaten entgegen und aktualisiert nach Empfang eines Steuerzeichens (Zeilenumbruch) den Status der Streaming-Ressource. Zu Beginn jedes Experiments sendet der CoAP-Client eine GET-Nachricht an die Streaming-Ressource des Servers (Abbildung 4.1). Dabei wird das Observe-Bit gesetzt, um den Client Server-seitig als Abonnenten zu registrieren. Sobald die Streaming-Ressource mit Daten aktualisiert wurde, sendet der Server eine entsprechende Notification an den Client. Server und Client werden auf derselben Hardware-Plattform ausgeführt. Je nach Testfall wurden verschiedene Leistungsklassen durch je einen Raspberry-Pi 1, Raspberry-Pi 3 [A 65] und das ZedBoard [A 70] getestet und evaluiert. Server und Client kommunizieren über eine 1 Gbit/s Ethernet-Verbindung, um den Jitter im Vergleich zu drahtloser Kommunikation wie z.B. WLAN zu minimieren. Um das Zeitverhalten der Server- und Client-Anwendung beurteilen zu können, wurden verschiedene Zeitpunkte aufgenommen. Sobald der Server Streamingdaten vorliegen hat, wird die Zeit t_1 aufgezeichnet. Nachdem die CoAP-Nachricht erzeugt und serialisiert wurde, wird unmittelbar vor Absenden über den UDP-Socket der Zeitpunkt t_2 aufgenommen. Zum Zeitpunkt t_3 wurde das Paket erfolgreich abgesendet. Der Zeitpunkt t_4 markiert den Client-seitigen Empfang der Nachricht. Das Parsen der Nachricht ist zum Zeitpunkt t_5 abgeschlossen.

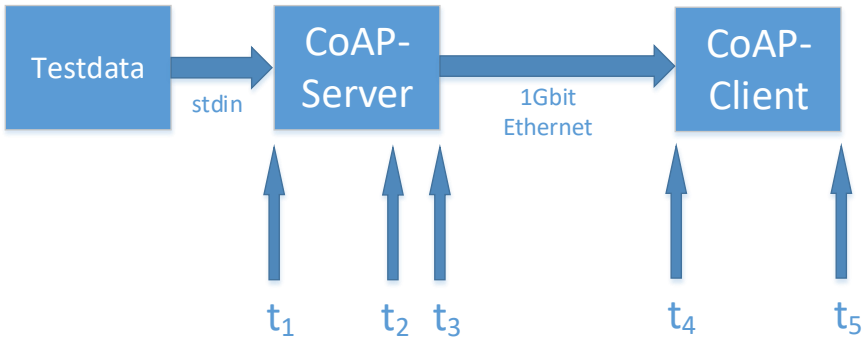

Abbildung 4.2: Schematischer Testaufbau bestehend aus Datenquelle (”Testdata” und CoAP-Server) und Datensenke (CoAP-Client) [B 2]

4.3 Ergebnisse und Evaluation

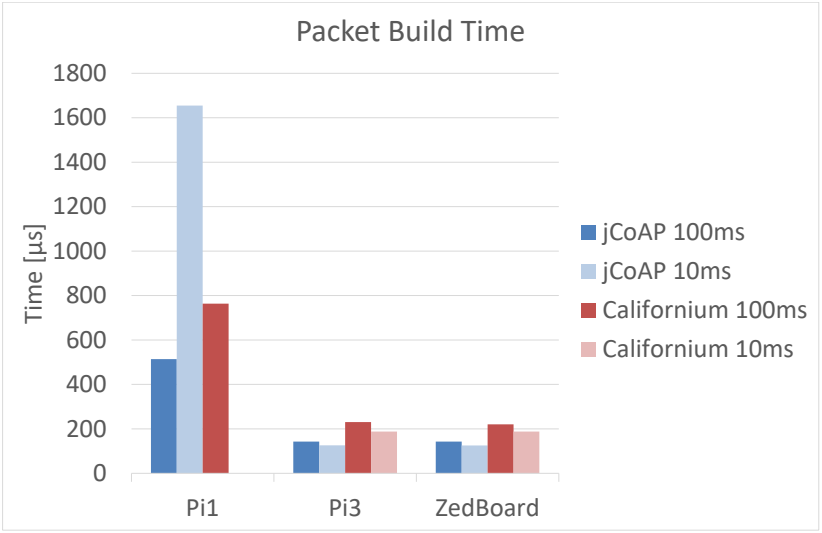

Abbildung 4.3: Zeitdauer t_2 - t_1, um eine Nachricht zu erzeugen [B 2]

Im ersten Testfall werden Nachrichten fester Payload-Größe im Intervall von 100 ms bzw. 10 ms vom CoAP-Server versendet. Jede Messung terminiert nach 10.000 Nachrichten. Dabei werden die Verarbeitungszeiten t_2 - t_1 zum Erzeugen einer Nachricht, die Dauer des UDP-Socketaufrufs des Servers (t_3 - t_2) und die Verarbeitungsdauer zum Parsen auf Client-Seite (t_5 - t_4) aufgenommen. Abbildung 4.3 stellt die durchschnittlichen Zeiten zur Erzeugung einer CoAP-Notification des Servers

auf verschiedenen Hardwareplattformen dar. Die Ausführungszeiten des ZedBoards und des Raspberry Pi 3 sind vergleichbar. Auf beiden Plattformen beträgt die Zeit zum Bauen einer Nachricht weniger als 200 μs. Außerdem ist erkennbar, dass die Zeiten von jCoAP kürzer als die Zeiten von Californium sind. Wird die Datenrate um den Faktor 10 erhöht, verringert sich die Ausführungszeit leicht. Der Raspberry Pi 1 ist die leistungsärmste Plattform des Versuchsaufbaus. Dies zeigt sich an deutlich längeren Zeiten, um eine CoAP-Nachricht zu erstellen. Bei Erhöhung der Datenrate war ein Sättigungseffekt durch eine massiv erhöhte Ausführungszeit erkennbar. Im Falle von jCoAP vergrößerte sich bei 10 ms Refresh-Intervall die Zeit um mehr als Faktor 3 von ca. 500 μs auf gut 1600 μs. Wurde der Californium CoAP-Server auf dem Raspberry Pi 1 mit 10 ms Refresh-Intervall ausgeführt, waren keine sinnvollen Messwerte bestimmbar. Derselbe Effekt zeigte sich ebenfalls, wenn Californium mit 1 ms Refresh-Intervall auf dem Raspberry Pi 3 ausgeführt wurde. Abbildung 4.4 zeigt den Anstieg der Ausführungszeit für jede einzelne CoAP-Benachrichtigung. Von wenigen 100 μs bei den ersten Nachrichten steigt die Ausführungszeit monoton bis auf über 250 ms bei den letzten Nachrichten. Mithilfe der Remote-Debugging Tools von Java wurde eine deutlich erhöhte Aktivität des GC ermittelt. Die jeweilige Java-Anwendung legt für jedes einzelne Paket viele Objekte an. Nach einer näheren Analyse werden in jCoAP weniger Objekte als in Californium für das Versenden einer Nachricht benötigt. Mit jedem Anlegen eines Objektes geht jeweils eine Speicherallokation auf dem Heap einher. Die jeweiligen Objekte haben nur eine begrenzte Lebensdauer, da sie nach dem Versenden der CoAP-Nachricht nicht mehr referenziert werden und somit der GC aktiv wird, um den Speicher freizugeben. Ab einer bestimmten Datenrate, die von der Performance der Hardwareplattform abhängig ist, kommt es zu einer Überlastsituation der Managementfunktionalitäten, die sich im Ansteigen der Paket-Verarbeitungszeit äußert.

Eine weitere untersuchte Zeitkomponente ist die Aufrufzeit des UDP-Sockets aus der JVM heraus. Nachdem der CoAP-Server dem Socket die serialisierte Nachricht per Argument übergeben hat, werden weitere Subroutinen des Betriebssystems aufgerufen, um die Daten in Form eines UDP-Datagrams über das Netzwerkinterface (1 GBit/s Ethernet) zu versenden. Die ermittelten Zeiten sind in Abbildung 4.5 dargestellt. Der Raspberry Pi 1 weist im Gegensatz zum Raspberry Pi 3 und dem ZedBoard eine deutlich höhere Zeit auf. Generell liegen die Zeiten unterhalb der Verarbeitungszeit, um die eingehenden Streamingdaten in CoAP-Nachrichten

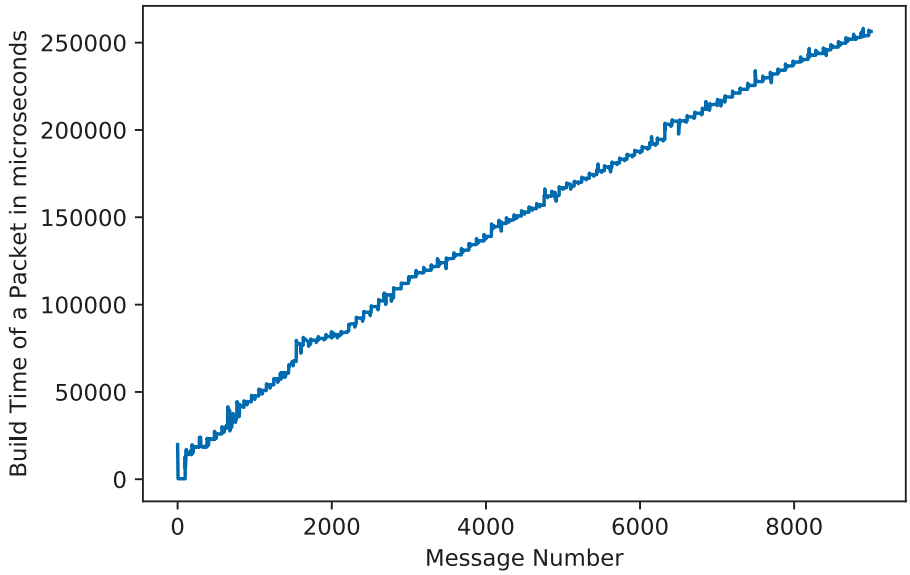

Abbildung 4.4: Anstieg der Ausführungszeit beim Erzeugen einer CoAP-Nachricht durch den Server (Californium auf Raspberry Pi 3) [B 2]

zu verpacken. Nachdem eine Nachricht über den Socket versendet wurde, wird von jCoAP bzw. Californium die nächste Nachricht aus dem Sendepuffer gelesen und an den Socket übergeben. Somit lässt sich eine theoretische Grenze der minimalen Periode zwischen zwei Nachrichten und damit die maximale theoretische Übertragungsrate ermitteln. Für das ZedBoard und den Raspberry Pi 3 beträgt diese Grenze unabhängig von jCoAP oder Californium ca. 20.000 Nachrichten/s bzw. 20 MB/s. Der Raspberry Pi 1 erzielt theoretisch 2800 Nachrichten/s und 2,8 MB/s für jCoAP und ca. 4000 Nachrichten/s und 4 MB/s für Californium. Ein genauer Grund für die Diskrepanz zwischen jCoAP und Californium auf dem Raspberry Pi 1 konnte nicht ermittelt werden. Durch die Begrenzungen der JVM, die Sendezyklen von kleiner als 1 ms nicht zulassen, werden die theoretischen Maximalwerte nicht erreicht.

Nachdem der Client-UDP-Socket ein eingehendes Datagramm an die Java-Anwendung weiterleitet, wird die CoAP-Nachricht geparsed. Das Zeitintervall wird durch die Zeitpunkte t_4 und t_5 begrenzt. Abbildung 4.6 visualisiert die Verarbeitungszeiten in Abhängigkeit der Plattform, der verwendeten CoAP-Implementierung und des Refresh-Intervalls. Der Performance-Unterschied zwischen Raspberry Pi 3 bzw. ZedBoard zum Raspberry Pi 1 ist deutlich erkennbar. Aufgrund dessen, dass Californium als Server nicht mit einem Refresh-Intervall von 10 ms auf dem Raspberry

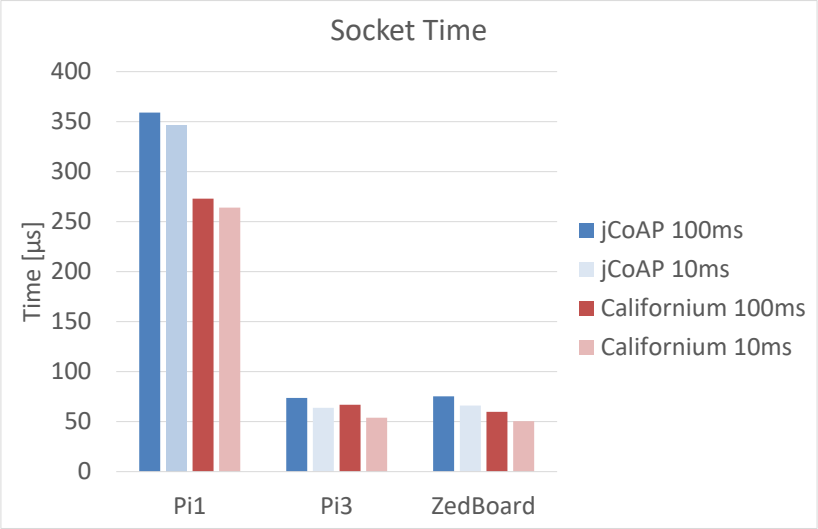

Abbildung 4.5: Dauer bis ein Datagram über den UDP Socket versendet wurde, sodass die aufrufende Routine weitere Datagramme an den Socket reichen kann [B 2]

Pi 1 ausgeführt werden konnte, liegt kein Ergebnis für die Client-Anwendung vor. Unabhängig von der Hardware-Plattform ist erkennbar, dass die Californium Client-Anwendung eine langsamere Nachrichtenverarbeitungszeit aufweist als die jCoAP Client-Anwendung.

Um den Einfluss der JVM auf das Zeitverhalten der Anwendung besser zu analysieren, wurde die Streaming-Anwendung mit der CoAP C-Implementierung libcoap [A 69] implementiert. Die durchschnittlichen Ausführungszeiten der Generierung einer CoAP-Nachricht sind in Abbildung 4.7 dargestellt. Analog zu den vorherigen Messwerten der Java-Anwendungen weisen der Raspberry Pi 3 und das ZedBoard eine ähnliche Performance hinsichtlich der Erstellung einer CoAP-Nachricht auf. Jede Nachricht wurde im Durchschnitt in weniger als 4 μs erzeugt und serialisiert. Der Raspberry Pi 1 benötigt für diese Aufgabe ca. 14 μs.

Abbildung 4.8 visualisiert die mittlere Verarbeitungszeit einer eingehenden CoAP-Nachricht auf Client-Seite. Auf dem Raspberry Pi 3 und auf dem ZedBoard benötigt die Logik der libcoap-Implementierung rund 12 μs. Dies ist im Vergleich zur Generierung einer Nachricht die 3-fache Zeit. Der Raspberry Pi 1 verarbeitet jede ein-

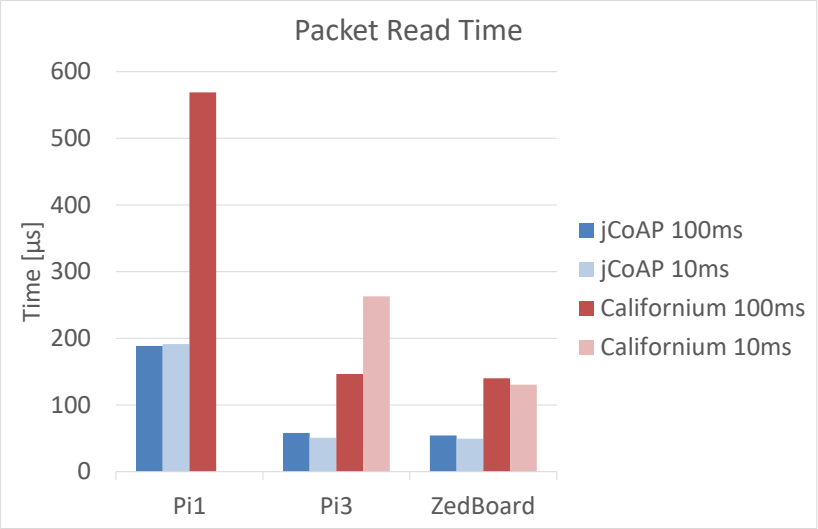

Abbildung 4.6: Client-seitige Parsing-Zeit für eine CoAP-Nachricht [B 2]

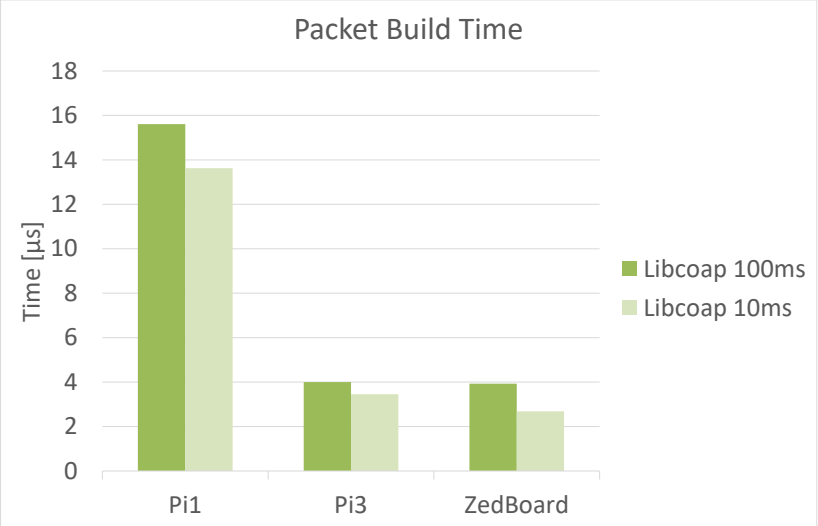

Abbildung 4.7: Dauer der Nachrichtengenerierung unter libcoap [B 2]

gehende Nachricht in ca. 50 μs. Hier ist die Ausführungszeit des Servers 3,5-mal höher als die Ausführungszeit des Clients.

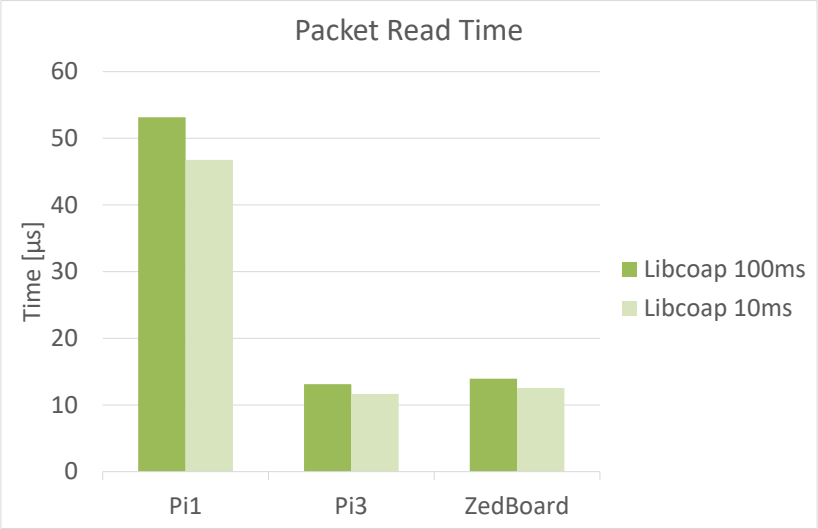

Abbildung 4.8: Verarbeitungszeit einer eingehenden Nachricht unter libcoap [B 2]

Die Summen der durchschnittlichen Ausführungszeiten werden in der Tabelle 4.1 für ein 100 ms Refresh-Intervall und in der Tabelle 4.2 für ein 10 ms Refresh-Intervall aufgelistet. Die erhöhten Werte für Californium und jCoAP auf dem Raspberry Pi 1 sind durch oben genannte Überlastungseffekte erklärbar. Insbesondere durch die Performance-Anforderungen von Californium und bei einer Datenrate von 100 kB/s (entsprechen 10 ms Refresh-Intervall) kann ein System der Leistungsklasse des Raspberry Pi 1 den Stream nicht mehr übertragen. Der Raspberry Pi 3 und das ZedBoard sind hinsichtlich der Performance vergleichbar. Auch hier hat jCoAP einen geringeren Performancebedarf als Californium. Die libcoap-Implementierung ist ca. 10-mal so schnell wie jCoAP. Mit der C-Implementierung war es möglich, ein Sendeintervall von 1 ms zu erreichen. Bei einer Payload-größe von 1024 B entspricht dies einer Datenrate von 1 MB/s. Alle Ergebnisse des Raspberry Pi 3 und des ZedBoards lassen auf den ersten Blick eine gute Eignung für Live-Streaming-Anwendungen zu, da die Verarbeitungsdauer in allen Testfällen kleiner als 500 μs ist. Dieser Overhead ist gering im Vergleich zu den Verarbeitungszeiten der Audio-/Videodatencodierung und -decodierung. Bei den in Tabelle 4.1 und 4.2 aufgelisteten Werten handelt es sich um gemittelte Werte. Während der Untersuchungen mit Beispiel-Videomaterial samt Rendering via ffmpeg [A 71] fielen Probleme (stockende Wiedergabe und Arte-

fakte) beim Versand mit den Java-basierten Anwendungen auf. Dabei wurde sowohl MJPEG [A 72] als auch H.264 [A 62] codiertes Material versendet und decodiert.

Tabelle 4.1: Gesamte Verarbeitungsdauer [μs] zwischen Server und Client bei 100 ms Sendeintervall [B 2]

Implementierung	**Raspberry Pi 1**	**Raspberry Pi 3**	**ZedBoard**
jCoAP	702	201	197
Californium	1332	377	360
libcoap	69	17	18

Tabelle 4.2: Gesamte Verarbeitungsdauer [μs] zwischen Server und Client bei 10 ms Sendeintervall [B 2]

Implementierung	**Raspberry Pi 1**	**Raspberry Pi 3**	**ZedBoard**
jCoAP	1846	177	174
Californium	258.929	451	318
libcoap	60	15	15

Eine nähere Untersuchung der Ausführungszeiten jeder einzelnen Nachricht offenbart einen großen Jitter. Die nachfolgenden Messungen wurden zur Vergleichbarkeit auf dem ZedBoard ausgeführt. Die Ausführungszeiten der Nachrichtengenerierung durch den Server und der Nachrichtenverarbeitung durch den Client wurden für je 9000 Nachrichten ermittelt und in den Abbildungen 4.9 und 4.10 für Californium bzw. Abbildung 4.11 und 4.12 für jCoAP aufbereitet.

In allen Testfällen ist nach den ersten 1500 Nachrichten ein deutlicher Abfall der Verarbeitungszeit erkennbar. Die kürzeren Verarbeitungszeiten sind durch Laufzeitoptimierungen der JVM erklärbar. Wird ein Codeabschnitt 1500 mal ausgeführt, so wird dieser Abschnitt durch einen Run-time Compiler in einen optimierten Code übersetzt. Außerdem sind in allen Fällen die Ausführungszeiten starken Fluktuationen unterworfen. Die Fluktuationen liegen ca. zwei Dekaden über dem Mittelwert. Aufgrund ihres seltenen Auftretens beeinflussen sie jedoch den Mittelwert kaum. Im direkten Vergleich zwischen jCoAP und Californium weist jCoAP seltenere Fluktuationen auf. Dies liegt an der performanteren Codestruktur von jCoAP, da zur Laufzeit weniger Objekte erzeugt und damit weniger Speicherallokationen samt

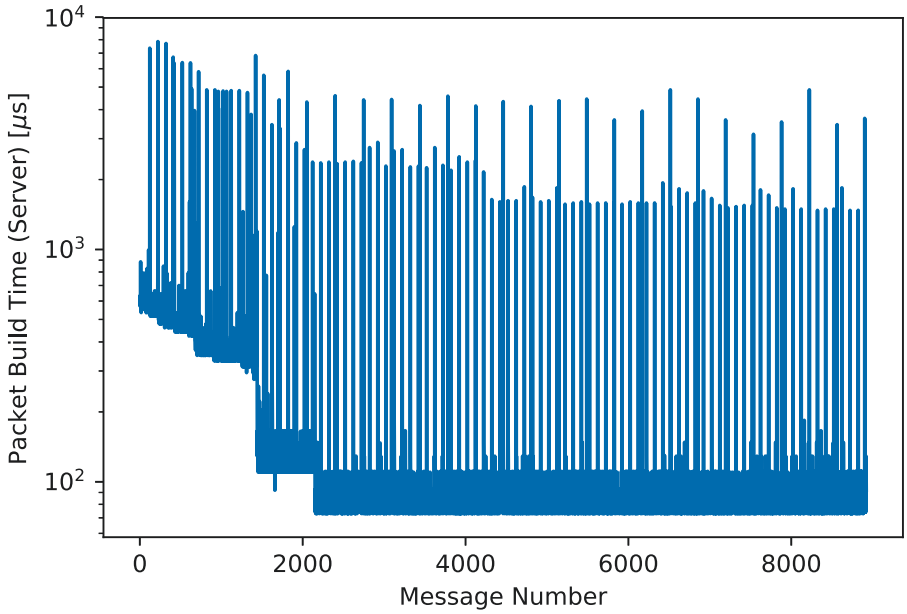

Abbildung 4.9: Zeit, die der Californium-Server benötigt, ein Paket zu generieren (10 ms Refresh-Rate) [B 2]

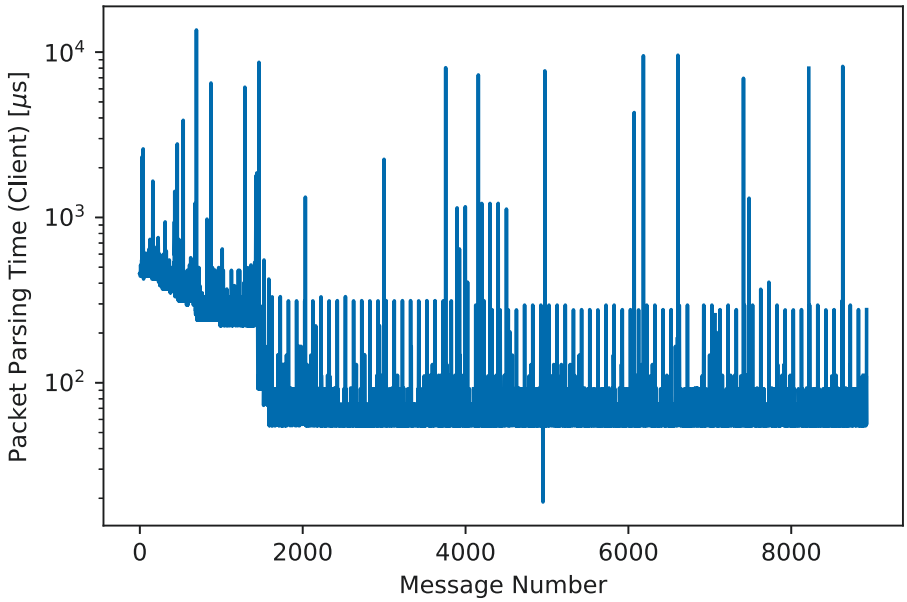

Abbildung 4.10: Zeit, die der Californium-Client benötigt, ein Paket zu parsen (10 ms Refresh-Rate) [B 2]

-freigaben ausgeführt werden müssen. Die VisualVM [A 73] ermöglicht es zur Laufzeit Statusdaten einer JVM abzurufen und graphisch darzustellen. So lassen sich

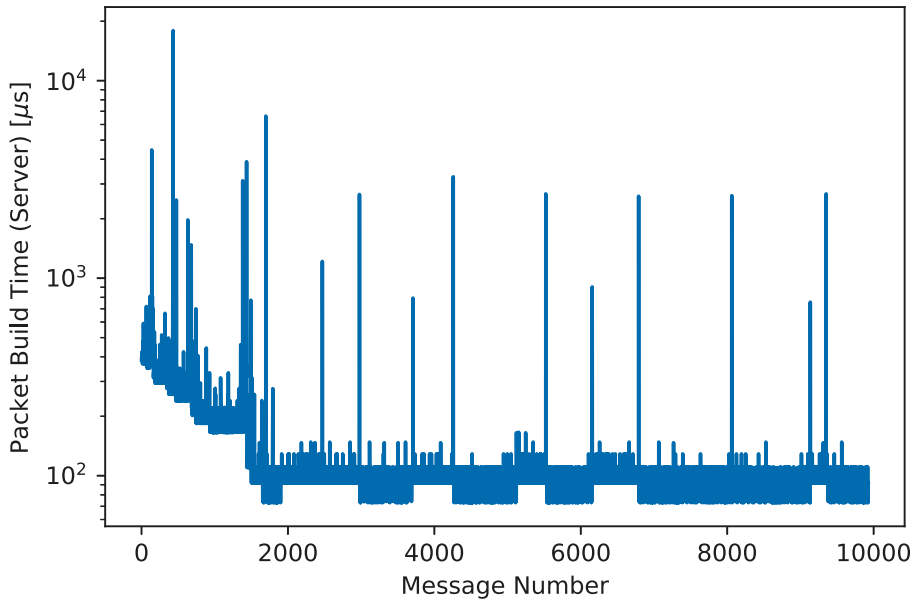

Abbildung 4.11: Zeit, die der jCoAP-Server benötigt, ein Paket zu generieren (10 ms Refresh-Rate) [B 2]

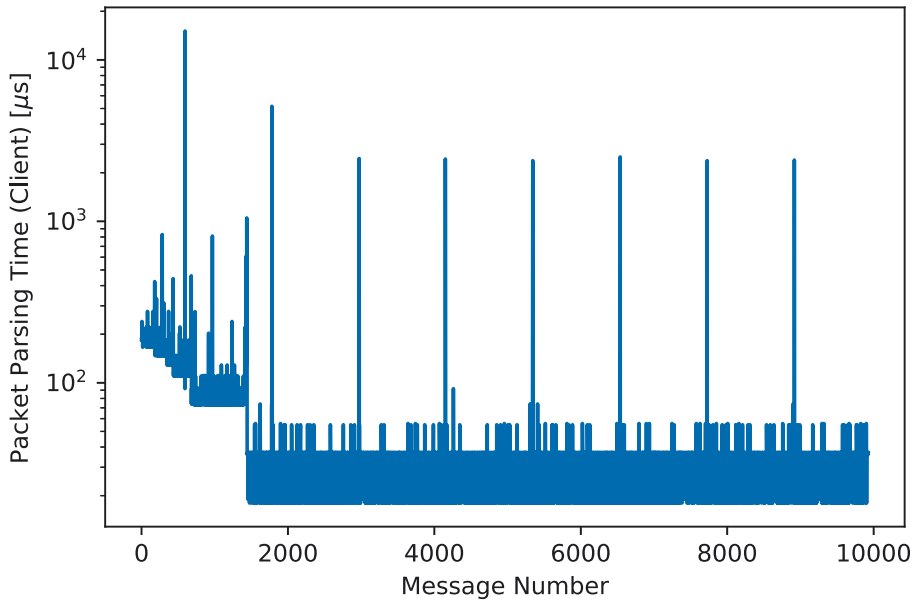

Abbildung 4.12: Zeit, die der jCoAP-Client benötigt, ein Paket zu parsen (10 ms Refresh-Rate) [B 2]

u.a. die Größen des Dynamischen Speichers (engl.: Heap) und die Aktivität des GC beobachten. Der Heap-Speicher wird zur Laufzeit alloziert um Daten, wie z.B.

CoAP-Nachrichten, abzulegen. Werden die Daten nicht mehr benötigt, so werden die Speicherbereiche wieder freigegeben. Eine Analyse mittels der VisualVM [A 73] ergab, dass das Muster der GC-Aktivität mit den Fluktuationen der Nachrichtenverarbeitungszeiten korreliert.

Um Aussagen über die gesamte Latenz und deren Schwankung treffen zu können, wurde auf zwei Raspberry Pi 3 jCoAP als performantere CoAP-Implementierung ausgeführt. Dabei kommuniziert ein CoAP-Client mit einem Echo-Server, der die vom Client gesendete Payload (1024 Byte) zurücksendet. Die vom Client gemessene Round-Trip-Time (RTT) ist in Abbildung 4.13 dargestellt. Die durchschnittliche RTT von 1,7 ms wird von den einzelnen Ausreißerwerten kaum beeinflusst. Von 50.000 RRT-Messungen liegen lediglich 650 oberhalb von 3 ms. Innerhalb von mehr als 10 ms lagen 56 Messungen. Am kritischsten waren 8 Nachrichten (Hin- und Rückrichtung), die länger als 100 ms benötigt haben. Da einige Nachrichten mehrere 100 ms benötigen, müssen die empfangenen Daten mindestens für die maximal mögliche Zeitdauer von RTT/2 gepuffert werden. Im untersuchten Szenario müsste der Empfangspuffer mindestens 300 ms an Audio-Videomaterial puffern, um eine korrekte Decodierung ohne Artefakte zu gewährleisten.

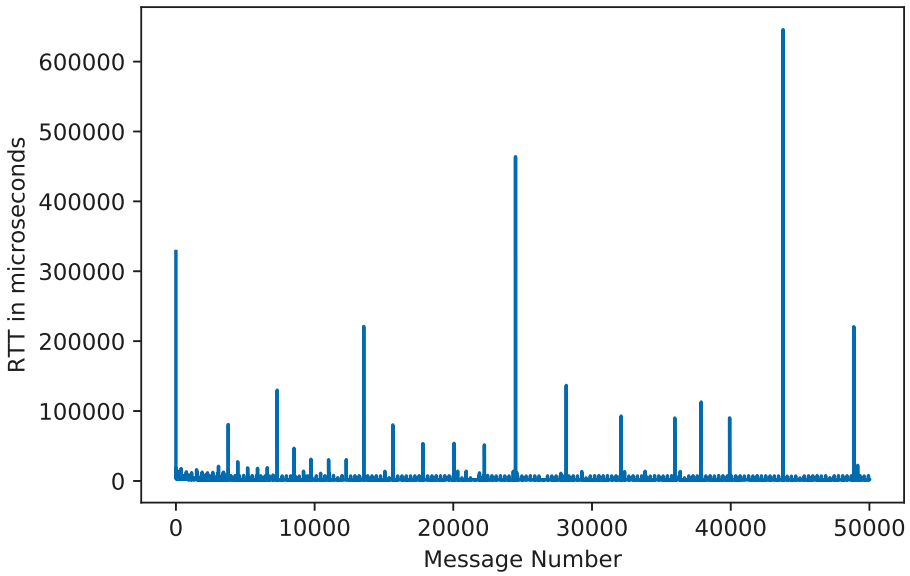

Abbildung 4.13: Round-Trip-Time zwischen jCoAP-Server und -Cient [B 2]

4.4 Zwischenfazit

In diesem Abschnitt wurde CoAP hinsichtlich seiner Eignung für die Übertragung von Streaming-Daten untersucht. Besonderes Augenmerk lag auf der Untersuchung der Nachrichtenverarbeitungslatenz, um den Anwendungsfall einer Audio-/Video-Livestreaming Anwendung zu betrachten. Es ist deutlich erkennbar, dass sich CoAP zum Streamen von Daten eignet. Mit einer C-Implementierung konnte eine Datenrate von 1 MB/s (1024 Byte Payload und 1 ms Sendeintervall) erzielt werden. Die Ausführungszeit, CoAP-Nachrichten zu erzeugen und zu parsen, fällt mit 100 μs sehr gering aus. Mit modernen Videokompressionsverfahren, wie H.264 [A 62] und H.265 [A 62], sind Übertragungen von 24 fps mit 1920 x 1080 Pixeln möglich. Für Anwendungen mit geringeren Anforderungen an den Jitter (kleiner 300 ms), konnten mit einer Java-Implementierung 100 kB/s im Experimentalaufbau zuverlässig erreicht werden. Die beiden typischen Anwendungsmuster (sporadischer Datenaustausch von Sensorwerten und Steuerbefehlen sowie Datenstreaming mit hoher Datenrate und niedriger Latenz) lassen sich mit Hilfe von CoAP implementieren. Das Resultat ist ein System, das aus wenig verschiedenen Protokollen und damit Teilkomponenten besteht und somit eine verbesserte Sicherheit (Safety und Security) aufweist. Der ausgearbeitete Protokollstapel (Abbildung 3.1) dient als Grundlage für die Gerätekommunikation innerhalb der GA. Anhand dieser zukunftssicheren Referenz-GA werden Security-Probleme identifiziert.

5 Bedrohungs- und Anforderungsanalyse

Zunächst werden die Fähigkeiten eines Angreifers basierend auf anerkannten wissenschaftlichen Analysen beschrieben. Da die Anwendungsschicht der GA auf Webtechnologien beruht, liegt der besondere Fokus darauf. Dabei werden Schlüsse gezogen, inwiefern die größten Bedrohungen auf Webanwendungen auf Internet-der-Dinge-Szenarien übertragbar sind. Abschließend werden Anforderungen an eine Security-Architektur für GA-Systeme abgeleitet.

5.1 Angreifermodell

Damit ein Sicherheitskonzept sinnvoll geplant werden kann, müssen die Fähigkeiten und die Arbeitsweise von Angreifern analysiert werden. Oftmals werden allgemeingültige Modelle verwendet, um für spezielle Anwendungsfälle ein abgeleitetes Angreifermodell zu erstellen. Da bislang in der Literatur keine einheitlichen Angreifermodelle, die speziell für die Gebäudeautomation gelten, existieren, dienen allgemeine Modelle zur Analyse des Angreifers [A 74]. Das sogenannte Dolev-Yao-Modell [A 75], benannt nach seinen Entwicklern, gilt als ein annerkanntes Angreifermodell, dass sich auf die Internetkommunikation bezieht. Analog zum Internet ist der Angreifer Teil des Netzwerks. Er ist in der Lage, Nachrichten zu lesen und zu schreiben. Ohne jegliche Security-Mechanismen können die Schutzziele Vertraulichkeit, Integrität und Authentizität nicht gewährleistet werden. Dem Angreifer werden somit sehr weitgehende Möglichkeiten zugeschrieben. Jedoch ist es ihm nicht möglich, kryptographische Verfahren zu brechen und die fehlende Kenntnis über notwendiges Schlüsselmaterial zu umgehen.

In [A 76] wird das Modell nach Dolev-Yao um die Bedrohung durch einzelne kompromittierte Geräte des Internet-der-Dinge erweitert. Bei einer Vielzahl von Geräten, die sich oftmals in öffentlich zugänglichen Bereichen befinden, steigt die Gefahr, dass ein Angreifer durch physischen Zugang an geheimes Schlüsselmaterial gelangt. Dadurch erhöht sich die Mächtigkeit, da sich der Angreifer als vertrauenswürdiges Gerät mit passendem Schlüsselmaterial ausgeben kann.

In [A 77] betrachten die Autoren die Tatsache, dass Geräte entsorgt, verkauft oder neu kommissioniert werden. Dadurch ist es möglich, dass Dritte an vertrauliches Schlüsselmaterial gelangen, bzw. die Firmware manipulieren und eigenen Code

ausführen. Die Kosten für einen solchen Angriff gelten als gering, während die Mächtigkeit eines Angreifers deutlich steigt.

Die Autoren von [A 78] betrachten die Security-Implikationen verschiedener Anwendungsstrukturen. Anwendungen können sowohl zentralisiert als auch dezentralisiert organisiert sein. Beim zentralisierten Ansatz steuert ein zentraler Controller sämtliche Funktionalitäten. Dadurch haben Angreifer ein großes Interesse, den zentralen Knoten zu kompromittieren um möglichst viele sensible Daten zu lesen bzw. steuernd auf Prozesse zuzugreifen. Daher muss der zentrale Controller besonders gegen Attacken gehärtet sein. Dezentrale Steuerungsstrukturen weisen im Gegensatz dazu einen größeren Angriffsvektor auf, da mehrere Geräte als Steuerelemente fungieren. Jedoch ist der Schaden durch kompromittierte vertrauliche Daten und Funktionsausfall vergleichsweise gering. Damit eine dezentral organisierte Anwendungsschicht gelingen kann, muss die Umsetzung besonders nutzerfreundlich sein.

5.2 Angriffe auf Webanwendungen und ihre Übertragbarkeit auf Gebäudeautomation

Als Alternative zu herkömmlichen Gebäudeautomationsprotokollen wie KNX und BACnet eignen sich Webtechnologien. Insbesondere eignet sich CoAP durch den geringen Nachrichtenoverhead für die M2M-Kommunikation. Durch die große Ähnlichkeit von CoAP RESTful APIs zu HTTP(S)-basierten Webanwendungen lassen sich die Sicherheitsbetrachtungen übertragen. Das Open Web Application Security Project (OWASP) publiziert regelmäßig generelle Sicherheitsanalysen von Webapplikationen. Ein anerkanntes Projekt ist die OWASP Top 10 ”The Ten Most Critical Web Application Security Risks” in der aktuellen Fassung aus dem Jahr 2017 [A 79]. Darin werden die häufigsten und kritischsten Security-Risiken aufgeführt. Ein Angriffsvektor zielt auf Sicherheitsschwachstellen ab, was technische Auswirkungen und einen Einfluss auf das Geschäft des Unternehmens zur Folge hat. Dieses Schema ist auf die Kommunikationsinterfaces von Geräten der Gebäudeautomation übertragbar. Ein Angreifer, der Netzzugriff hat, stellt eine Bedrohung für die einzelnen Geräte dar. In Tabelle 5.1 werden die einzelnen Risiken der OWASP Top 10 Liste erläutert und die analoge Gültigkeit für Geräte der Gebäudeautomation abgeleitet.

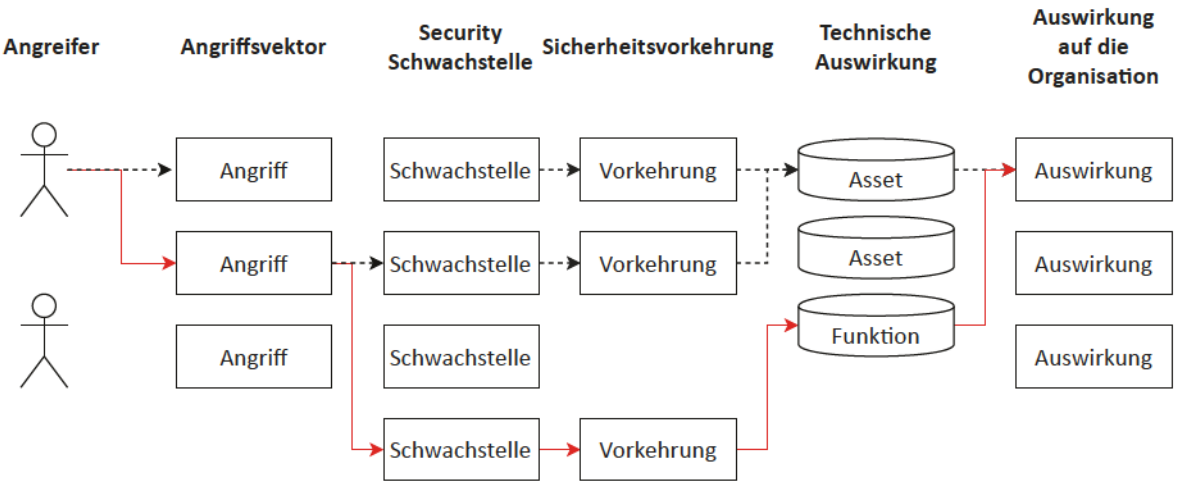

Abbildung 5.1: OWASP-Risiken [A 79]

5.2.1 Angreifer

Ein Angreifer ist eine einzelne Person oder eine Organisation, die versucht, technische Schutzmaßnahmen zu umgehen, um ein bestimmtes Ziel zu verfolgen.

5.2.2 Angriffsvektor

Als Angriffsvektor wird eine Angriffstechnik bezeichnet, um einen Computer, ein eingebettetes System oder ein technisches System für eigene Zwecke zu manipulieren oder unbefugt Daten abzugreifen. Je nach Definition wird die Kombination eines Angriffswegs unter Anwendung einer oder mehrerer Angriffstechniken bezeichnet. Ein Beispiel dafür ist das gezielte Kontaktieren eines einzelnen Mitarbeiters per Phishing E-Mail, um an gültige Zugangsdaten für den VPN-Server eines Unternehmens zu gelangen. Bei der Phishing Mail handelt es sich um eine gefälschte Mail von einer vertrauenswürdigen Person. Der Angreifer nutzt Schwachstellen der menschlichen Psyche, um die Abschottung des Unternehmensnetzes für Unbefugte durch ein VPN zu umgehen. Der Angriffsweg kann mit Zugriff auf das interne Netzwerk durch Anwendung weiterer Angriffstechniken fortgeführt werden.

5.2.3 Security-Schwachstelle

Eine Security-Schwachstelle ist im Allgemeinen ein Fehler in einem technischen System, der durch Ausnutzen zur Kompromittierung führen kann. Dies kann bei-

spielsweise eine fehlerhafte Programmverzweigung innerhalb einer Software sein. Nicht jede Schwachstelle ist jedoch für einen Angreifer erreichbar. Durch Softwareinterne Kontrollflussoperationen sind bestimmte Codeabschnitte nicht erreichbar. Ist eine Schwachstelle für einen Angreifer verfügbar, so führt dies zu einer Verwundbarkeit (engl.: Vulnerability). Ein Softwaretool, ein Skript, oder Ähnliches zum schematischen Ausnutzen dieser Verwundbarkeit wird ”Exploit” genannt.

5.2.4 Sicherheitsvorkehrung

Eine Sicherheitsvorkehrung ist eine technische Maßnahme, um Güter oder Funktionen vor dem Zugriff durch einen Angreifer zu unterbinden, zu erkennen oder zu stoppen. So kann beispielsweise der Zutritt zu einem lokalen Netz durch eine sogenannte “Network Access Control” (NAC) gewährleistet werden. Eine Maßnahme zur Erkennung eines Angriffs sind intelligente Filterregeln, die den Datenverkehr zwischen Geräten analysieren. Ein Portscan, bei dem TCP oder UDP Ports nach offenen Anwendungen gescannt werden, lässt sich durch bestimmte Traffic-Muster erkennen. Ist es einem Angreifer gelungen, bei einem Angriff erfolgreich Daten eines Geräts zu verschlüsseln, um für die Freigabe selbiger Daten Lösegeld zu fordern, sind Daten-Backups an sicheren Orten (air-gapped, Laufwerke ohne Netzwerkverbindung) als eine Sicherheitsvorkehrung zu nennen.

5.2.5 Technische Auswirkung

Jeder erfolgreiche Angriff hat technische Auswirkungen auf einzelne Geräte, Teilnehmer oder das Gesamtsystem. Im Falle der Gebäudeautomation versuchen Angreifer in der Regel sensible/personenbezogene Daten abzugreifen oder die Steuerung von Geräten und Anlagen zu übernehmen.

5.2.6 Auswirkung auf die Organisation

Der Angriff auf ein System hat zwangsläufig Auswirkungen auf die Organisation oder das Unternehmen. Angreifer verfolgen verschiedene Motive. Dazu zählen: Sabotage, um einen wirtschaftlichen Schaden hervorzurufen, die Monetarisierung des Angriffs durch den Verkauf von Datenbanken (z.B.: Kreditkartendaten) oder die Erpressung von Lösegeld für die Freigabe wichtiger Daten durch sogenannte ”Ransomware” oder die Demonstration der eigenen Fähigkeiten eines Angreifers.

Abbildung 5.1 beschreibt in allgemeiner Form Risiken, die es einem Angreifer ermöglichen, eine Auswirkung auf technische Assets (schützenswerte Güter) und Funktionen betreffender Organisation oder Unternehmen zu erzielen. Da es sich um Angriffe auf Webanwendungen handelt, besteht ein Angriff aus dem Ausnutzen einer Security-Schwachstelle, um Sicherheitsvorkehrungen zu umgehen und das Angriffsziel zu erreichen. Eine Sequenz aus verschiedenen ausgenutzten Security-Schwachstellen ist nicht vorgesehen.

Tabelle 5.1: OWASP - Security-Risiken für Webapplikationen und Geräte der GA [A 79]

Risiko	**Bedeutung**	**Bedrohung für Endgerät der GA**
Injection	Ein Interpreter empfängt nicht vertrauenswürdige Daten in Form einer Query (z.B. SQL Query).	Ein CoAP-Server kann Anfragen inklusive URL Query oder mit Query innerhalb der Payload entgegennehmen.
Broken Authentication and Session Management	Funktionen, die die Nutzer-Authentizität prüfen und das Sitzungsmanagement implementieren, können Softwarefehler enthalten.	Ein Angreifer kann sich als ein anderer Teilnehmer (menschlicher Nutzer oder Gerät) ausgeben und Sitzungen übernehmen. Im Fall von CoAP können Sitzungs-Cookies verwendet werden.
Cross Site Scripting (XSS)	Ein Server sendet nicht vertrauenswürdige Daten an einen Client, sodass sensible Daten seitens des Clients offenbart werden.	Nicht vertrauenswürdige Daten können auch über Protokolle wie CoAP vom Server versendet werden.
Insecure Direct Object References	Über die Anwendung ist ein ungewollter Zugriff auf interne Daten wie Dateien und kryptographische Schlüssel möglich.	Das Szenario ist protokollunabhängig. Durch Fehler in der Anwendungslogik können sensible interne Daten für einen Angreifer lesbar werden.
Security Misconfiguration	Fehlerhafte Sicherheitskonfiguration seitens des Servers	Die Nutzung einer Sicherheitsschicht, beispielsweise DTLS, kann ungewollter Weise deaktiviert sein. Außerdem könnte der Server durch fehlerhafte Konfiguration nicht vertrauenswürdige Public Keys von Clients akzeptieren.

Risiko	**Bedeutung**	**Bedrohung für Endgerät der GA**
Sensitive Data Exposure	Einem Angreifer ist es möglich, über die Webanwendung an sensible Daten wie Kreditkartennummern, Passwörter oder Passworthashes zu gelangen.	Die Daten der Endgeräte beinhalten sensible Informationen, die Rückschlüsse auf Gewohnheiten und die Anwesenheit von Bewohnern zulassen.
Missing Functional Level Access Control	Fehlende Prüfung von Nutzereingaben durch Webanwendung führt zu unerwünschtem Zugriff auf Funktionalitäten.	Jede CoAP-Ressource repräsentiert eine Funktionalität. Durch manipulierte Nutzereingaben können entsprechende Funktionen unerwünschter Weise ausgelöst werden.
Cross-Site Request Forgery (CSRF)	Während einer aktiven Sitzung kann ein Angreifer den Browser veranlassen unerwünschte Anfragen zu senden.	Anstelle eines Browsers können andere (CoAP-) Clients betroffen sein.
Using Components with Unknown Vulnerabilities	Einzelne Softwaremodule, die Fehler beinhalten können, können durch einen Angreifer ausgenutzt werden, um sensible Daten abzugreifen oder einen arbiträren Code einzuschleusen.	Endgeräte, die in der Gebäudeautomation Software ausführen, weisen dieselben Risiken auf. Insbesondere der Einsatz von Closed-Source Software macht das Aufspüren von Security-Bugs schwierig.
Unvalidated Redirects and Forwards	Browser wird durch Webanwendung an andere nicht-vertrauenswürdige Server umgeleitet.	Antworten eines CoAP-Servers können Links an andere Geräte-APIs enthalten. Der Client kontaktiert daraufhin nicht-vertrauenswürdige Teilnehmer oder Webanwendungen.

5.3 Anforderungen an die Security-Architektur einer Gebäudeautomation

Durch die zunehmende Vernetzung von Geräten mittels Webprotokollen und -architekturen gelten dieselben Security-Risiken für eingebettete Systeme. Angriffe auf Webanwendungen lassen sich direkt auf Dienste, die von Geräten angeboten werden, übertragen. Das Portfolio an Angriffen muss dazu nur leicht an die abgewandelten M2M-Standards angepasst werden. Dadurch, dass es sehr leicht möglich ist, physisch an Geräte eines Netzwerks zu gelangen, Schlüsselmaterial zu kompromittieren oder die Firmware zu manipulieren, kann sich ein Angreifer mit Leichtigkeit als ein legitimer/vertrauenswürdiger Teilnehmer ausgeben. Der Angriff auf ein GA-System lässt sich ausweiten, um verschiedene Absichten (Sabotage, Spionage, oder Ähnliches) zu verfolgen. Die Auswirkungen lassen sich durch dezentrale Anwendungsstrukturen verringern. Damit Angriffe nicht ohne Weiteres auf potentiell alle Geräte des Netzes ausgeweitet werden, muss die Angriffsfläche reduziert werden. Eine absolute Sicherheit ist nicht möglich. Jedoch lässt sich der Schwierigkeitsgrad durch dezentral organisierte und abgeschottete Anwendungen erhöhen. Außerdem muss ein Security-Konzept die Fluktuation von Geräten ins Netzwerk und aus dem Netzwerk berücksichtigen. Geeignete Mechanismen, die eine sichere Kommissionierung und Konfiguration zur Laufzeit ermöglichen, müssen anwenderfreundlich sein. Folgende Security-Andorderungen werden durch die in Kapitel 6 vorgestellte Security-Architektur adressiert:

- Ausweitung eines Angriffs auf weitere Geräte verhindern
- Sichere Generierung von Schlüsselmaterial für Geräte
- Berücksichtigung von neu kommissionierten Geräten
- Sicheres Entfernen von Geräten aus dem System
- Ende-zu-Ende Sicherheit zwischen Geräten
- Berücksichtigung von Ressourcen-Beschränkungen
- Nutzerfreundlichkeit während der Anwendungserstellung
- Dezentrale Ausführung der Gerätesteuerung

6 BIM-basierte Planung von Trust Zones

In diesem Kapitel wird ein neuartiger Algorithmus präsentiert, der jedes einzelne Gerät der GA mit individuellen Anwendungs- und Security-Konfigurationsdaten versorgt. Der besondere Fokus liegt dabei auf Gebäudeinformationsmodellen, die eine wertvolle Datenquelle darstellen um Geräte in abgeschottete Trust Zones zu gruppieren. Dieses Verfahren wurde in [B 4], [B 5] und [B 6] publiziert. In den vorgestellten Abläufen von der Kommissionierung bis hin zur Konfiguration und Rekonfiguration zur Laufzeit werden ausschließlich anerkannte Security-Protokolle angewendet. Abschließend wurde der Algorithmus, der Geräte auf Basis ihrer Attribute in Trust Zones einteilt, prototypisch implementiert und experimentell evaluiert.

6.1 Verwandte Arbeiten

In der Literatur existieren Arbeiten, die Security auf Systemebene betrachten. Es genügt nicht, die einzelnen Teilnehmer verschlüsselt und signiert kommunizieren zu lassen. Einzelne Angreifer, die sich im Netzwerk befinden, können Schaden anrichten. Die GA muss als ein System bestehend aus Teilnehmern, Anwendungen und Netzwerkinfrastruktur begriffen werden. Das Feld ”Software-defined Security” umfasst Konzepte, die eine logische Steuerebene beschreiben. Diese Steuerebene konfiguriert das Netzwerk und damit die Kommunikationsflüsse zwischen den einzelnen Teilnehmern. ”Software-defined Security” kann daher als eine Untergruppe der ”Software-defined Networking”-Klasse (SDN) angesehen werden. Beim ”Software-defined Networking” steuert eine übergeordnete Softwareinstanz die Netzwerkinfrastruktur, um bestimmte Anforderungen (wie Latenz und Datendurchsatzrate) zu erfüllen.

In [A 80] stellen die Autoren eine Software-defined Security-Architektur vor, die basierend auf einer künstlichen Intelligenz Anomalien im Netzwerk erkennt. Es werden keine Gegenmaßnahmen vorgeschlagen, die im Falle einer Anomalität das betreffende Gerät netzwerktechnisch isolieren. In der Publikation [A 81] wird eine Software-defined Security-Architektur präsentiert, die einen Security Controller beinhaltet, der das Netzwerk und die Endgeräte steuert und Logdaten auswertet. Der Security Controller kann optional eine externe Wissensdatenbank verwenden, um Sicherheitsrichtlinien abzuleiten. Jedoch wird in der Publikation SDN-basierte

Netzwerkinfrastruktur verwendet. Eine WLAN-basierte Kommunikation (ohne spezielle SDN-konforme Netzwerkgeräte) ist nicht vorgesehen. Das Konzept ist allgemeiner Natur und geht nicht auf die speziellen Gebäudeinformationsmodelle ein. Eine ähnliche Architektur wird in [A 82] beschrieben. Das System basiert ebenso auf speziellen SDN-Switches. Die Arbeitsweise der Steuereinheit wird lediglich in einer simulativen Umgebung gezeigt.

Die Autoren von [A 83] stellen die Vorteile eines Digital Twin heraus. Dieser repräsentiert zum einen ein reales System und zum anderen kann er genutzt werden, um steuernd auf das System zu wirken.

In [A 84] stellen die Autoren ein Konzept vor, das wissensbasierte Systeme für das Gebäudemanagement vorsieht. Durch die eingeführte Abstraktionsschicht wird eine vereinfachte, effizientere Kommunikation mit den einzelnen Endgeräten erzielt. Der vorgestellte Ansatz hilft bei der Umsetzung von Anwendungen zwischen Sensoren und Aktoren. Die Wissensbasis wird nicht zur Verbesserung der Security verwendet.

Security-Aufgaben an eine übergeordnete Instanz zu delegieren, findet sich in verschiedenen Ansätzen. Diese werden in [A 85] aufgelistet und verglichen. Bei den vorgestellten Konzepten [A 86], [A 87], [A 88], [A 89], [A 90], [A 91], [A 92] und [A 93] liegt der Fokus auf Schlüsselverteilungsverfahren, damit Endgeräte Anfragen autorisieren und die Authentizität von Nachrichten validieren können. Die Absicherung der Kommunikation erfolgt auf Transportebene. Keiner der Ansätze stellt die Vorteile einer Verschlüsselung auf MAC-Ebene heraus. Gebäudeinformationsmodelle werden in keinem anderen Konzept zur Berechnung von Securitykonfigurationen (inkl. Schlüsselmaterial) verwendet.

Der nachfolgende Abschnitt beschreibt, wie Gebäude modelliert werden können. Das Gebäudemodell dient als zentraler Baustein des Digital Twin. Damit ist das Gebäudemodell eine wichtige Datenquelle, um das Gesamtsystem zu partitionieren und Security-Konfigurationen für die einzelnen Endgeräte abzuleiten.

Bislang existiert keine Security Architektur, die speziell auf GA-Systeme zugeschnitten ist und entsprechende Gebäudemodelle verwendet. Es finden sich keine Konzepte, deren Eignung auf realer eingebetteter Hardware evaluiert wurde. Die Vorteile der Absicherung der GA-Kommunikation auf MAC-Ebene wird in keiner Veröffentlichung durch einen Security Controller unterstützt.

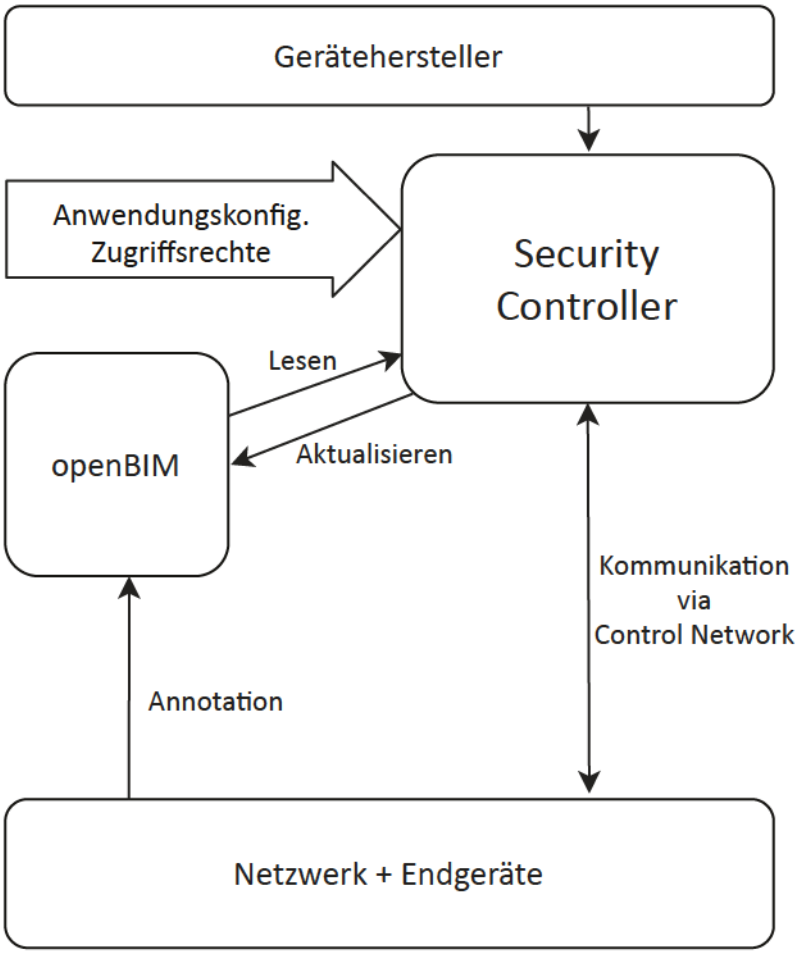

Abbildung 6.1: Architektur des Gesamtsystems [B 4]

6.2 Architektur

In Abbildung 6.1 ist die vorgeschlagenene Sicherheitsarchitektur dargestellt. Das BAS, bestehend aus der Gesamtheit an Endgeräten und Netzwerkinfrastruktur wird durch einen Security Controller (SC) konfiguriert. Der SC stellt einen Vertrauensanker dar, der basierend auf einer Ground-Truth-Datenbank (openBIM mit zusätzlichen Gerätedaten) die Netzwerkinfrastruktur und die Anwendungen der einzelnen Endgeräte (End Device - ED) konfiguriert. Zusätzlich zur Ground-Truth-Datenbank ist es möglich, durch legitimierte Nutzer (Administrator) gebäudespezifische Anwendungskonfigurationen zwischen Sensoren und Aktoren zu definieren und Security Policies festzulegen. In den nachfolgenden Abschnitten werden die einzelnen Phasen des Sicherheitskonzeptes näher erläutert. Zunächst wird der Kommissionierungsprozess ausgeführt. Dabei wird jedes ED mit dem SC verknüpft und mit

initialen Einstellungen versorgt, um einem Konfigurationsnetz beizutreten. Anschließend sammelt der SC Informationen über das Netzwerk. Die Verknüpfungen zwischen den einzelnen EDs auf Anwendungsebene werden formal verifiziert. Nach erfolgreicher Verifikation werden die abstrakten Anwendungsgraphen auf reale Geräte abgebildet. Anschließend werden die Geräte in separate Domänen, die Trust Zones, eingeteilt und mit entsprechenden Konfigurationsdaten versorgt. Nachdem alle Schritte ausgeführt worden sind, können die EDs in abgeschotteten Trust Zones miteinander kommunizieren.

6.3 Phasen

In Tabelle 6.1 werden die einzelnen Phasen von der Kommissionierung bis zur Planung und Anwendung von Trust Zones erläutert. Im ersten Schritt werden die Geräte kommissioniert. Dabei erhalten sie Schlüsselmaterial, um einem Konfigurationsnetzwerk beizutreten. Nachdem alle Geräte kommissioniert worden sind, sammelt der Security Controller Informationen der Netzwerkebene. Anschließend werden die Verknüpfungen zwischen den Geräten virtuell erstellt und formal auf Korrektheit verifiziert. Die entstandenen Anwendungsgraphen werden auf reale Geräte abgebildet. Dabei werden allgemeingültige Sicherheits-Policies eingehalten. Sämtliche Endgeräte werden in abgeschottete Netzwerkbereiche eingeteilt, um unerwünschte Kommunikationswege zu unterbinden.

Tabelle 6.1: Phasen der Netzwerk- und Anwendungsplanung

Phase	**Bedeutung**
1) Kommissionierung von Endgeräten	Während der Kommissionierungsphase wird jedes Endgerät durch einen Techniker mit dem Konfigurationsnetz verbunden (Abbildung 6.2). Zusätzlich zu den out-of-band übertragenen Konfigurationsnetz-Zugangsdaten erhält jedes Gerät vom SC ein Schlüsselpaar (öffentlicher und privater Schlüssel). Durch die entstandene Public-Key-Infrastruktur ist es während der Konfiguration und des Betriebs des GA-Systems möglich, Nachrichtenauthentizität und -integrität zu gewährleisten. Der SC speichert jedes Gerät in einer lokalen, geschützten Datenbank ab. Neben dem öffentlichen Schlüssel werden der Installationsort und der Gerätetyp durch einen Techniker eingetragen bzw. verifiziert. Diese im SC gespeicherten Daten werden als Ground Truth bezeichnet. Auf Basis dieser gesicherten Informationen ist es möglich, automatisiert Anwendungen und das Netzwerk zu planen.
2) Sammlung von Informationen der Netzwerkebene	Nachdem alle Geräte installiert und zur Ground-Truth-Datenbank hinzugefügt wurden, sammelt der SC globales Wissen zur Netzwerktopologie. Dazu gehören Informationen zur Konnektivität via Ethernet und/oder WLAN. Der SC erhält Ortsinformationen zu Ethernet-Leitungen und angeschlossenen Ethernet-Switches. Weiterhin ruft der SC die Peer-Link-Liste jedes WLAN-Mesh-Geräts ab, um einen Netzwerkgraphen zu erstellen.
3) Erstellung und Verifikation der Anwendungsebene	Um die Sensoren und Aktoren einer GA zu verknüpfen, werden ITTT-Regeln erstellt. Jede Regel wird als gerichteter, azyklischer Anwendungsgraph modelliert. Logische Fehler innerhalb einzelner Regeln und zwischen verschiedenen Regeln werden durch formale Verifikation ausgeschlossen.
4) Abbildung der Anwendungen auf Geräte	Nachdem die logische Korrektheit der Regeln gewährleistet ist, werden die Knoten der korrespondierenden Anwendungsgraphen auf reale Geräte der Ground-Truth-Datenbank abgebildet. Alle Geräte eines Anwendungsgraphen werden zu einer Trust Zone hinzugefügt.
5) Policy-basierte Bildung von Trust Zones	Sollten die Geräte eines Anwendungsgraphen keine netzwerktechnische Konnektivität aufweisen, müssen die Kanten des Graphen durch weitere Netzwerkinfrastruktur realisiert werden. Ein Algorithmus fügt WLAN-Mesh-Geräte als Infrastruktur-Peers einer Trust Zone hinzu. Dabei werden der Installationsort (Verknüpfung mit BIM) und eine allgemeingültige Policy berücksichtigt (Abbildung 6.9).

6.3.1 Kommissionierung von Endgeräten

Während der ersten Phase werden die Endgeräte mit dem Konfigurationsnetzwerk verbunden. Dabei vermittelt ein legitimierter Nutzer per Smartphone zwischen dem Endgerät und dem SC, um eine Public-Key-Infrastruktur zu errichten, Geräte-Metadaten auszutauschen und den Gerätestandort festzulegen. Der Ablauf zwischen Endgerät, Nutzer und SC ist in Abbildung 6.2 dargestellt. Folgende Annahmen werden getroffen:

1. Der SC verfügt über eine Liste mit Nutzern samt ihrer Zugangsdaten. Diese Nutzer sind autorisiert, Endgeräte zu kommissionieren.
2. Das Smartphone und der SC verfügen über Schlüsselmaterial, um eine geschützte Verbindung miteinander aufzubauen.
3. Es wird ein sicheres Authentifizierungsverfahren verwendet, um den Nutzer zu identifizieren und den Kommissionierungsprozess zu starten.
4. Das Smartphone zur Mensch-Maschine-Interaktion wird als vertrauenswürdig und sicher angenommen. Die auf dem Smartphone ausgeführte Software implementiert die in [A 94] vorgestellte Schlüsselaushandlung und die in dieser Arbeit vorgeschlagenen Mechanismen korrekt.
5. Die Erzeugung von Schlüsselpaaren durch den SC kann nicht von Angreifern manipuliert werden.
6. Private Schlüssel werden sicher auf Endgeräten gespeichert.
7. Es werden sichere asymmetrische Verschlüsselungsverfahren eingesetzt.
8. Das eingesetzte Verfahren zur Aushandlung von Shared Keys zwischen zwei Geräten auf Basis der PKI ist sicher und korrekt implementiert.

In Abbildung 6.4 findet sich der Programmablaufplan des SC um die einzelnen EDs zu kommissionieren. Der SC verfügt über eine Liste mit Nutzern, die autorisiert sind, neue Geräte hinzuzufügen. Um Geräte hinzufügen zu können, meldet sich der Nutzer per Smartphone am SC an (Abbildung 6.2). Nach erfolgreichem Login-Vorgang wird Schlüsselmaterial zwischen Endgerät und SC ausgehandelt. Das technische Problem besteht darin, dass ein Gerät dieses Schlüsselmaterial bzw. Daten für eine Schlüsselaushandlung über einen gesicherten Übertragungsweg erhalten muss.

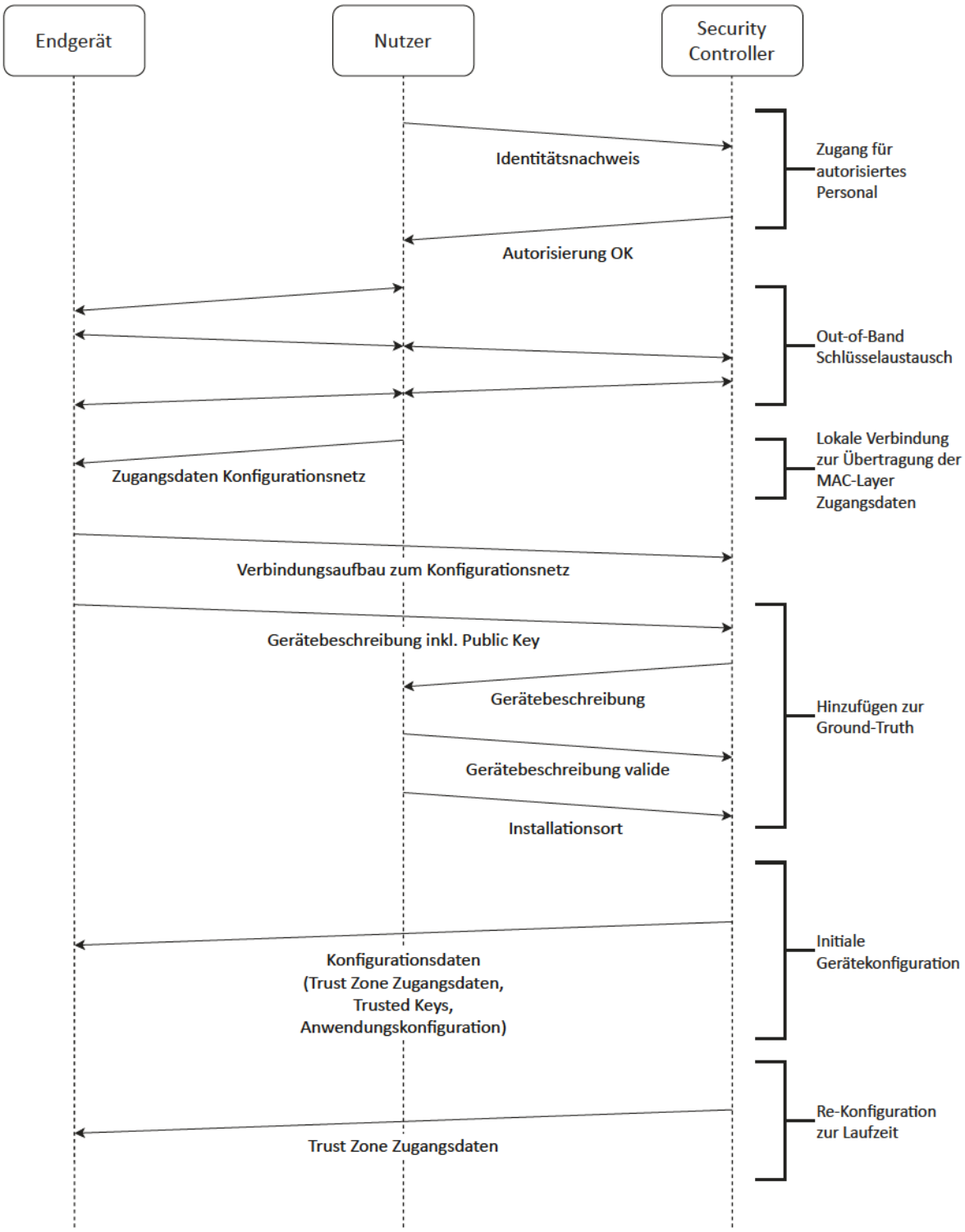

Abbildung 6.2: Kommissionierung eines Endgeräts durch einen Nutzer [B 5]

Dieser ist jedoch zu Beginn der Kommissionierung noch nicht vorhanden. Da es sich bei den Endgeräten der GA um eingebettete Systeme mit limitierten Ein- und Ausgabemöglichkeiten handelt, können Schlüssel oder Shared Secrets nicht oder nur umständlich vom Nutzer gesetzt werden. Im Falle eines Gerätes mit nur einem Taster (Lichtschalter oder Sensor) müsste ein menschlicher Bediener einen 256-bit AES-Schlüssel oder ein 256-bit Pre-shared Secret out-of-band per Drücken des Tasters eingeben. Eine benutzerfreundliche Codierung ist nicht möglich. Als Verfahren zur sicheren Schlüsselaushandlung (symmetrischer Schlüssel) wird auf das Verfahren [A 94], das auf einem authentifizierten Diffie-Hellman-Schlüsselaustausch nach [A 95] beruht, zurückgegriffen.

Damit nach erfolgreichem Schlüsselaustausch Endgerät und SC sicher (verschlüsselt und signiert) miteinander kommunizieren können, muss das Endgerät dem Konfigurationsnetz beitreten. Das Konfigurationsnetz basiert auf Ethernet und IEEE 802.11s WLAN-Mesh, um verschiedene Geräte zu unterstützen. Die Sicherheit des Konfigurationsnetzes basiert im Falle von Ethernet auf IEEE 802.1AE [A 96]. Dieser Standard, welcher auch unter dem Namen ”MACsec” bekannt ist, dient der Network Access Control (NAC). Die benötigten Zugangsdaten werden vom Nutzer per Smartphone an das Endgerät übertragen. Das WLAN-Mesh-basierte Konfigurationsnetz verwendet das SAE-Verfahren [A 97] zur Absicherung der MAC-Schicht. Die Zugangsdaten für SAE bestehen aus der SSID und dem Pre-shared Secret. Nach Erhalt dieser Daten führt das Endgerät mit einem beliebigen anderen Endgerät, das bereits Teil des Konfigurationsnetzes ist, per SAE einen Schlüsselaustausch durch, um dem gesicherten WLAN-Mesh-Netzwerk beizutreten. Der vom ED implementierte Zustandsautomat ist in Abbildung 6.3 dargestellt.

Im nachfolgenden Schritt wird die Gerätebeschreibung des Endgerätes vom SC abgerufen. Der Nutzer verifiziert den Gerätetyp, sodass dieser Teil der Ground Truth wird. Weiterhin ist der Installationsort zum Zeitpunkt der Kommissionierung bekannt. Dieser wird durch den Nutzer via Smartphone GUI der SC Ground Truth Datenbank hinzugefügt. Das BIM enthält Daten, um das Gebäude als 3-dimensionale Struktur zu visualisieren. Somit ist es möglich, eine entsprechende graphische Nutzeroberfläche zu designen, um den Standort zu annotieren. Lokalisierungsverfahren eignen sich weniger, da die Lokalisierungsgenauigkeit fehlerhafte Standortdaten liefert.

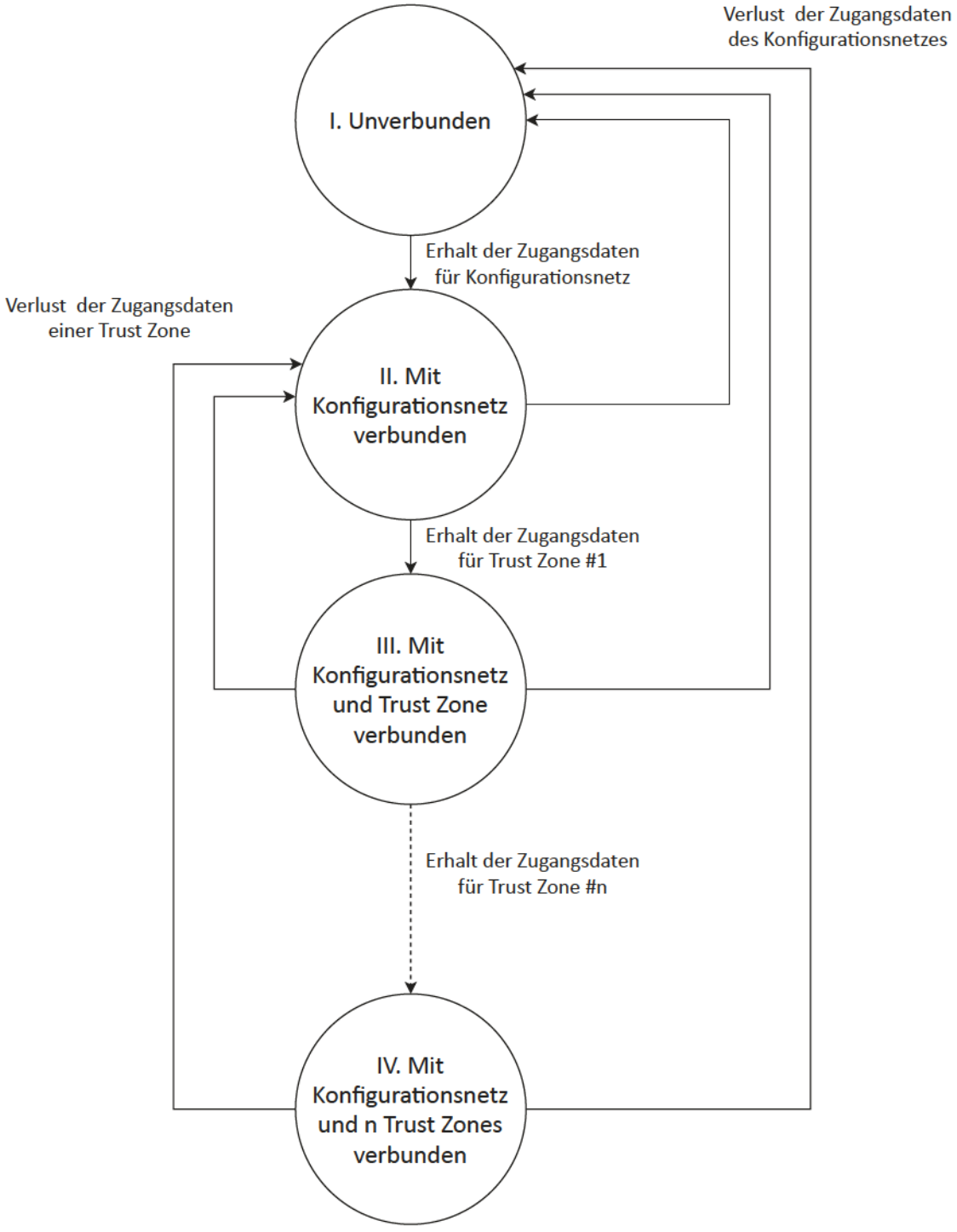

Abbildung 6.3: Zustandsautomat eines Endgeräts [B 5]

Nachdem alle Geräte in die PKI eingebunden wurden und eine Konnektivität zum Konfigurationsnetzwerk erhalten haben, folgt das Einholen von Informationen der Netzwerkebene.

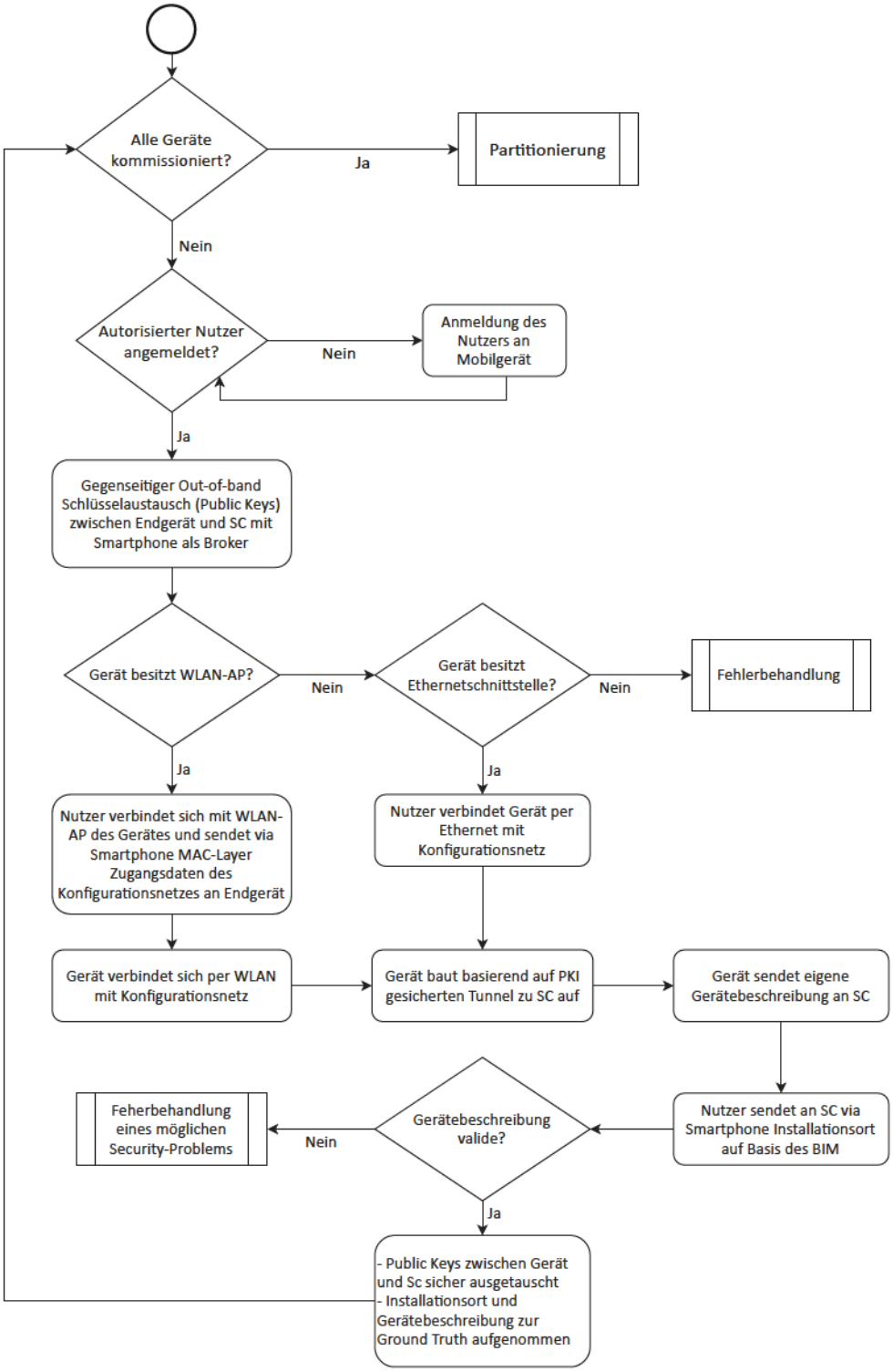

Abbildung 6.4: Kommisionierung eines Endgeräts [B 5]

6.3.2 Sammlung von Informationen der Netzwerkebene

Um eine Netzwerkpartitionierung unter Berücksichtigung der netzwerktechnischen Distanz durchführen zu können, benötigt der SC Informationen über die einzelnen Verbindungen jedes Endgerätes. Dabei wird zwischen Ethernet- und WLAN-Mesh-Verbindungen unterschieden. Ethernet-Kabelverbindungen von jedem Endgerät werden vom Nutzer manuell via Smartphone GUI an das BIM annotiert. WLAN-basierte Links zwischen Endgeräten des Konfigurationsnetzes werden mit einem automatisierten Verfahren gesammelt. Das Konzept, WLAN-Mesh-Peer Statusinformationen per SNMP [A 98] von einem zentralen Knoten, dem WLAN-Mesh-Manager, abzurufen, wurde in [B 7] vorgestellt. Der als “Mesh-Manager” bezeichnete Knoten steht analog zum SC, der Statusinformationen der einzelnen Endgeräte per SNMP abruft. In Abbildung 6.5 sind SC und Endgerät schematisch dargestellt. Beide Geräte kommunizieren über das IEEE 802.11s Mesh-basierte Konfigurationsnetz. Das Kernelmodul *mac80211* dient der Integration der 802.11s Implementierung *open80211s*. In der aktuellen Version des Linux-Kernels ist die 802.11s Implementierung Bestandteil des Software-WLAN-MAC-Layers. Das Endgerät führt einen SNMP Server aus, der in Java implementiert ist. Durch Aufrufen des Kommandozeilentools *iw* aus der Java Virtual Machine heraus, wird die MAC-Schicht Routingtabelle samt Metrik ausgelesen. SNMP unterstützt das Auslesen von Statusinformationen, die in Form von Statusobjekten von einem Server angeboten werden. Die Routingtabelleneinträge werden ebenfalls durch Statusobjekte repräsentiert. Jedes Statusobjekt wird per GET-Operation vom Server abgerufen. Der Server wird von jedem Endgerät ausgeführt. Es werden lediglich eingehende Anfragen vom SC aus dem Konfigurationsnetz-Interface angenommen.

Das HWMP (Hybrid Wireless Mesh Protocol) wird von jedem Mesh Point standardmäßig unterstützt, um als einheitliches Profil das Routing im Mesh-Netzwerk zu definieren [A 99]. Die ALM dient als Weg-Metrik zwischen zwei MPs [A 48]. Das reaktive Ad-Hoc On-Demand Distance Vector (AODV) Routingprotokoll [A 100] basiert auf der ALM, um die netzwerktechnische Distanz zwischen Quelle und Ziel eines Frames zu quantifizieren. Bei der ALM handelt es sich um die akkumulierten 1-Hop Kosten, die während einer Mulit-Hop Übertragungsstrecke entstehen. Die *Airtime Cost* c_a eines Links zwischen zwei Mesh Peers beschreibt die geschätzte Gesamtzeit für die Übertragung eines Frames zwischen zwei Endpunkten.

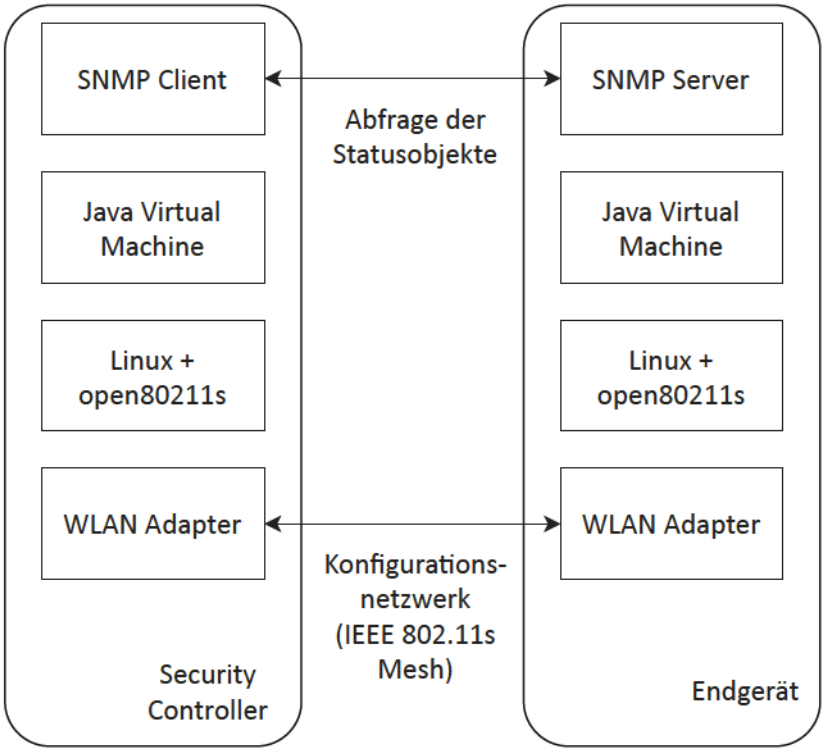

Abbildung 6.5: Abfrage der Routingtabelle eines Endgerätes durch den Security Controller via SNMP

Sie berechnet sich gemäß dem WLAN-Standard nach folgender Formel [A 48]:

$$c_a = \left[O + \frac{B_t}{r}\right] \cdot \frac{1}{1 - e_{fr}} \tag{1}$$

Die Variable O beschreibt die Kanalzugriffszeit. Sie wird als konstant angenommen. B_t beschreibt die Größe eines Test-Frames. Der Default-Wert ist mit 8192 bit beziffert. r steht für die Datenrate, mit der der Test-Frame übertragen wird. Als Einheit der Datenrate wird Mbit/s verwendet. Da durch fehlerhafte Frame Übertragungen erneute Sendeversuche nötig sind, erhöht sich die mittlere Kanalzugriffszeit. In der Praxis unter Linux geschieht dies nachdem das Retry-Limit (die maximale Anzahl an Neuübertragungsversuchen) erreicht wurde. Daher wird in der Airtime Cost Formel die Fehlerrate e_{fr} berücksichtigt. Nach der IEEE 802.11s-Erweiterung werden die Schätzung der Fehlerrate e_{fr} sowie die Konstante O nicht beziffert. Dies ist dem Hersteller überlassen [A 101].

6.3.3 Erstellung und Verifikation der Anwendungsebene

Die Geräteverknüpfungen auf Anwendungsebene basieren auf dem Konzept, das in [B 3] [6.1] publiziert wurde. Traditionell werden Geräte der Feldebene auf Steuerungsebene durch Speicherprogrammierbare Steuerungen (SPS) verknüpft. SPS-Anlagen werden fest mit den einzelnen Geräten verbunden, um eine Kommunikation zum Lesen von Sensorwerten und zum Steuern von Aktorik zu ermöglichen. Die logischen Funktionen können von der Administration der Gebäudeleittechnik frei definiert werden. Dazu wird die von der SPS-Anlage ausgeführte Software modifiziert. Da dieser zentralisierte Ansatz anfällig für Ausfälle der Steuerungsanlage ist, da es zu einem Ausfall aller Funktionen innerhalb der GA kommen kann, eignen sich dezentrale Ansätze zur Steuerung besser. Das in [B 3] publizierte Konzept führt einen Konfigurationsservice ein, der generisch für eingebettete Systeme implementiert werden kann. Das Interface des Konfigurationsservices bietet die in Tabelle 6.2 aufgeführten Funktionen an. Im Gegensatz zu [B 3] werden die einzelnen Funktionalitäten nicht in Form eines DPWS-basierten Web Services sondern durch eine RESTful API im Netzwerk angeboten.

Tabelle 6.2: RESTful API des Konfigurationsservices

Ressource	**Beispieldaten**
/config/getSupportedInterfaces	Motion Detector
/config/ConnectDevice	null
/config/getConnectedDevices	Motion Detector [ID]
/config/SetRule	null
/config/getRules	null

Die von einem Gerät unterstützten Interfaces können somit abgefragt werden. Bezeichnungen für entsprechende Geräte-Interfaces (z.B.: Sensor::Bewegungsmelder, Sensor::Temperaturfühler, Sensor::Helligkeitssensor,

[6.1] Das Grundkonzept, mittels Smartphone DPWS-basierte Geräte per WS-Discovery zu finden, um sie durch einen selbst-definierten Konfigurationsservice miteinander zu verbinden, stammt von Herrn Dr.-Ing. Vlado Altmann (1. Co-Autor). Die Eigenleistung besteht in der formalen Beschreibung der Konfigurationsregeln, der Ausarbeitung von logischen Regelkonflikten und der Überführung in Binary-Decision-Diagramme zur formalen Verifikation der Abwesenheit von Konfigurationsfehlern. Der von Herrn Dr.-Ing. Vlado Altmann vorgeschlagene Konfigurationsservice für DPWS-basierte Geräte wurde auf CoAP-basierte Geräte übertragen und maßgeblich durch zusätzliche Funktionen zur Integration in eine abgesicherte Gebäudeautomation erweitert.

Aktor::Fenstersteuerung und Aktor::Beleuchtung) müssen samt Datenmodell herstellerübergreifend standardisiert sein. Die Open Mobile Alliance [A 46] löst das Problem durch öffentlich einsehbare Standards. Der Konfigurationsservice nimmt unterstützte Geräte entgegen. Dabei werden folgende Daten vom SC übermittelt:

- IP-Adresse
- Protokoll-Stack
- Pre-Shared Key (PSK)

Damit ein Gerät, das den Konfigurationsservice ausführt, Sensoren und Aktoren abfragen bzw. steuern kann, müssen neben der IP-Adresse der zu verwendende Protokoll-Stack samt Schlüsselmaterial (PSK) vom SC übermittelt werden. Dieses Konzept ist somit auf alle Protokolle übertragbar, die einen PSK zur kryptographischen Absicherung der Kommunikation unterstützen. Dazu zählen DTLS, TLS und OSCORE. Je nach Protokoll werden Pre-shared Secrets unterstützt, um einen Sitzungsschlüssel zwischen beiden Kommunikationspartnern auszuhandeln. Die unterstützten Protokoll-Stacks müssen durch eine Selbstbeschreibung jedes Endgerätes abrufbar sein, damit der SC entsprechende Pre-shared Secrets bzw. Pre-shared Keys individuell zuordnen kann.

Es ist möglich, dass ein Aktor den Konfigurationsservice ausführt und Sensorwerte von anderen Geräten empfängt (2-stufiger Anwendungsgraph in Abbildung 6.6). Außerdem besteht die Möglichkeit, dass der Konfigurationsservice eines Gerätes Sensordaten auswertet und gemäß der Konfigurationsregel ein anderes Gerät steuert. Der resultierende 3-stufige Anwendungsgraph ist in Abbildung 6.7 dargestellt.

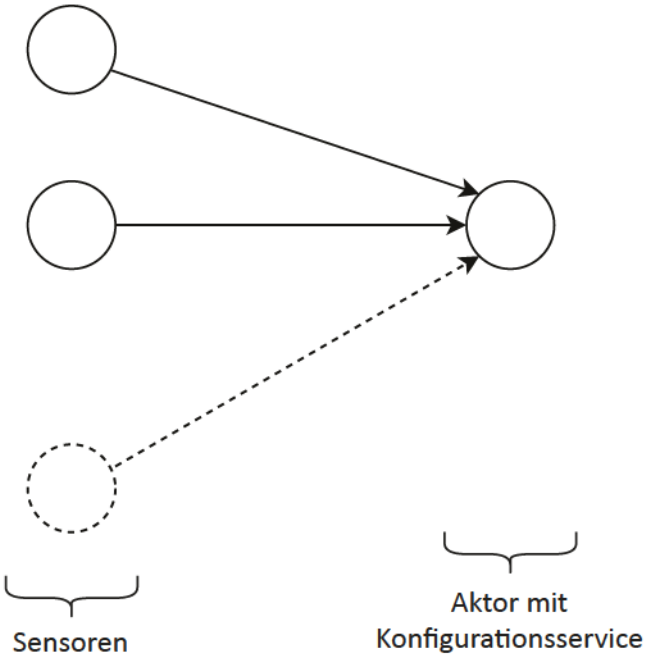

Abbildung 6.6: Anwendungsgraph bestehend aus Sensoren, die mit einem Gerät verbunden sind, das den Konfigurationsservice anbietet

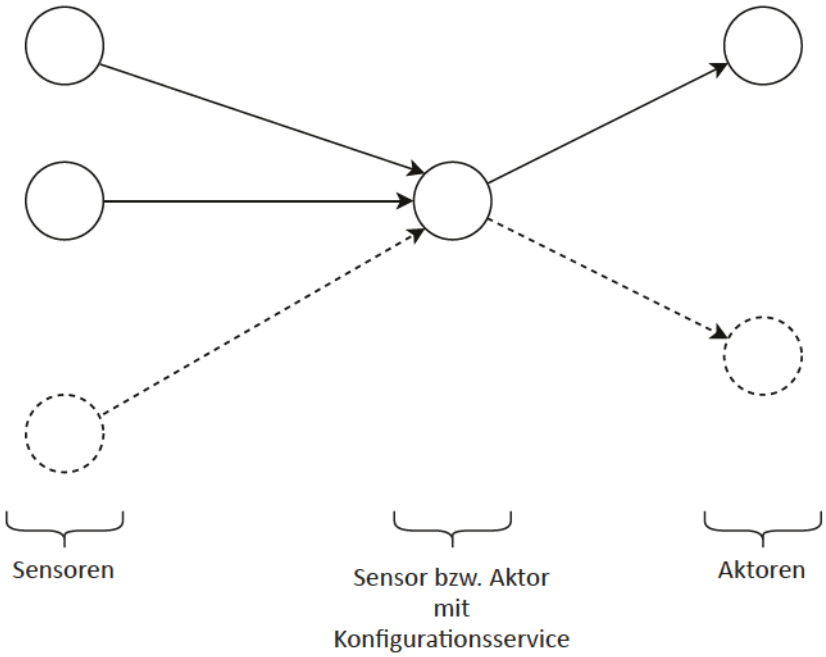

Abbildung 6.7: Anwendungsgraph bestehend aus einem Gerät, das den Konfigurationsservice anbietet und mit Sensoren und Aktoren verbunden ist

6.3.3.1 Formale Beschreibung von Konfigurationsregeln

Jede Regel (6) [B 3] besteht aus einzelnen Bedingungen (4), die durch boolesche Operatoren (5) miteinander verknüpft sind. Jede Bedingung nach (4) (mathematischer Term) beschreibt eine physikalische Größe (Parameter), die durch einen Operator (2) mit einem Schwellenwert (Element der ganzen Zahlen) verglichen wird. Eine sogenannte Messfunktion (3) bildet jeden Parameter auf eine ganze Zahl ab, sodass ein Vergleich mit einem Schwellenwert möglich ist. (7) beschreibt ein Beispiel für eine Regel, die ein Fenster öffnet wenn die Außentemperatur kleiner ist als der Sollwert und die Innentemperatur größer ist als der Sollwert.

$$OPERATION := \{=, \neq, >, \geq, <, \leq\} \tag{2}$$

$$f : PARAMETER \rightarrow \mathbf{Z} \tag{3}$$

$$\begin{aligned} CONDITION := \{&PARAMETER \times \\ &OPERATION \times \mathbf{Z}\} \end{aligned} \tag{4}$$

$$\begin{aligned} BOOLOPERATOR := \{&AND, OR, NAND, \\ &NOR, XOR, XNOR\} \end{aligned} \tag{5}$$

$$RULES := \mathcal{P}(CONDITION) \times BOOLOPERATOR \tag{6}$$

$$Fenster\ auf = (Aussentemperatur < Sollwert) AND (Innentemperatur > Sollwert) \tag{7}$$

6.3.3.2 Konfigurationsfehler

Es ist möglich, dass Konfigurationsregeln logische Fehler beinhalten [B 6]. Außerdem ist es möglich, dass zwei oder mehrere Konfigurationsregeln dieselbe Aktorik steuern. Dabei kann es zu Kollisionen kommen. In Tabelle 6.3 werden alle Möglichkeiten von Konfigurationsfehlern samt Gegenmaßnahmen erläutert.

Tabelle 6.3: Konfigurationsregel-Fehlertypen

Fehlertyp	**Erläuterung**	**Gegenmaßnahme**
1) Inhärenter Fehler	Logische Fehler können innerhalb einer Regel auftreten, sodass die Funktion unabhängig von den Argumenten immer wahr (Tautologie) oder falsch (Kontradiktion) ist.	Konfigurationsregeln nach Formel (6) lassen sich durch Substitution der Bedingungen (4) in boolesche Funktionen umwandeln. Durch eine Beschreibung der jeweiligen Funktion als Binary Decision Diagram (deutsch: Binäres Entscheidungsdiagramm) lässt sich eine Tautologie bzw. Kontradiktion erkennen (siehe Abschnitt 6.3.3.3).
2) Kollision zweier Regeln des selben Steuergerätes	Zwei Regeln, die dieselbe Aktorik steuern, können widersprüchliche Steuerbefehle senden. Die Regeln werden von einem einzelnen Steuergerät ausgeführt.	Der SC führt eine formale Verifikation der Regeln (nach Abschnitt 6.3.3.3) aus, um Kollisionen auszuschließen.
3) Kollision zweier Regeln verschiedener Steuergeräte	Analog zu Punkt 2 kollidieren zwei oder mehr eigenständige Regeln.	Der Algorithmus des SC verifiziert die einzelnen Regeln, die potentiell kollidieren können. Werden die Parameter einer Regel zur Laufzeit durch einen Nutzer verändert, muss das entsprechende Steuergerät die Logik auf Konfigurationsfehler nach Punkt 1 und 2 prüfen.

6.3.3.3 Formale Verifikation der Anwendungslogik

In [B 6] wurde ein formales Verfahren zur Verifikation der Anwendungslogik vorgestellt. Dabei kommen binäre Entscheidungsdiagramme (BDDs, Binary Decision Diagram) zum Einsatz. Sie repräsentieren boolesche Funktionen. Jeder Knoten stellt eine Variable dar. Da jede Variable zwei Zustände (wahr - durchgezogene Linie und falsch - gestrichelte Linie) annehmen kann, gehen von jedem Knoten zwei Kanten ab. Die Blätter des Graphen repräsentieren die Funktionswerte der booleschen Funktion. Wird die Reihenfolge der Variablen berücksichtigt, so spricht man von einem geordneten binären Entscheidungsbaum (OBDD, Ordered Binary Decision Diagram). OBDDs lassen sich miteinander verknüpfen und vereinfachen.

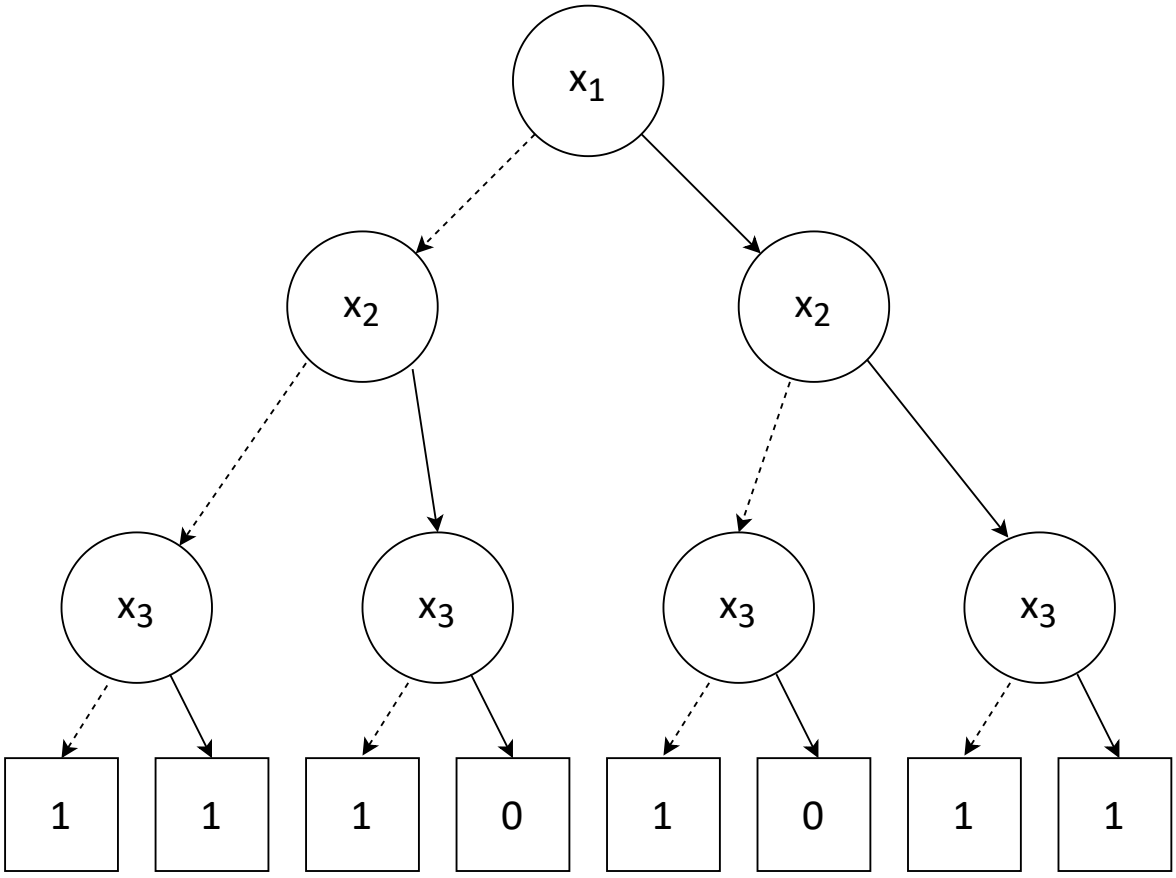

Abbildung 6.8: Binärer Entscheidungsbaum zur Repräsentation einer booleschen Funktion mit drei Variablen

Die Konfigurationsregeln nach Abschnitt 6.3.3.1 lassen sich durch Substitution der Terme mit Vergleichsoperator in boolesche Funktionen überführen. Damit lässt sich

jede Regel mit einem BDD modellieren. Wenn zwei oder mehr Regeln dieselbe Aktorik steuern, treten potentiell Kollisionen auf. Die einzelnen booleschen Funktionen liefern bei einer bestimmten Eingangsvariablenbelegung verschiedene Werte, sodass ein Widerspruch entsteht. Wenn zwei BDDs dieselbe Variablenreihenfolge besitzen, lassen sie sich mit zwei OBDDs beschreiben, sodass eine Verknüpfung beider Graphen miteinander möglich ist. Wenn beide Funktionen XOR (exclusiv oder) verknüpft werden und mindestens ein Blattknoten wahr ist, so existiert mindestens eine Eingangsvariablenbelegung, die zu einem Konflikt führt. Sogenannte SAT-Solver können angewendet werden, um entsprechende Variablenbelegungen zu identifizieren. Im Gegensatz dazu kann das Fehlen von Fehlerzuständen bewiesen werden, wenn alle Blätter falsch sind. Durch Substitution der Variablen mit den Vergleichstermen kann der Fehlerfall eingegrenzt werden. Logische Konfigurationsfehler lassen sich während der Inbetriebnahme der GA identifizieren. Weiterhin können angepasste Regeln zur Laufzeit mit dem vorgestellten Verfahren formal verifiziert werden. Da die Konfigurationsregeln in einer GA üblicherweise keine große Tiefe (wenige Variablen) aufweisen, sind keine besonderen Komplexitätsbetrachtungen nötig.

6.3.4 Abbildung der Anwendungen auf Geräte

Da es sich bei der Erstellung der Anwendungsgraphen um eine abstrakte Darstellungsform handelt, müssen im nachfolgenden Schritt die formal verifizierten Graphen auf reale Geräte abgebildet werden. Dazu werden Gerätemetadaten, die während der Kommissionierungsphase in die Ground-Truth-Datenbank aufgenommen wurden, ausgewertet.

6.3.5 Policy-basierte Berechnung von Trust Zones

Der entwickelte Algorithmus (Abbildung 6.9) zur Berechnung von Trust Zones muss sicherstellen, dass alle Anwendungen ausgeführt werden können. Das heißt, dass eine zuverlässige netzwerktechnische Konnektivität zwischen den Geräten eines Anwendungsgraphen existieren muss. Der Algorithmus berücksichtigt vermaschte Netzwerke, bei denen jedes Gerät als Netzwerkinfrastruktur (mit der Fähigkeit Frames/Pakete weiterzuleiten) betrachtet wird. Netzwerke, die auf dedizierter Netzwerkhardware beruhen bzw. aus einem Verbund aus Netzwerkgeräten und ver-

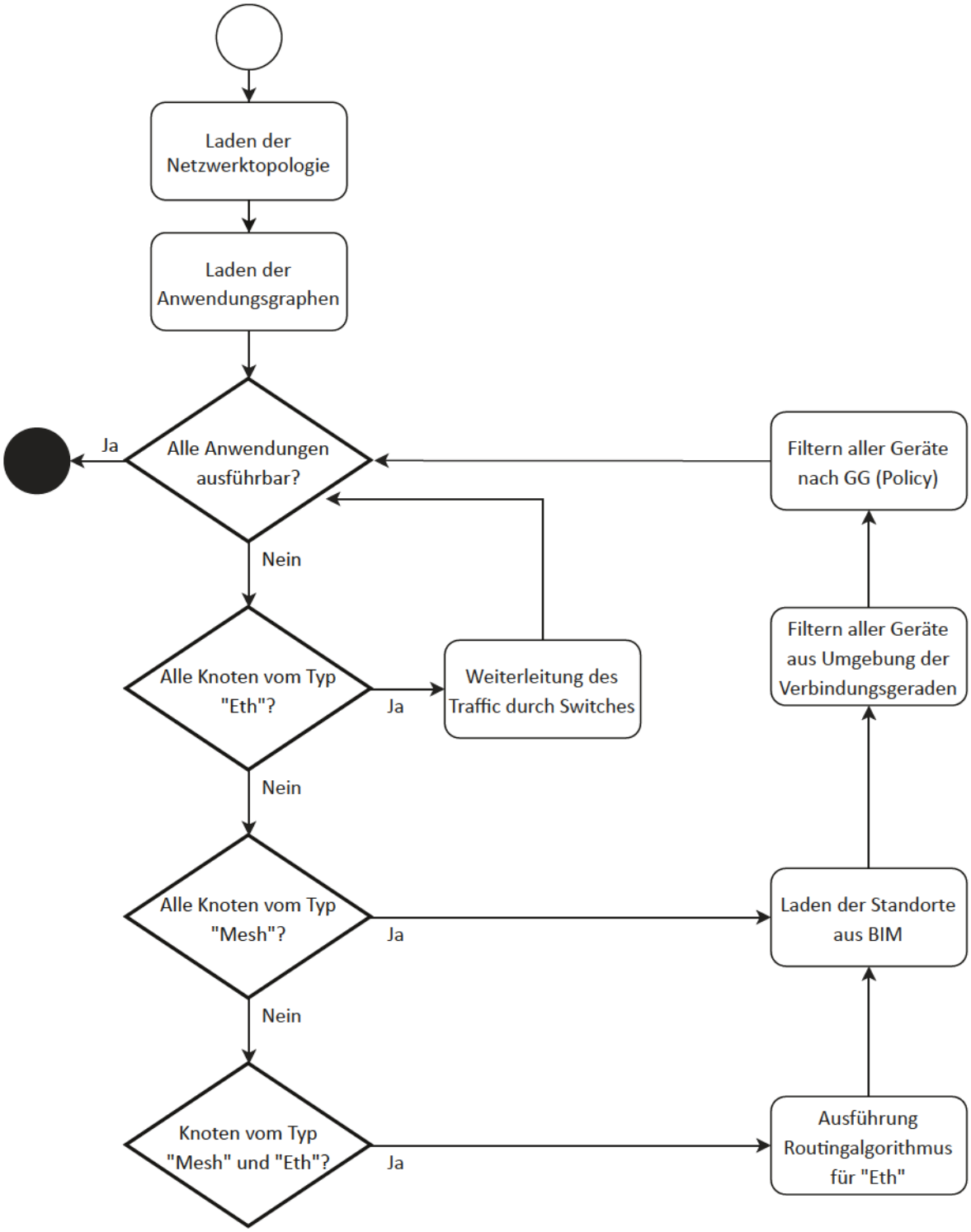

Abbildung 6.9: Policy-basierte Planung von Trust Zones [B 5]

maschten Netzwerkbereichen bestehen, können konfiguriert werden. Exemplarisch wurden reine vermaschte Netzwerke betrachtet. Basierend auf dem physischen Installationsort (gespeichert in: Ground-Truth-Datenbank) trifft der Algorithmus für jedes Gerät eine Entscheidung, ob das Gerät ein Kandidat für eine Trust Zone ist oder nicht. Security Risiken, die auf den Geräteinstallationsort zurückzuführen sind, werden für bestehende Gebäudeautomationssysteme mit statischer Netzwerkkonfiguration in der verwandten Arbeit [A 24] hervorgehoben. In dieser Arbeit dagegen fließen Standortdaten in die Berechnung von Trust Zones ein. Zur einfacheren Darstellung werden die EDs in einem zweidimensionalen kartesischen Koordinatensystem dargestellt. Die Daten lassen sich aus dem openBIM gewinnen. In [B 5] wurde ein erster naiver Ansatz vorgestellt, der den Installationsort zwischen Geräten auswertet, um zusätzliche Knoten vorzuselektieren. Es wurde die Annahme getroffen, dass Geräte in einer GA gleichverteilt platziert sind. Jeder Raum verfügt u.a. über Beleuchtung, Bewegungsmelder, Raumklimasensorik und Aktoren zur Fenster- und Türsteuerung. Dabei ist es vorgesehen, dass mehr EDs als nötig Teil der Trust Zone werden, da jedes einzelne Gerät als zusätzliche Netzwerkinfrastruktur angesehen wird. Fallen Geräte zur Weiterleitung von Frames aus, bestimmt das Routing von IEEE 802.11s automatisch eine neue Route. Somit ist eine Ausfallsicherheit gegeben. In Abbildung 6.10 ist ein einfacher Anwendungsgraph bestehend aus einer Steuereinheit, die Sensordaten von einem Sensor abruft und einen Aktor steuert, dargestellt. In der Umgebung der direkten Verbindung (ungeachtet von Hindernissen und Materialeigenschaften, die sich aus dem openBIM gewinnen lassen) zwischen zwei Geräten des Anwendungsgraphen werden alle EDs ausgewählt. Sie sind als Trust-Zone-Kandidaten zu betrachten, da jedes Gerät, das Teil der Trust Zone werden soll, den Security Policies genügen muss.

Abbildung 6.10: Ergebnis des Algorithmus zur Bestimmung von Trust-Zone-Kandidaten [B 5]

Da der Selektionsalgorithmus aus [B 5] lediglich Geräte aus der Umgebung der Streckenhalbierenden betrachtet und weitere sinnvolle Knoten nicht berücksichtigt werden, wurde eine Verbesserung eingeführt: Der modifizierte Algorithmus [6.2] betrachtet alle EDs, die sich in einem Korridor der Breite w befinden. In Abbildung 6.11 sind in einem kartesischen Koordinatensystem durch eine eigene Simulation in Java 100 EDs zufällig gleichverteilt platziert worden. Drei Geräte, die sich möglichst weit von

[6.2] Quellcode unter `https://gitlab.amd.e-technik.uni-rostock.de/Building-Automation/demo-trust-zone-formation` verfügbar

einander entfernt befinden, bilden den Anwendungsgraphen. Exemplarisch wurde die Korridorbreite w=1 m gesetzt. Die Auswahl an Trust-Zone-Kandidaten ist grau dargestellt.

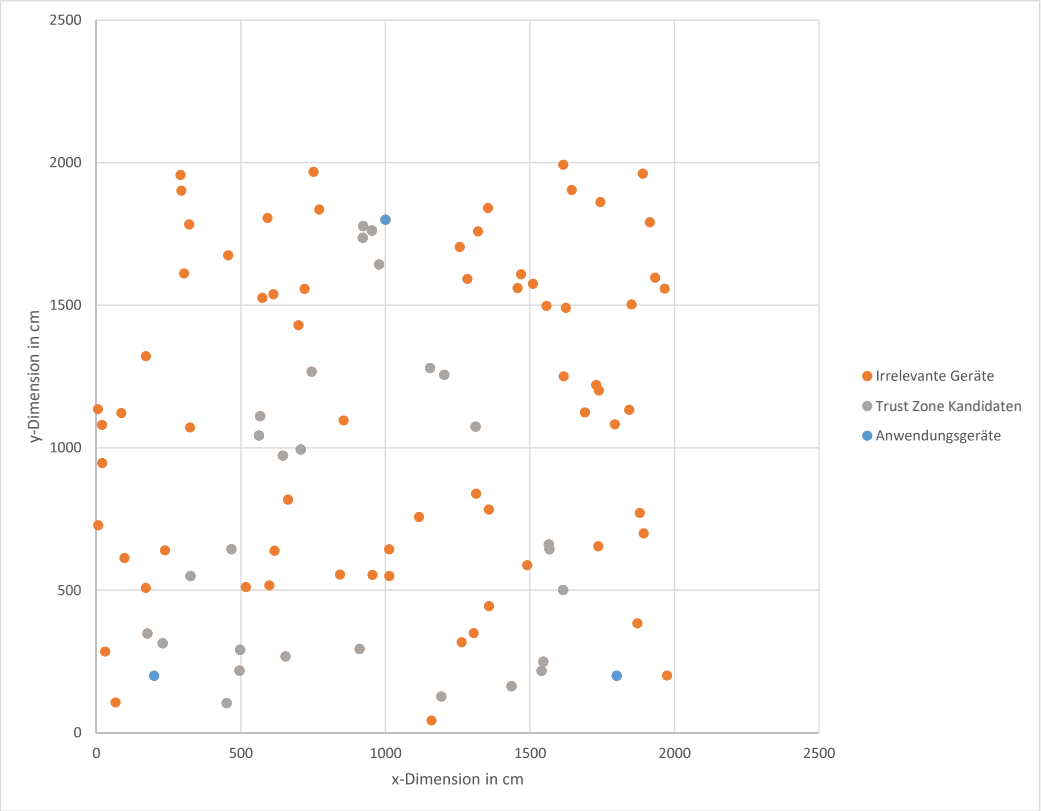

Abbildung 6.11: Modifizierter Algorithmus zur Bestimmung von Trust-Zone-Kandidaten (100 Geräte)

Um die geometrische Vorauswahl besser darstellen zu können, wurde das System auf 1000 EDs erweitert. Das Ergebnis findet sich in Abbildung 6.12.

Des Weiteren wurde der Einfluss der Breite w auf den prozentualen Anteil an EDs, die als Trust-Zone-Kandidaten markiert werden, bestimmt. Das Ergebnis findet sich in Abbildung 6.13. Jeder Messwert ist der Durchschnitt aus 1000 Durchläufen, bei denen jeweils eine neue zufällige Knotenverteilung (inkl. 3 Anwendungsgeräten) berechnet wurde. Es ist erkennbar, dass bereits bei einer kleinen Breite von 5 m ca. 60% aller EDs berücksichtigt werden. Je nach Grad der Überbestimmtheit der Trust Zone kann der Wert angepasst werden. Dies muss jedoch anhand realer Geräteinstallationen geschehen, da die Testdaten lediglich die Gleichverteilung abbilden.

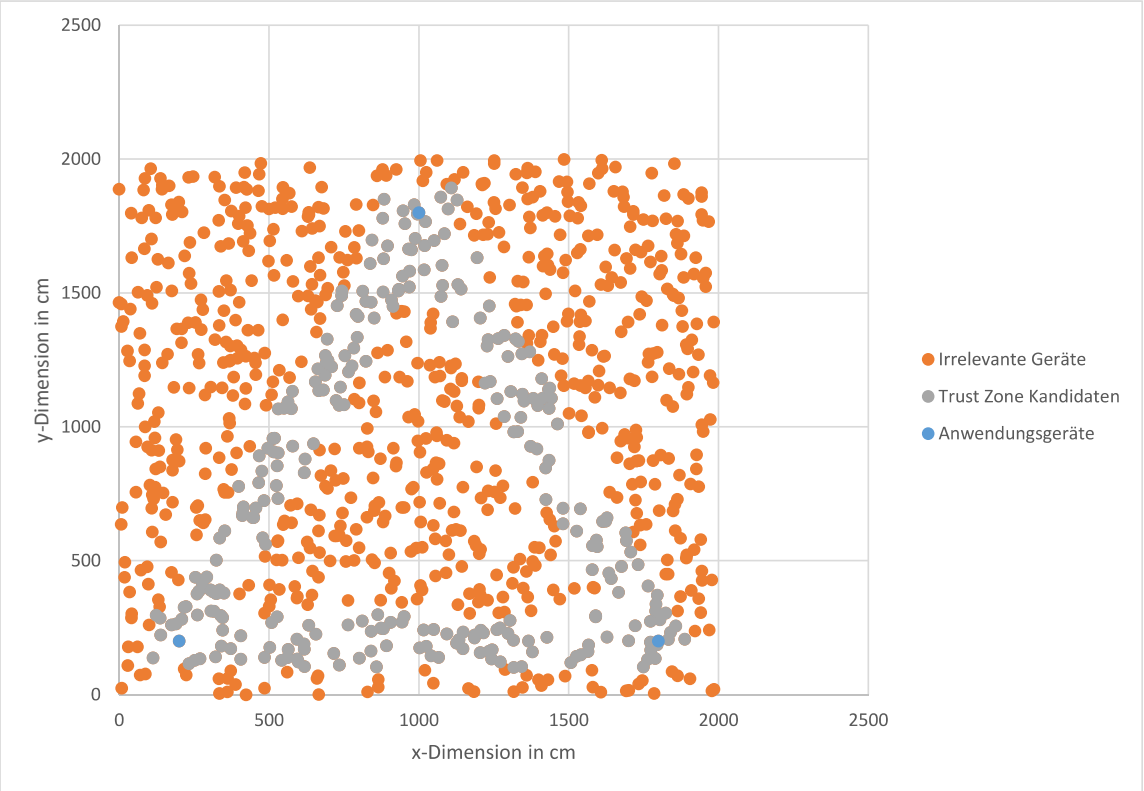

Abbildung 6.12: Modifizierter Algorithmus zur Bestimmung von Trust-Zone-Kandidaten (1000 Geräte)

Außerdem wurde die Ausführungszeit des Algorithmus gestimmt. Da jedes ED unabhängig von anderen EDs gemäß seiner Eigenschaften einer Gruppe zugeordnet wird, ergibt sich eine lineare Laufzeitkomplexität ($R^2 < 0{,}99$). Die Experimentalergebnisse aus Abbildung 6.14 belegen den linearen Zusammenhang zwischen Knotenanzahl und Ausführungszeit. Der Algorithmus wurde je auf einem Raspberry Pi 3, der der Leistungsklasse des SC entspricht und auf einem Desktop PC ausgeführt. Dabei wurde der Algorithmus je Messwert 1000-mal auf verschiedene Geräteverteilungen ausgeführt und die Ausführungszeit mittels JVM-Zeitmessung bestimmt. Der Raspberry Pi 3 benötigt ca. 200 ms, um für einen Anwendungsgraphen aus 10.000 Geräten alle relevanten Trust-Zone-Kandidaten zu bestimmen. In einem realen Anwendungsfall sind deutlich weniger als 10.000 Geräte vorhanden, sodass dieser Wert einen Extremfall darstellt. Um n Anwendungen durch den Algorithmus abzubilden, beträgt die Ausführungszeit weniger als $n * 200\,ms$. Der tatsächliche Wert liegt jedoch unter der Geraden $n * 200\,ms$, da Geräte, die eine Anwendung bilden, bereits

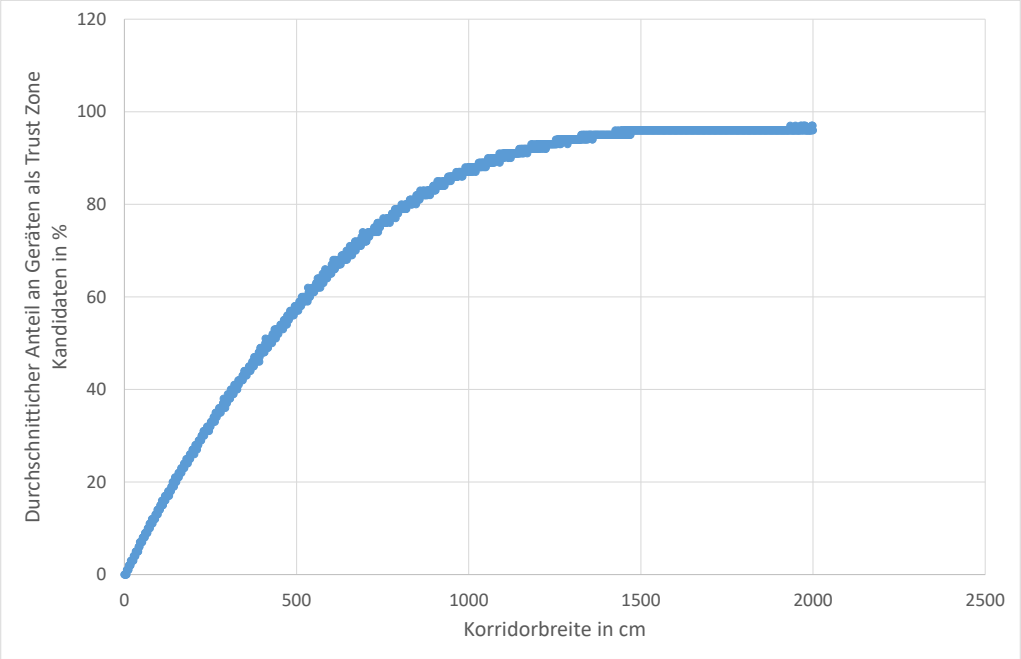

Abbildung 6.13: Prozentualer Anteil an Geräten als Trust-Zone-Kandidaten in Abhängigkeit der Korridorbreite

Teil einer Trust Zone sein können. Je mehr Anwendungen abgebildet wurden, desto höher ist die Wahrscheinlichkeit dafür.

Nach der Ermittlung der Trust Zone Kandidaten eliminiert der Algorithmus alle Geräte, die die Anforderungen der Security Policies nicht erfüllen. Je nach Anzahl und Formulierung der Policies werden mehr oder weniger EDs der Trust Zone hinzugefügt. Mögliche Security Policies sind:

- Alle Geräte müssen einer Anwendungsdomäne zugehörig sein
- Kein Gerät darf sich in einem öffentlich zugänglichen Bereich befinden
- Alle Geräte müssen die neueste Firmware-Version ausführen
- Lediglich Geräte derselben Kritikalitätsstufe dürfen miteinander kommunizieren

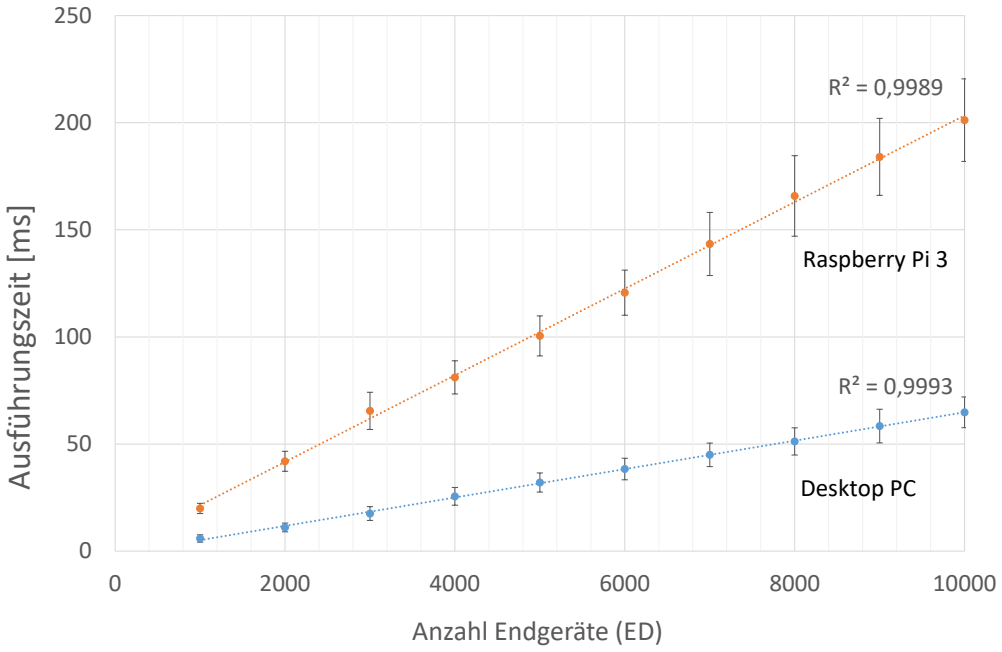

Abbildung 6.14: Ausführungszeit des Entscheidungsalgorithmus zur Bestimmung von Trust-Zone-Kandidaten [B 5]

Die einzelnen Security Policies können miteinander kombiniert und parametrisiert werden. Abbildung 6.15 zeigt das Verhältnis der ED-Klasse zur Trust-Zone-Klasse. Es werden exemplarisch zwei Objekte einer Lampe und eines Türschlosses dargestellt. Beide Geräte unterscheiden sich wesentlich in der Kritikalitätsstufe, die durch die Anwendungsdomäne bedingt ist. Für Anwendungen, die einen kleineren Kritikalitätsstufenwert aufweisen, können gemäß der entsprechenden Policy strengere Anforderungen gelten.

Basiert die Kommunikation auf einer vermaschten Netzwerktopologie, dient jedes weitere ED als Infrastruktur-Gerät, um Frames weiterzuleiten. Im Falle von IEEE 802.11s werden durch das AODV (Ad-hoc On-demand Distance Vector) Routingprotokoll dynamisch zur Laufzeit bestimmt. Fällt eine Route aus, so organisiert sich das Netzwerk ohne Zutun des Anwenders selbst. Der qualitative Zusammenhang zwischen der Strenge verschiedener Security Policies auf die Angriffsfläche einer Trust Zone ist in Abbildung 6.16 dargestellt. Je mehr EDs durch eine lockere Policy

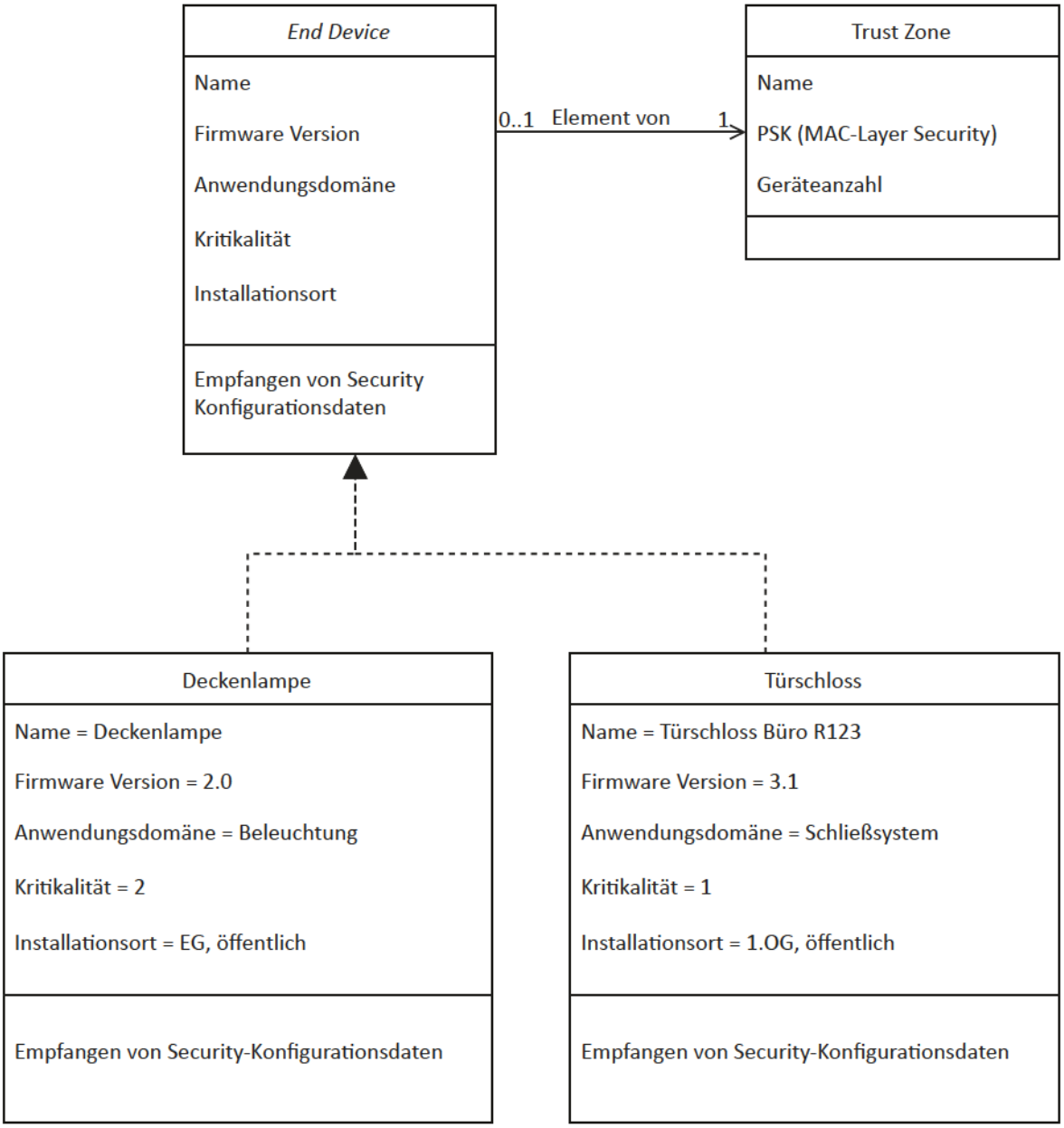

Abbildung 6.15: Beziehung zwischen ED-Klasse zu Trust Zone-Klasse mit zwei ED-Objekten, die sich wesentlich in Anwendungsdomäne und Kritikalität unterscheiden

als Infrastruktur-Knoten hinzugefügt werden, desto größer ist die MAC-Layer Ausfallsicherheit. Jedoch sinkt durch die vergrößerte Angriffsfläche das Security-Level.

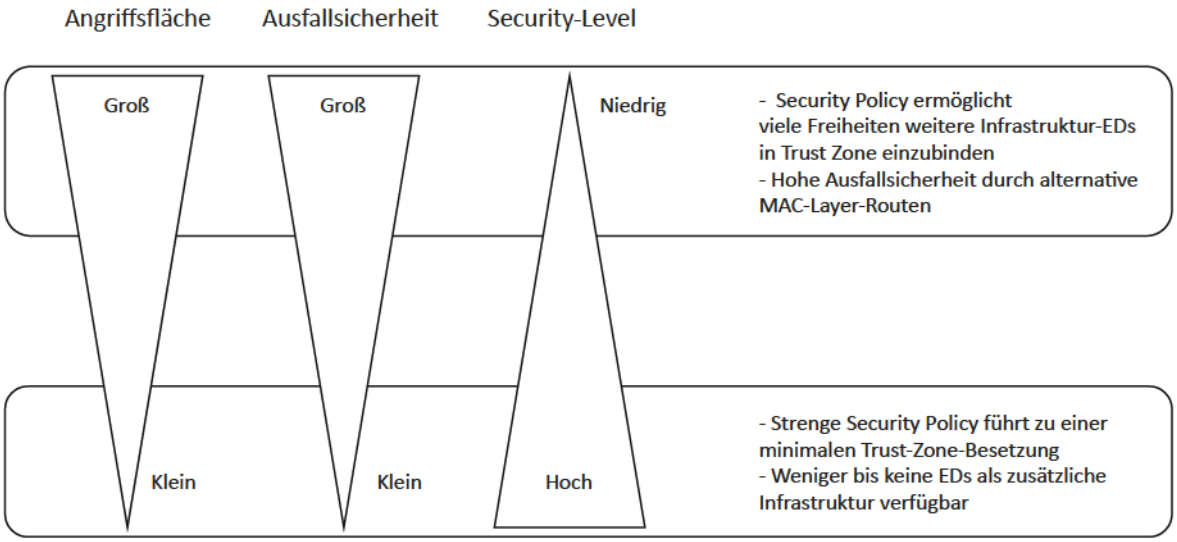

Abbildung 6.16: Zusammenhang zwischen Angriffsfläche, Ausfallsicherheit und Security-Level

6.4 Verringerung von Latenz und Energiebedarf

Sobald sich EDs in einer Trust Zone befinden, kommunizieren sie ausschließlich mit vertrauenswürdigen Geräten. Es ist somit möglich, auf eine Absicherung der Kommunikation auf Transport- oder Anwendungsebene zu verzichten. Lediglich die Absicherung der WLAN-Datenframes genügt, um Datenvertraulichkeit, -integrität und -authentizität gegenüber Manipulationsversuchen Dritter abzusichern. In der Literatur existieren verschiedene Arbeiten, die Performance-Untersuchungen von CoAP und MQTT für IoT Anwendungen durchgeführt haben [A 102] [A 103], [A 104], [A 105]. Wird auf DTLS oder OSCORE verzichtet, so reduziert sich die Latenz deutlich (Faktor 2 (OSCORE mit Handshake) bis 8 (DTLS mit Handshake)) [A 102]. Je nach Leistungsfähigkeit der Plattform sinkt die Antwortzeit auf unter 100 ms, sodass eine verbesserte Haptik erzielt wird.

Da die Daten nicht mehrfach auf MAC und Transportschicht abgesichert werden müssen, reduziert sich der Energiebedarf [A 105], [A 106]. In [A 106] führen die Autoren Energiemessungen auf dem ressourcenbeschränkten Firefly Board mit einem CC2538 Chipsatz (ARM Cortex-M3) [A 107] durch. Es zeigte sich, dass OSCORE weniger Energie benötigt als DTLS. OSCORE benötigt jedoch je nach Payload-Größe 8-28% mehr Energie als eine reine CoAP-Übertragung. Wird DTLS verwendet, so ist im Vergleich zu CoAP der Energiebedarf 17-59% höher. Außerdem

messen die Autoren unter Nutzung von OSCORE einen im Vergleich zu CoAP um 2% erhöten RAM- und einen um 12% erhöhten ROM-Bedarf. Der erhöhte ROM-Bedarf wird hauptsächlich durch benötigte Kryptographiebibliotheken verursacht. Wird DTLS verwendet, so erhöht sich im Vergleich zu CoAP der ROM-Bedarf sogar im 27% und der RAM-Bedarf um 17%. Da jedoch jedes ED als Teil des Konfigurationsnetzes über eine Ende-zu-Ende verschlüsselte Verbindung mit dem SC kommunizieren muss, kann auf eine DTLS- oder OSCORE-Implementierung nicht verzichtet werden. Basierend auf den Ergebnissen aus [A 107] ist OSCORE gegenüber DTLS vorzuziehen.

Die Autoren untersuchen in [A 108] den Energieumsatz eines WLAN-Microcontroller (ESP32). Dabei werden CoAP-Nachrichten mit und ohne DTLS übertragen. Werden die Daten lediglich auf MAC-Layer verschlüsselt, reduziert sich die Energie pro Nachricht (1024 Byte Payload) im Vergleich zu DTLS um 17%. Ein Sendevorgang einer einzelnen CoAP-Nachricht mit 1024 Byte Payload benötigt ca. 5,8 μWh. Durch Verwendung von DTLS zur Absicherung erhöht sich die Energie pro Nachricht auf 7 μWh. Außerdem muss der DTLS Handshake in der Energiebilanz berücksichtigt werden. Dabei handeln die Kommunikationspartner einen Sitzungsschlüssel aus. Je nach verwendetem kryptographischen Algorithmus variiert die Energie für einen Handshake zwischen 50 und 500 μWh.

Der Energieumsatz für die Versendung einer Nachricht ist von einer Vielzahl von Parametern abhängig. Einflussfaktoren sind: Payloadgröße, Cipher Suite, Security Protokoll, Session Dauer, Implementierung, Frame Loss Rate, Hardware Plattform und Abstände zwischen den Kommunikationspartnern. Daher sind pauschale Vorhersagen für den Energieumsatz nicht möglich. Die in diesem Kapitel aufgeführten Referenzen geben einen Überblick über die Größenordnung des Einsparpotentials.

Auf Basis der entwickelten Security-Architektur ist es möglich, optional auf Transport- oder Anwendungsschicht-Security zu verzichten. Gerätehersteller haben zur Designzeit des EDs keine Information über den Einsatzbereich und den Anwendungsgraphen. Unter bestimmten Randbedingungen (Policies) kann der SC einzelne Geräte in separierten (MAC-Schicht-Security) Trust Zones gruppieren. Das Resultat sind eine verbesserte Haptik, eine reduzierte Speicherbebelegung und ein verringerter Energiebedarf.

6.5 Weitere Security-Implikationen

Auf Basis der entwickelten Security Architektur eröffnen sich weitere Anwendungsfelder. In diesem Abschnitt werden mögliche Anwendungen skizziert, die zur Verbesserung der Security beitragen.

6.5.1 Security Controller-vermittelter Updatemechanismus

Es ist konzeptionell möglich, den SC als Vermittler von Firmwareupdates zu verwenden. Der Vorteil besteht darin, dass EDs keine Verbindung zum Internet aufbauen müssen. Somit müssen EDs keine Serverzertifikate überprüfen, um eine geschützte Verbindung via TLS zum Updateserver des Herstellers aufzubauen. Für batteriebetriebene Geräte entfällt damit das zyklische Abfragen nach Updates, was in einem geringerem Energiebedarf resultiert. Der SC kann hierbei als Vermittler fungieren. Er erhält über eine mit TLS abgesicherte Verbindung das entsprechende Binary vom Hersteller. Da eine gesicherte Verbindung über das Konfigurationsnetz zu jedem ED existiert, kann das Update über einen erweiterten Konfigurationsservice verteilt werden.

6.5.2 Sicheres Entfernen von Geräten aus dem Netzwerk

Wird ein Gerät aus dem Netzwerk entfernt, weil es z.B. ersetzt werden soll, müssen die gespeicherten Zugangsdaten als kompromittiert angesehen werden. Ein Angreifer könnte die Zugangsdaten missbrauchen, um Zugriff auf Trust Zones zu erhalten. Daher ist es notwendig das Gebäudemodell als Digital Twin zu aktualisieren. Sobald das Gerät physisch entfernt wird, bildet der SC die betroffenen Anwendungsgraphen neu auf die bestehenden EDs ab und berechnet ggf. die Trust Zones neu. Selbst wenn keine veränderte Trust-Zone-Besetzung notwendig wird, werden allen Geräten der betroffenen Trust Zone neue Zugangsdaten zugewiesen. Die potentiell öffentlichen Zugangsdaten haben für einen Angreifer keinen Nutzen.

6.5.3 Rekonfiguration durch Sicherheitsbedrohung

Verschiedene Ereignisse können den SC veranlassen, Gegenmaßnahmen zu ergreifen und das Netzwerk samt den EDs neu zu konfigurieren. Falls bekannt wurde, dass Geräte Sicherheitslücken in der Firmware aufweisen, können die betroffenen

Geräte in abgekapselte Trust Zones zur Quarantäne verschoben werden, bis eine Lösung in Form eines Updates verfügbar ist. Die Re-Konfiguration des Netzes mit neuen Zugangsdaten für die MAC-Schicht wurde bereits in ANTs [B 8] implementiert und experimentell in einem vermaschten WLAN nach IEEE 802.11s untersucht.

6.6 Implementierung des Security Controllers auf Basis des Schutzprofils Smart Meter Gateway (BSI-CC-PP-0073)

Die Sicherheit des Gesamtsystems hängt maßgeblich von der Sicherheit des SCs ab. Er bestimmt u.a. kryptographisches Schlüsselmaterial und strukturiert das Netzwerk auf Basis einer Ground-Truth-Datenbank. Daher werden besondere Anforderungen an die Hardware und Software gestellt. Der SC sollte über Hardware verfügen, die es ihm ermöglicht, Schlüsselmaterial mit hoher Entropie zu erzeugen. Weiterhin muss dieses Schlüsselmaterial jedem ED zugeordnet und sicher gespeichert werden. Zusätzlich muss der SC Metadaten über das Gebäude und die einzelnen Geräte (Typ, Anwendungsdomäne, Installationsort) speichern. Diese Daten dürfen nicht von Dritten ausgelesen und manipuliert werden. Der SC sollte im 24/7-Betrieb laufen, um jederzeit das Netzwerk rekonfigurieren zu können. Ein Ausfall des SC führt zwar zu keinem funktionalen Ausfall der GA. Jedoch ist es nicht mehr möglich, dynamisch auf Angriffe zu reagieren und einzelne Geräte in Quarantäne zu überführen. Wenn Gerätehersteller Wissen über Zero-Day Exploits haben, sollen sie in der Lage sein, diese Information dem SC zu übermitteln. Dadurch kann der SC geeignete Gegenmaßnamen, wie zum Beispiel die Isolation der betroffenen Geräte (siehe ANTs), veranlassen.

Das BSI hat mit dem Smart-Meter-Gateway-Schutzprofil ein grundlegendes Dokument veröffentlicht, um die Hardware und Software eines Smart Meter Gateway zu zertifizieren. Die Eignung des Schutzprofils zur Absicherung der Gerätekommunikation wurde im Forschungsprojekt [B 6] näher untersucht. Es besteht hinsichtlich der Anforderungen eine große Ähnlichkeit zum SC. Das Smart Meter Gateway speichert u.a. Verbrauchsdaten, ohne dass diese Daten von Dritten ausgelesen und manipuliert werden können. Des Weiteren definiert das Schutzprofil die einzelnen Kommunikationsflüsse zwischen dem SMGW, lokalen Geräten (Verbrauchszähler und Aktoren wie Blockheizkraftwerke) und dem Internet. Nach dem SMGW-Schutzprofil ist es möglich, dass autorisierte Teilnehmer einen Fernzugriff auf das SMGW erhal-

ten. Weiterhin ist das SMGW in der Lage, die Kommunikation mit lokalen Geräten zu initiieren und Steuerdaten an lokale Geräte zu senden bzw. Verbrauchsdaten abzurufen. Das SMGW-Schutzprofil definiert, welche kryptographischen Verfahren zur Absicherung der Kommunikation verwendet werden dürfen. Außerdem wird auf weitere Schutzprofile verwiesen, die spezielle Hardware zur Erzeugung von Zufallszahlen/Schlüsseln beschreiben. Dabei handelt es sich um ein Trusted Platform Module (TPM). In [B 6] wurden keine Konflikte bei der Umsetzung eines SC auf Basis des SMGW-Schutzprofils gefunden. Daher ist es möglich, den SC auf einer zertifizierten Hardware auszuführen, um einen hohen Security-Standard zu erfüllen.

6.7 Zwischenfazit

In diesem Kapitel wurde eine neuartige, auf BIM-basierende Security Architektur vorgestellt: Die Architektur umfasst alle Schritte des Produktlebenszyklus von Geräten. Basierend auf anerkannten kryptographischen Verfahren und Protokollen wird eine initiale Vertrauensbeziehung zwischen Endgeräten und dem Security Controller hergestellt. Hierbei spielt die Wissensbasis des Security Controllers eine entscheidende Rolle, das Netzwerk und die Endgeräte zu konfigurieren. In der Forschung finden sich bislang keine Arbeiten, die Gebäudeinformationsmodelle als wertvolle Datenquelle zur Netzwerkpartitionierung verwenden. Es werden universelle Security Policies mit Gebäude- und Anwendungsmodellen verknüpft, um für jedes Gerät individuelle Security Konfigurationsdaten zu berechnen. Eine weitere Besonderheit des GA-Systems ist die dezentrale Steuerungsstruktur. Dadurch ist es im Gegensatz zu anderen Arbeiten sehr einfach möglich, das Netzwerk zur Laufzeit in abgeschottete Netzwerke (Trust Zones) zu unterteilen. Durch die vorgeschlagene formale Modellierung und Verifikation der Anwendungslogik können Fehlerfälle, die die Funktionssicherheit betreffen, ausgeschlossen werden. Wenn die Funktionssicherheit der Anwendungsschicht nicht gegeben wäre, könnte ein Angreifer dies für seine Zwecke ausnutzen. Das Sicherheitsproblem aus dem Safety-Kontext würde sich zu einem Security-Problem entwickeln. Der besondere Fokus dieses Kapitels lag auf den Abläufen des Security Controllers. Die relevanten Funktionen des BIM-basierten Partitionierungsalgorithmus wurden implementiert und experimentell evaluiert. Es konnte nachgewiesen werden, dass der Partitionierungsalgorithmus eine lineare Zeitkomplexität aufweist. Daher können auch große Netze mit vielen Endgeräten konfiguriert werden. Die entwickelte Security Architektur kann außerdem zur Laufzeit auf Be-

drohungen, Angriffe und Anwendungsänderungen reagieren und Trust Zones neu berechnen. Es wurde gezeigt, nach welchem Verfahren die individuellen Konfigurationsdaten für die EDs berechnet werden. Im nachfolgenden Kapitel werden die EDs näher betrachtet. Dabei werden insbesondere Implementierungsrichtlinien für Gerätehersteller abgeleitet und experimentell evaluiert.

7 ANTs - Application-driven Network Trust Zones

In den vergangenen Jahren wurden große Anstrengungen unternommen, die Kommunikation von Geräten zu vereinheitlichen und zu standardisieren. Um Geräte, die unterschiedliche M2M-Protokolle unterstützen, zu einem gesamten System zu verbinden, haben sich Gateways etabliert, die zwischen den unterschiedlichen Nachrichtenformaten und Protokollabläufen übersetzen. Des Weiteren sind Lösungen entwickelt worden, um unterschiedliche Physical-Layer mithilfe von Gateways zu überbrücken. Durch den zunehmenden Grad an netzwerktechnischer Konnektivität entstehen Sicherheitsprobleme und -risiken. Gelingt es einem Angreifer ein einzelnes Gerät zu kontrollieren, so ist er in der Lage, innerhalb eines lokalen Netzwerks weitere Attacken gegen andere Netzwerkteilnehmer auszuführen (Abbildung 7.1). Somit ist es notwendig, die Kommunikation aller Geräte so weit wie möglich einzuschränken ohne die Anwendungsfunktionalität zu unterbinden. Das nachfolgend vorgestellte Konzept wurde in der eigenen Publikation [B 9] veröffentlicht und in [B 8] experimentell evaluiert.

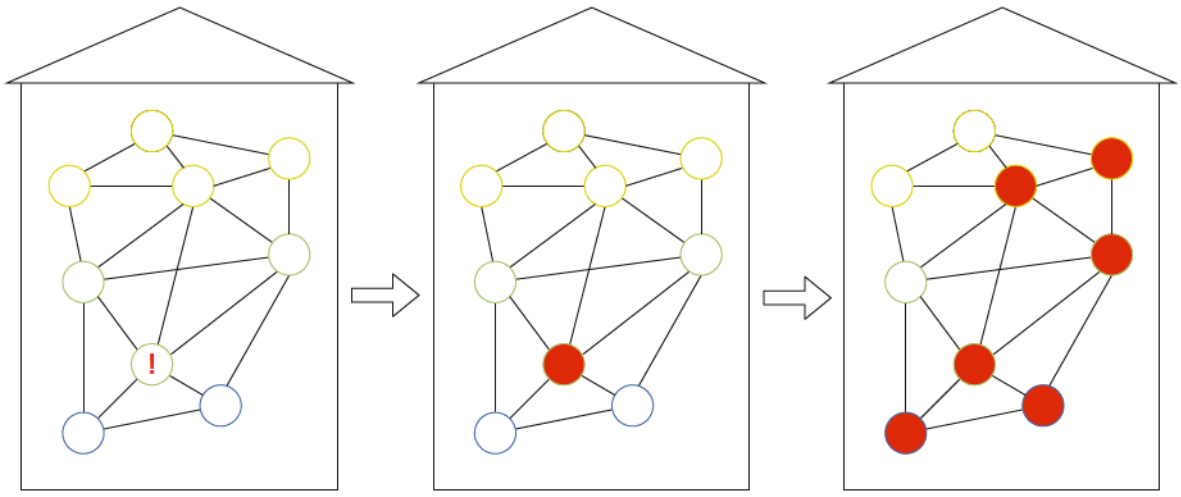

Abbildung 7.1: Ausweitung eines Angriffs auf Geräte aller Anwendungsdomänen unter Zuhilfenahme eines einzelnen kompromittierten Gerätes

7.1 Verwandte Arbeiten

Es finden sich verschiedene Arbeiten, die sich mit Vertrauensbeziehungen zwischen eingebetteten Systemen auseinandersetzen. In [A 109] werden die einzelnen Ansätze zusammengefasst. Der Vertrauensbegriff ist in der Literatur nicht einheitlich definiert. Verhalten sich zwei oder mehr Geräte nach den Protokollen, so können sie als vertrauenswürdig angesehen werden. Die Art und Weise, wie Vertrauen berechnet oder definiert wird, hängt stark vom Algorithmus ab. ”Vertrauen” (engl.: Trust) wird häufig als übergeordnete Eigenschaft betrachtet, die oberhalb der Anwendungsschicht des ISO/OSI-Modells angesiedelt ist. Allgegenwärtig sind Verfahren, wie sie beim Online-Einkauf angewendet werden. Kunden loggen sich mit ihren Nutzerdaten an einer Webanwendung ein. Die Verbindung ist in der Regel über TLS abgesichert. Dabei weist sich der Server über ein x.509-Zertifikat [A 110] aus, sodass der Browser die Authentizität sicherstellen kann. Das Vertrauen des Kunden zum Online-Shop wird u.a. durch den Ruf des Unternehmens, Kundenrezensionen und durch eigene Erfahrungen geprägt. Weiterhin ist es dem Online-Shop-Betreiber möglich, die Vertrauenswürdigkeit des Kunden durch verschiedene Parameter (z.B. pünktlicher Zahlungseingang, Bestellhistorie oder Erfahrungen anderer Händler) zu quantifizieren. Zusammengefasst beschreibt dieses Konzept, dass sich fremde Teilnehmer/Geräte durch ein übergeordnetes Verfahren absichern. Die Algorithmen des Web-of-Trust basieren auf diesem Prinzip. Sie berechnen nach verschiedenen Ansätzen zur Laufzeit für jedes Gerät oder jeden Dienst ein Vertrauensniveau [A 111], [A 112], [A 113], [A 114], [A 115], [A 116], [A 117], [A 118], [A 119]. Diese Verfahren benötigen jedoch häufig eine bestimmte Einschwingzeit. Während dieser Zeit kommunizieren die Geräte des Systems miteinander, sodass sich Vertrauensbeziehungen entwickeln können. Wird ein Gerät zur Laufzeit von einem Angreifer übernommen, sodass es arbiträren Code ausführt, sind Verfahren des Web-of-Trust häufig nicht in der Lage, adäquat zu reagieren. Erst nach einer bestimmten Anzahl von Interaktionen oder bösartigem Verhalten sinkt das Vertrauensniveau des kompromittierten Gerätes mehr oder weniger schnell. Angriffe, die passiver Natur sind, um sensible Daten aus einem Netzwerk abzugreifen, sind nur schwer oder gar nicht zu erkennen. Ein weiteres Merkmal ist der reaktive Charakter dieser Algorithmen. Erst nachdem bösartiges Verhalten vom System beobachtet wurde, werden Gegenmaßnahmen eingeleitet. Oftmals werden lediglich Anfragen von Clients auf Anwendungsschicht blockiert oder Dienste von Servern nicht mehr kontaktiert. ANTs

dagegen arbeitet proaktiv. Bevor es zum Schadenfall kommt, können Gegenmaßnahmen erfolgen. Besteht Wissen über die Existenz einer Software-Schwachstelle, können die betroffenen Geräte in eine netzwerktechnische Quarantäne überführt werden. Die Isolation der Geräte erfolgt auf MAC-Schicht, sodass ein bösartiges Gerät weitestgehend von den anderen Netzwerkteilnehmern isoliert wird.

7.2 Ansatz zur anwendungsgetriebenen Trust-Zone-Bildung

Das entwickelte Konzept zur Isolation von Geräten setzt auf der MAC-Schicht an. Nachfolgend wird dieses Verfahren mit anderen Methoden verglichen. Anschließend wird die Umsetzung anhand einer vermaschten WLAN Infrastruktur nach IEEE 802.11s beschrieben.

7.2.1 Virtuelle MAC-Interfaces zur Isolation von Kommunikationsdomänen

Geräte, die entweder als Sensoren, Aktoren oder Bedienelemente fungieren, werden im Nachfolgenden als EDs (End Devices, dt.: Endgeräte) bezeichnet. EDs kommunizieren miteinander über IP-basierte Netzwerkschnittstellen. Eine Partitionierung des Netzwerkes lässt sich auf unterschiedliche Arten implementieren. VLANs bieten die Möglichkeit, Ethernet Traffic durch einen VLAN-Switch zu isolieren. Dadurch ist ein Schutz gegeben, sodass Geräte eines VLANs nicht mit anderen VLANs kommunizieren können. VLANs werden häufig in Unternehmensnetzen eingesetzt, um Geschäftsbereiche voneinander zu trennen und ein gewisses Maß an Sicherheit zu gewährleisten. Außerdem ist es möglich, den Traffic unter Verwendung von VPNs (Virtual Private Network) zu trennen. Dies ist sowohl auf drahtgebundene als auch drahtlose Kommunikation anwendbar. Für den Einsatz auf die EDs einer GA sind VPNs jedoch schlecht geeignet, da für jedes Netz ein VPN-Server ausgeführt werden muss. Fällt dieser aus, so ist keine Kommunikation der EDs untereinander mehr möglich. Eine Abschottung auf MAC-Layer bietet den Vorteil, dass sämtliche darüberliegende Protokolle abgesichert werden. EDs können bei Bedarf ohne Weiteres über virtuelle MAC-Interfaces mit mehreren abgeschotteten Bereichen kommunizieren. Frames, die nicht aus dem entsprechenden Netz kommen, werden frühzeitig verworfen. Daher sollen im Nachfolgenden virtuelle MAC-Interfaces als Grundlage für das Sicherheitskonzept dienen. Es werden prototypische Implementierungen von IEEE 802.11s WLAN Mesh Geräten vorgestellt. Die Kommunikation

wird auf dem MAC-Layer ohne besondere Hardware/Software-Anforderungen an die EDs abgesichert.

7.2.2 Umsetzung

Geräte sollen gemäß ihrer Zugehörigkeit zu einer Anwendung in abgeschottete Netzwerkbereiche (Trust Zones) eingeordnet werden. Eine direkte Kommunikation zwischen den Geräten einer Trust Zone ist möglich, während Frames von Geräten einer anderen Trust Zone nicht empfangen werden können. Dies wird durch Abschottung auf dem MAC-Layer realisiert. Das von jedem Gerät benötigte Shared Secret wird vom SC über einen geschützten Kanal übertragen (Aufbau des geschützten Kanals: siehe Abschnitt 6.3.1 und Abbildung 6.2). Abbildung 7.2 zeigt ein Netzwerk aus EDs, die über ein vermaschtes Netzwerk miteinander verbunden sind (IEEE 802.11s WLAN-Mesh). Der SC wird durch ein Gerät implementiert, das drahtgebunden über ein sogenanntes Mesh Gate mit dem WLAN-Mesh-Netzwerk kommuniziert. Jedes ED erhält individuelle Konfigurationsdaten, um einer bestimmten Trust Zone beizutreten (Kapitel 6). In Abbildung 7.2 ist exemplarisch eine Trust Zone grün hervorgehoben. Alle anderen Geräte sind nicht in der Lage, Frames über eine fremde Trust Zone zu senden, da sie kein Schlüsselmaterial besitzen.

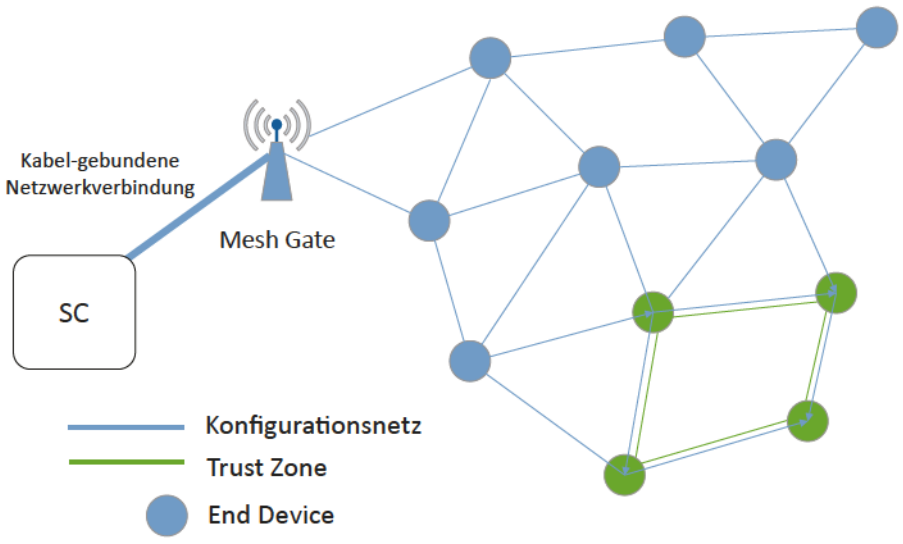

Abbildung 7.2: SC konfiguriert alle EDs via Konfigurationsnetz, damit EDs über ein separates MAC-Interface Anwendungen bereitstellen können [B 9]

Abbildung 7.3 zeigt den detaillierten Protokoll-Stack eines EDs. Das ED ist über ein separates MAC-Interface mit dem Konfigurationsnetzwerk (blau hervorgehoben) verbunden. Über das Konfigurationsnetz erhält das ED Konfigurationsdaten, um über ein weiteres MAC-Interface der Trust Zone (grün markiert) beizutreten. Um dem Konfigurationsnetz beitreten zu können, benötigt das ED Zugangsdaten, um eine auf MAC-Layer verschlüsselte Verbindung aufzubauen. Für Demonstrationszwecke wurde SAE verwendet, um die Absicherung zu realisieren. Nach Beitritt des Konfigurationsnetzes erhält das ED vom DHCP-Server die IP-Konfigurationsdaten (Punkt 2). Während der Kommissionierung des Gerätes wird zwischen SC und ED Schlüsselmaterial ausgetauscht, damit eine Absicherung (hinsichtlich Vertraulichkeit, Integrität und Authentizität) gewährleistet werden kann (Punkt 3). Dies ist erforderlich, da sich alle EDs im Konfigurationsnetz befinden und ein kompromittiertes Gerät keine vertraulichen Daten fremder Trust Zones abfangen oder manipulieren kann. Auf Anwendungsschicht (Punkt 4) führt das ED einen Konfigurationsserver aus, der lediglich Verbindungen vom SC annimmt. Somit ist eine sehr strenge Filterung (ähnlich einer Firewall) vorgesehen, um die Angriffsfläche zu reduzieren. Prinzipiell lässt sich eine Ende-zu-Ende-Sicherheit auf Anwendungsschicht gewährleisten. Wird ein Konfigurationsserver auf Basis von CoAP verwendet, bietet sich OSCORE an. Da OSCORE analog zu DTLS die PSK-Methode anbietet, bei der vorher festgelegte Schlüssel (während der Kommissionierungsphase auf sicherem Wege ausgetauscht) verwendet werden, ist ein einfacher Austausch von DTLS durch OSCORE möglich. Nach Erhalt der Konfigurationsdaten für die Trust Zone (Zugangsdaten in Form von SSID und Passwort (5), IP-Konfiguration (6) und Public-/Private-Key (7)), kann das ED seine Dienste beispielsweise in Form eines Webdienstes innerhalb der Trust Zone anbieten (Punkt 8).

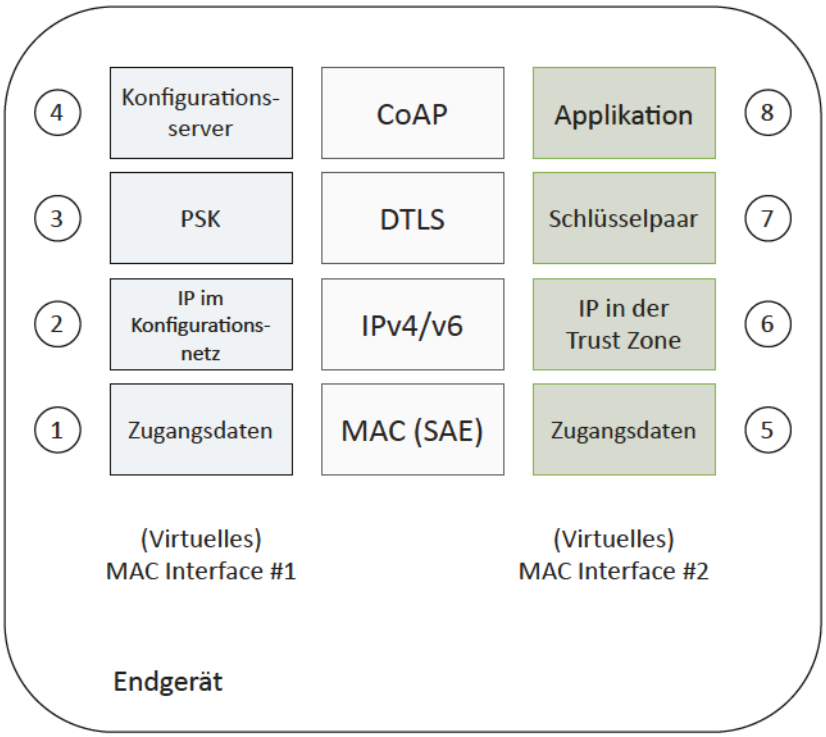

Abbildung 7.3: Virtuelle MAC-Interfaces eines EDs [B 9]

7.3 Experimentelle Evaluation

Zunächst wird der Testaufbau nach [A 120] beschrieben. Auf dieser Geräte- und Netzwerkinfrastruktur wird die im Konzept (Abschnitt 7.2) beschriebene Software ausgeführt. Vom SC werden Zugangsdaten für die MAC-Schicht verteilt und von den EDs unmittelbar nach Erhalt angewendet.

7.3.1 Testbed

Um Geräte, die an verschiedenen Orten eines Gebäudes installiert sind, innerhalb eines Labors mit ungenutzten WLAN-Kanälen zu evaluieren, wurde ein spezieller Aufbau verwendet. Der experimentelle Testaufbau nach [A 120] besteht aus 25 WLAN-Geräten, die über eine Mesh-Topologie miteinander verbunden sind (Abbildung 7.4). Die Geräte sind in einem 5x5 Knoten Gitter angeordnet. Jedes Gerät baut einen Link zu jedem benachbarten Knoten auf. Somit steht jedes Gerät mit bis zu acht anderen Teilnehmern in Verbindung. Geräte, die sich an einer übernächsten Position befinden, sind nicht direkt, sondern über eine 2-Hop-Verbindung

erreichbar. Damit reale Multihop-Pfade in einem verkleinerten räumlichen Maßstab realisiert werden können, beschreibt [A 120] die Konfiguration der einzelnen Geräte: Jedes Gerät basiert auf einem Intel Galileo Ein-Platinen-Computer, der ein Debian 8 Linux ausführt. Der Kernel Version 3.13 führt mithilfe von Linux Backports die Netzwerktreiber der Version 4.2.9 aus [A 121]. Sämtliche Geräte werden über das on-board Fast Ethernet Interface mit einem Steuernetzwerk verbunden. Jedes Gerät besitzt eine mPCIe WLAN-Karte des Typs Compex WLE200NX, welche dual-Band WLAN IEEE 802.11abgn und zwei Antennen unterstützen. Als WLAN-Kartentreiber wurde der ath9k-Treiber verwendet. Das Linux Kernelmodul mac80211 implementiert softwaretechnisch die MAC-Schicht nach IEEE 802.11 samt Mesh-Erweiterungen nach IEEE 802.11s [A 122]. Der 2x2:2 MIMO-Modus wird von der Netzwerkkarte unterstützt. Im Testaufbau wurde eine 2x2:1-Konfiguration mit Sende-/Empfangs-Diversität und einem Spatial Stream (dt.: räumlicher Datenstrom) angewendet. Um Störungen durch andere Funknetzwerke zu vermeiden, wurde Kanal 149 (5,745 GHz, 20 MHz Kanalbandbreite, HT-Modus) im 5 GHz-Bereich verwendet. Zusätzlich wird jede Antenne der Netzwerkkarten mit einem Dämpfungsglied ausgestattet, um die Sende- und Empfangsleistung zu reduzieren. Weiterhin wird die Sendeleistung der WLAN-Karte auf einen reduzierten Wert festgesetzt. Durch die Wahl eines festen MCS-Wertes (Modulation and Coding Scheme) wird ein definierter Satz an Sendesymbolen eingestellt, da unterschiedliche MCS-Werte zu unterschiedlichen Symbolen der digitalen Datenübertragung führen und somit zu einer verschiedenen Energie pro Symbol. Das von den gesendeten Symbolen abhängige Signal-Rausch-Verhältnis am Empfänger beeinflusst die Reichweite einer Single-Hop Strecke. Nach [A 120] wurde MCS 3 mit einer 16-QAM-Modulation und 1/2 FEC Kodierungsschema angewendet, um Single-Hop Übertragungen lediglich zu räumlich benachbarten Knoten zu gewährleisten. Sämtliche Parameter des Testaufbaus finden sich in Tabelle 7.1. Jeder Knoten befindet sich im horizontalen und vertikalen Abstand von 20 cm zu benachbarten Knoten. Diagonal benachbarte Knoten haben einen Abstand von ca. 33 cm zueinander. Die maximale Link-Distanz liegt im Intervall 33 cm bis 40 cm, um direkte Verbindungen (Single-Hops) zu nicht benachbarten Knoten zu unterbinden. Sollen Frames zu nicht benachbarten Knoten gesendet werden, wird das on-demand Routingprotokoll HWMP ausgeführt um entsprechende Pfade zu ermitteln.

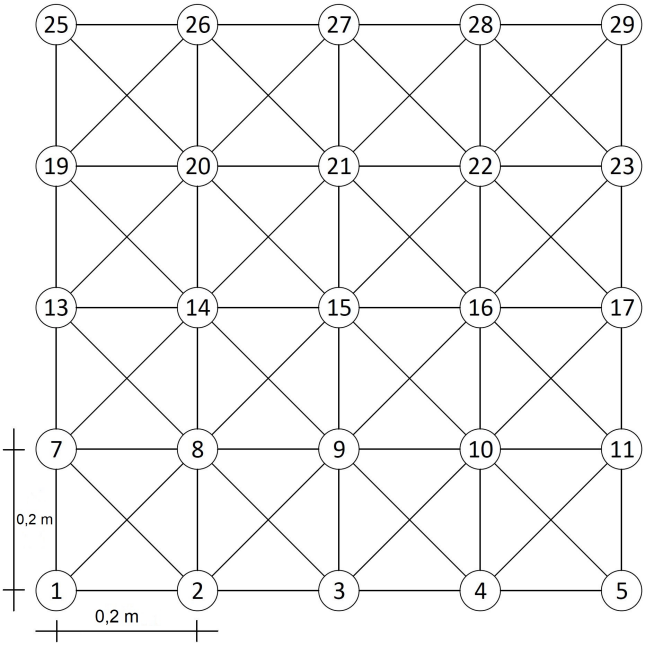

Abbildung 7.4: Räumliche Anordnung des Testaufbaus [A 123]

Tabelle 7.1: Konfiguration des Testaufbaus [A 123]

Parameter	**Wert**
Gerät	Intel Galileo Gen. 1
Prozessor	Quark X1000 400 MHz
RAM	256 MB DDR3
Betriebssystem	Debian 8 Linux (Kernel v4.9.27)
WLAN NIC	Compex WLE200NX mPCIe
WLAN NIC Chipset	Atheros AR9280
WLAN Treiber	ath9k
Antennen	2 x 5 dBi Omni-Directional Dual-Band
Antennen Mapping	2x2:1 (RX/TX Diversity)
Verbindungskabel	2 x 20 cm U.FL-RP-SMA
Dämpfungsglied	2 x Mini-Circuits VAT-30+ 30 dB
WLAN Kanal	149 (5745 MHz)
Kanalbandbreite	20 MHz (HT Mode)
Guard Interval	Long (800 ns)
Senderate	MCS 3 (QAM-16 1/2 FEC)
Maximale Linkdistanz (d)	0,3 m < d < 0,4 m

7.3.2 Verteilung von Konfigurationsdaten

Anhand einer prototypischen Implementierung wird die Verteilung individueller Konfigurationsdaten durch einen Knoten, dem SC, an andere Netzwerkteilnehmer gezeigt. Zunächst sind alle Endgeräte über ein dediziertes MAC-Interface mit einem Konfigurationsnetz verbunden. Jedes Endgerät führt einen Konfigurationsserver aus, der die individuellen Zugangsdaten für den Netzzugang vom SC erhält. Der Security Controller versendet per Unicast die Konfigurationsdaten (JSON-Datenformat der AuthSAE-Konfigurationsdatei in Listing 1 abgebildet), die zum Beitritt einer spezifischen MAC-Layer Trust Zone benötigt werden. Nach Erhalt der Daten baut jedes Endgerät über ein separates MAC-Interface eine Verbindung zur jeweiligen Trust Zone auf.

Abbildung 7.7 zeigt den verwendeten Software Stack eines Endgerätes. Die Komponenten, die zum Konfigurationsnetzwerk gehören, sind blau unterlegt. Jedes Endgerät führt einen Konfigurationsserver, der im Experiment mit jCoAP [A 67] implementiert wurde, aus. Über eine RESTful-API bieten die Endgeräte zwei Ressourcen an: ”/Wifi_Parameters”, um SSID und PSK zu erhalten und ”/Ping” um die erfolgreiche Anwendung der Zugangsdaten und die Konnektivität zu validieren (Abbildung 7.6). Der SC führt den Zustandsautomaten nach Abbildung 7.5 aus. Er liest die SSID und den PSK von der Kommandozeile ein, überträgt diese Zugangsdaten in die lokale Konfigurationsdatei des Mesh-Daemon und verschlüsselt den PSK für jedes Endgerät individuell. Somit ist sichergestellt, dass Zugangsdaten auf dem Übertragungsweg nicht verändert werden können und die Authentizität gewährleistet wird. Anschließend sendet der SC per CoAP-Client (unter Nutzung von jCoAP implementiert) die Zugangsdaten für die einzelnen Anwendungs-Interfaces an jedes Endgerät. Nachdem ein Endgerät die Daten erhalten hat, wird nach erfolgreicher Entschlüsselung und Prüfung der Absenderauthentizität die eigene Mesh-Daemon-Konfiguration aktualisiert. Unmittelbar darauf wird der alte Mesh-Daemon-Prozess beendet, um durch einen Neustart des Mesh-Daemon die neue Konfiguration anzuwenden. Im Versuchsaufbau wird die MAC-Layer-Security durch SAE realisiert [A 97]. Der ”Mesh Daemon” Prozess wird im Linux Userspace ausgeführt [A 124]. Er implementiert SAE, um über das mac80211 Kernel-Modul verschlüsselte Frames zu versenden und zu empfangen. Der Konfigurationsservice befindet sich im hörenden Zustand auf dem virtuellen Netzwerkinterface, das den Zugang zum Konfigurationsnetzwerk hat. Der CoAP-Client sendet einen einzelnen POST Request an

die RESTful-API. Dies geschieht per Unicast-Nachricht, um an jedes Gerät die individuellen Zugangsdaten zu senden. Da CoAP auf UDP basiert, wird der CoAP POST Request als Confirmable-Nachricht abgesendet, um durch ein ACK des Empfängers eine zuverlässige Nachrichtenübertragung zu gewährleisten [A 125]. In der Implementierung wurden die folgenden Standardparameter für den CoAP-Sicherungsmechanismus verwendet und evaluiert: ACK_RANDOM_FACTOR = 1,5 und MAX_RETRANSMIT = 4. Bleibt das ACK aus, so wird die Nachricht nach Ablaufen des ACK_TIMEOUT neu übertragen. Mit jedem fehlgeschlagenen Versuch verlängert sich das Timeout auf den Wert ACK_TIMEOUT * ACK_RANDOM_FACTOR, um Burst-Effekte im Netzwerk zu vermeiden. Der SC befindet sich auf Position ”1” im Mesh-Knotenraster nach Abbildung 7.4. An allen anderen Positionen befindet sich je ein Endgerät. Diese Anordnung stellt den Worst-Case-Anwendungsfall dar, der zu den längsten Pfaden zwischen SC und allen Endgeräten führt. Die Wahrscheinlichkeit, dass eine Nachricht erfolgreich übertragen wird, ist am niedrigsten. Der SC loggt die Anzahl aller CoAP-Neuübertragungen auf Anwendungsebene. Während insgesamt 24 Versuchsdurchläufen waren 75 % der Übertragungen beim ersten Versuch erfolgreich. Die restlichen 25 % der Übertragungen waren nach zwei bis fünf Versuchen erfolgreich. Der SC kontaktiert nach erfolgreicher Konfiguration eines Endgeräts das nächste Endgerät. Nach diesem Modus wurde neben der Anzahl an Neuübertragungen die Konfigurationsdauer des gesamten Netzes vom SC gemessen. Im Durchschnitt wurde das Netzwerk bestehend aus 24 Endgeräten innerhalb von 4,6 Sekunden konfiguriert. Nachdem der SC alle ACK-Nachrichten, die den Erhalt der Zugangsdaten quittieren, empfangen hat, wird eine GET-Anfrage an die jeweilige Ping-Ressource gesendet. Dadurch prüft der SC, ob jedes Endgerät die neuen Zugangsdaten erfolgreich anwenden konnte und dem rekonfigurierten Netzwerk beigetreten ist.

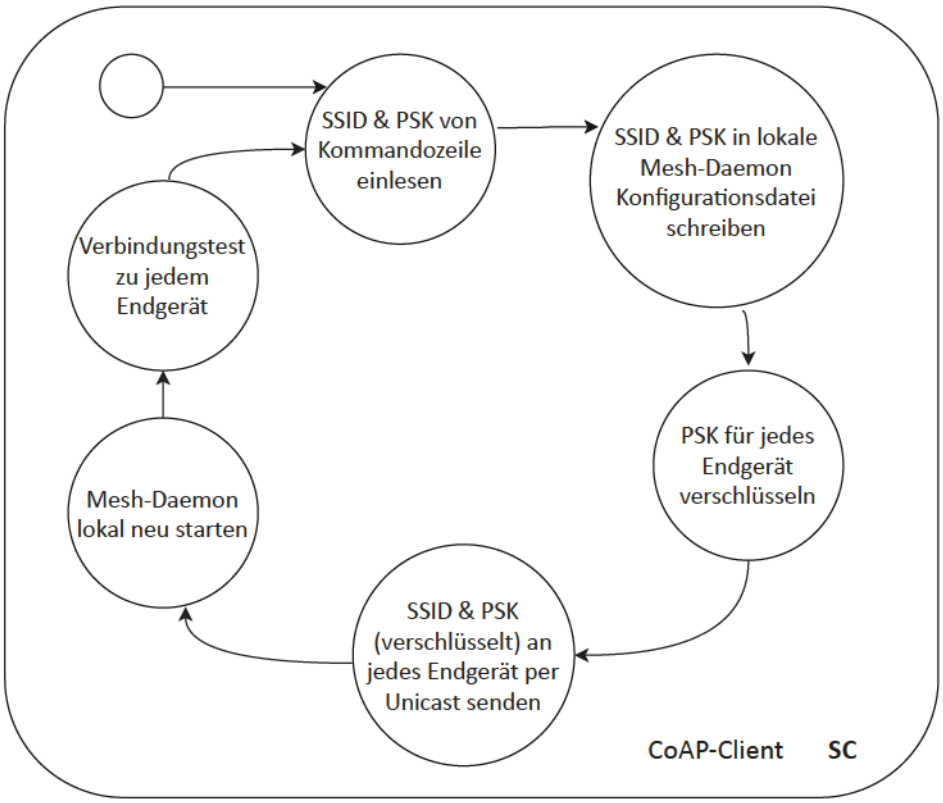

Abbildung 7.5: Zustandsautomat des SC

```
authsae: {
 sae:
  {
    password = "0x48BA1F9753E3A19F";
  };
 meshd:
  {
    meshid = "Trust Zone #0"
    interface = "sec0";
    channel = 6;
  };
};
```

Listing 1: Auszug der AuthSAE-Konfigurationsdatei

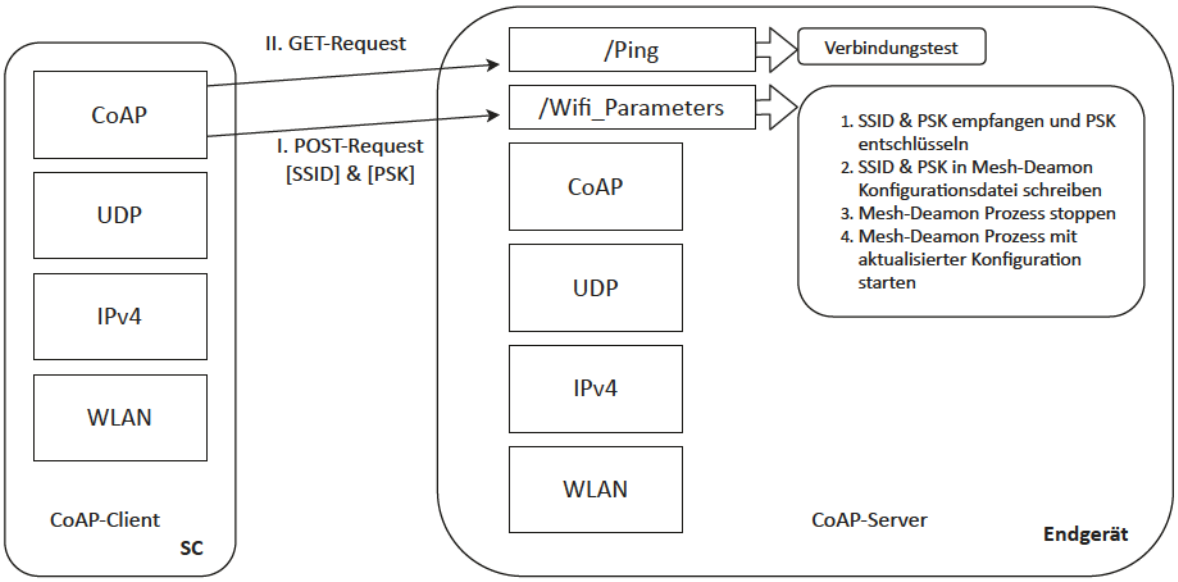

Abbildung 7.6: Zusammenspiel zwischen CoAP-Client des SC und CoAP-Server des Endgerätes

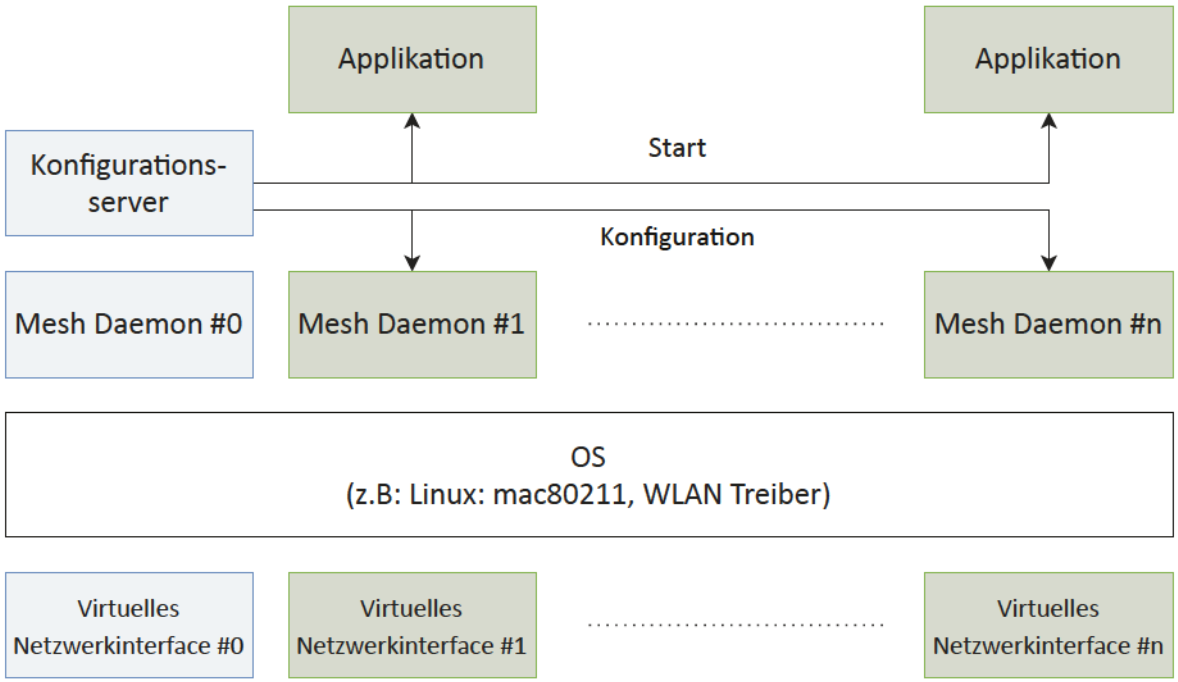

Abbildung 7.7: Softwarestack nach ANTs [B 9]

7.3.3 Performance-Evaluation virtueller MAC-Interfaces

In einem weiteren Experiment wurden mehrere virtuelle MAC-Interfaces auf einem Endgerät evaluiert. Dabei wurde der jeweilige Datendurchsatz gemessen. ANTs

sieht die Verwendung von mindestens zwei virtuellen MAC-Interfaces pro Gerät vor. Dabei konkurrieren die Interfaces um den WLAN-Kanalzugriff. Für Anwendungen mit sporadischem Traffic-Muster, wie sie beispielsweise bei Bewegungs- und Rauchmeldern auftreten, stellen parallele virtuelle MAC-Interfaces kein funktionales Problem dar. Jedoch können bei Anwendungen mit hohen Datenraten Konflikte hinsichtlich der Bandbreitenzuteilung der einzelnen Interfaces auftreten. Befindet sich eine Überwachungskamera über zwei Interfaces in zwei unabhängigen Trust Zones, so muss für beide Videodaten-Streams vom Linux-Interface-Scheduling eine faire Datenratenaufteilung erfolgen. Im Versuchsaufbau werden zwei Intel Galileo Boards mit den WLAN-Einstellungen nach Tabelle 7.1 konfiguriert. Lediglich wurde der MCS-Wert angepasst. Beide Geräte bauen eine Single-Hop-Verbindungen zueinander auf. Der MCS-Wert wurde auf den Wert 4 (QAM-16 1/2 FEC) gesetzt, wodurch sich bei einer Kanalbreite von 20 MHz im HT Modus eine feste Bruttodatenrate von 39 Mbit/s ergibt. Nach [A 123] ist MCS 3 das höchste Codierungs-Schema, welches auf der Galileo-Plattform noch mit Leistungsreserven genutzt werden kann.

In drei verschiedenen Testfällen produziert das Tool iperf der Version 2.0.5 einen UDP- bzw. TCP-basierten Datenstrom. Iperf wurde mit 8 MB Puffergröße, 1470 Byte/Datagramm Nachrichtengröße und einer anvisierten Datenrate von 100 Mbit/s für die UDP-Datenübertragung eingestellt. Die TCP-Nachrichtengrößen und Neuübertragungscharakteristiken folgen den Linux-Standardeinstellungen. Eine Zieldatenrate wurde während der TCP-Übertragung nicht festgelegt, da iperf versucht, im TCP-Modus die maximale Datenrate zu erzielen. Jeder Testfall betrachtet zusätzlich den Einfluss von SAE. Als Referenz dient der erste Testfall nach Abbildung 7.8. Über UDP bzw. TCP wird ein Datenstrom über ein MAC-Interface durch einen einzelnen Anwendungsprozess (iperf-Server) an ein anderes Gerät gesendet, das ebenfalls über ein einzelnes MAC-Interface und einen Anwendungsprozess (iperf-Client) die Daten empfängt. Nach mehreren Versuchsdurchläufen, bei denen jeweils SAE verwendet bzw. abgeschaltet wurde, wurden die erzielten Datenraten ermittelt. Im zweiten Testfall senden acht Anwendungsprozesse über ein MAC-Interface Daten an das andere Gerät, auf dem sich acht Datensenken befinden (Abbildung 7.9). In Abbildung 7.10 ist der dritte Testfall dargestellt, um acht unabhängige Datenströme über acht virtuelle MAC-Interfaces eines physikalischen Netzwerkinterfaces zu senden bzw. zu empfangen.

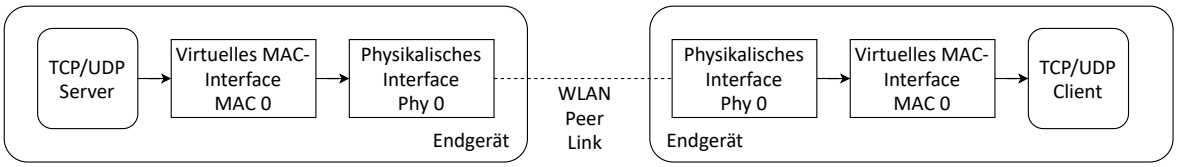

Abbildung 7.8: Referenztestfall 1 - Eine Anwendung (ein Server-Client-Paar) kommuniziert über ein virtuelles MAC-Interface [B 9]

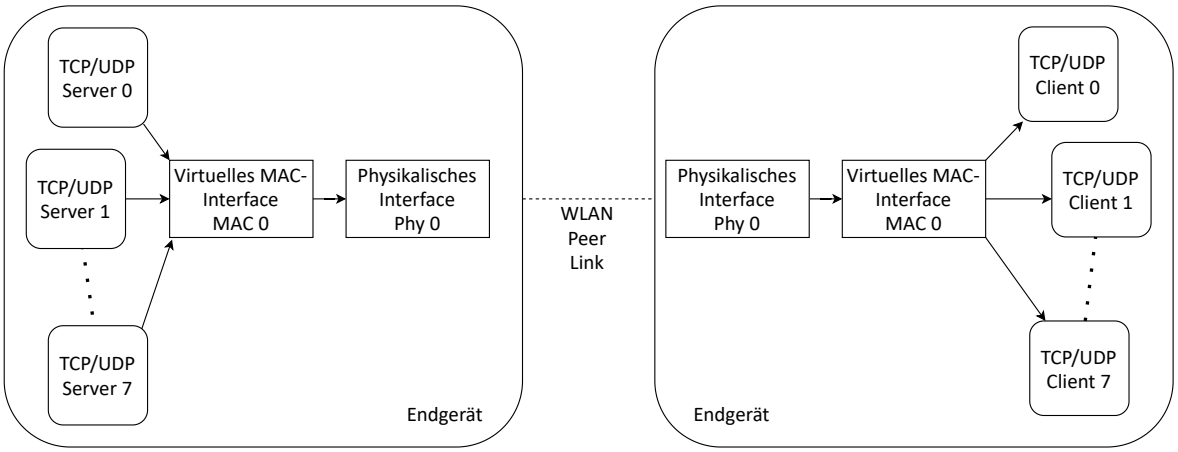

Abbildung 7.9: Testfall 2 - Acht Anwendungen (acht Server-Client-Paare) kommunizieren über ein virtuelles MAC-Interface [B 9]

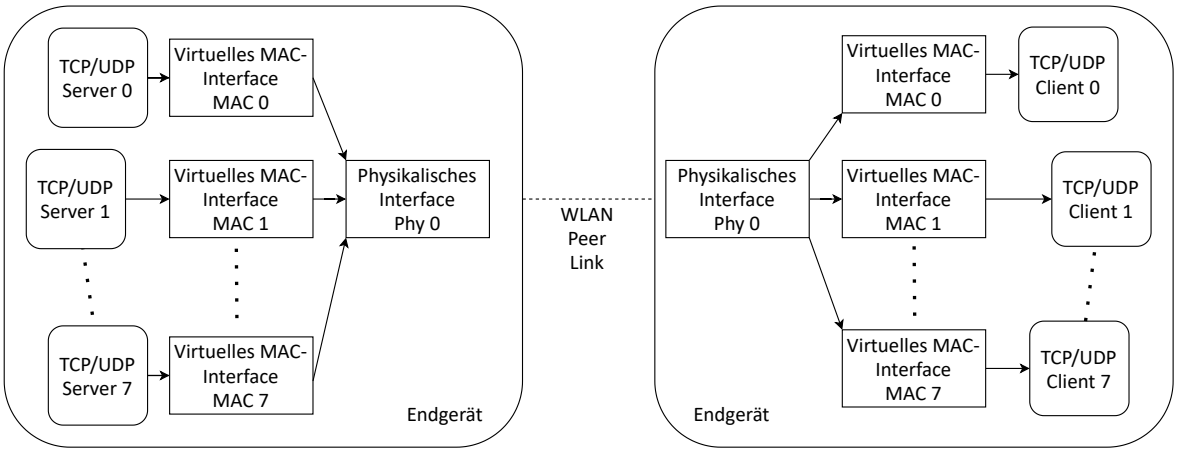

Abbildung 7.10: Testfall 3 - Acht Anwendung (acht Server-Client-Paare) kommunizieren über acht virtuelle MAC-Interfaces [B 9]

Die durchschnittlichen Datenraten inkl. Standardabweichung sind in Abbildung 7.11 dargestellt. Im Referenztest mit UDP liegt die erzielte Nettodatenrate ohne Absicherung durch SAE bei 31,6 Mbit/s. Mit SAE reduziert sich die Nettodatenrate auf 17,2 Mbit/s. Mit TCP wurden im Referenztest 23,2 Mbit/s ohne SAE und 14,2 Mbit/s

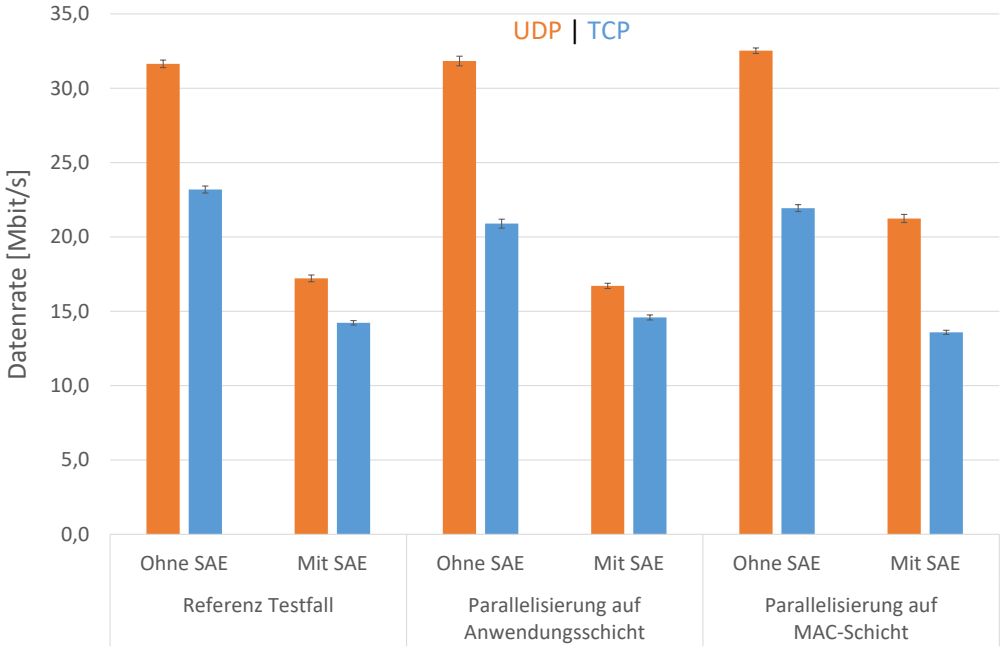

Abbildung 7.11: Vergleich der UDP- und TCP-Datenraten bei virtuellen MAC-Interfaces zu parallelen Streams auf Anwendungsschicht [B 9]

mit SAE erzielt. Werden alle Datenströme über ein MAC-Interface gesendet (Parallelisierung auf Anwendungsschicht), so werden ähnliche Datenraten in allen Varianten (UDP/TCP, mit/ohne SAE) erziehlt. Dabei wird die Summe der Datenraten der einzelnen Datenströme addiert. Die auf MAC-Schicht parallelisierten Datenströme zeigen ebenfalls keine Einbußen bezüglich der Gesamtdatenrate. Mit den durchgeführten Experimenten konnte gezeigt werden, dass das Scheduling der einzelnen Anwendungen und Datenströme keinen negativen Einfluss auf den gesamten Datendurchsatz hat. Somit ist die in ANTs vorgestellte Separation von Anwendungen auf MAC-Schicht geeignet, selbst Geräte mit höheren Anforderungen an den Datendurchsatz zu implementieren.

7.4 Angriffsflächen und Risikobewertung

Übernimmt ein Angreifer die Kontrolle eines EDs, hat er Zugriff auf alle anderen Geräte einer Trust Zone. Da jedes ED zudem Teil des Konfigurationsnetzes ist, ist eine Kommunikation zu allen anderen EDs prinzipiell möglich. Jedoch lässt sich durch harte Filterregeln, die lediglich Verbindungen durch den SC zulassen, die Angriffsfläche im Konfigurationsnetz deutlich verringern. Nachfolgend werden die Angriffsflächen näher betrachtet, die ein Angreifer hat, um Schaden außerhalb der eigenen Trust Zone anzurichten. Kontrolliert der Angreifer die Software eines EDs, lassen sich Angriffe auf WLAN-Kanäle anderer Trust Zones ausführen. Außerdem sind Seitenkanalangriffe denkbar, bei denen ein Informationskanal zwischen zwei Trust Zones entstehen könnte.

7.4.1 Zuverlässigkeit bei gestörtem WLAN-Kanal

Da ein Remote-Angreifer mithilfe eines kompromittierten Endgerätes die Kommunikation anderer Teilnehmer stören kann, wird die Zuverlässigkeit eines gestörten WLAN-Kanals näher untersucht. Ein kompromittiertes Gerät führt die Software des Angreifers aus (Remote Code Execution), um durch Jamming (dem gezielten Belegen eines WLAN-Kanals) die Kommunikation zu stören. Die Firmware der Netzwerkkarte wird als unversehrt betrachtet. Dem Remote-Angreifer ist es lediglich möglich, IEEE 802.11-konforme Frames zu versenden. Dadurch müsste ein Angreifer präsent sein, um durch spezielle Hardware Störsignale zu senden. Ein Angriff auf beliebig viele Gebäudeautomationssysteme würde schlecht skalieren. Ziel des Angreifers ist die Sabotage der Kommunikation zwischen SC und Endgeräten, um die Konfiguration zu verhindern. Daher wird untersucht, wie wirksam eine solche Kanalbelegung ist, um das Versenden von Konfigurationsdaten per Unicast an ein Endgerät zu unterbinden. Im Speziellen wird dabei CoAP untersucht, um Konfigurationsdaten innerhalb eines Frames ohne Fragmentierung zu übertragen. Des Weiteren wird die Sicherungsschicht von CoAP untersucht. Dazu wurde eine CoAP-Serveranwendung in C programmiert. Als Basis dient die CoAP-C-Implementierung libcoap [A 69]. Als CoAP-Client wird der in [A 69] enthaltene Client verwendet. Der Client abonniert eine Ressource des Servers, die zyklisch im Sekundentakt aktualisiert wird. Mit jeder Aktualisierung sendet der Server eine einzelne CoAP-Notification mit der aktuellen Uhrzeit. Die Notification wird als Confirmable-Nachricht übermittelt, sodass der Client den Erhalt mit einer ACK-Nachricht quittiert. Bleibt die ACK-Nachricht

aus, so sendet der Server die Notification erneut. Die Payload samt Message ID werden während des Experimentes protokolliert, sodass Aussagen über die Anzahl an Nachrichtenverlusten getroffen werden können. Der Kommunikationsablauf zwischen Server und Client ist in Abbildung 7.12 dargestellt.

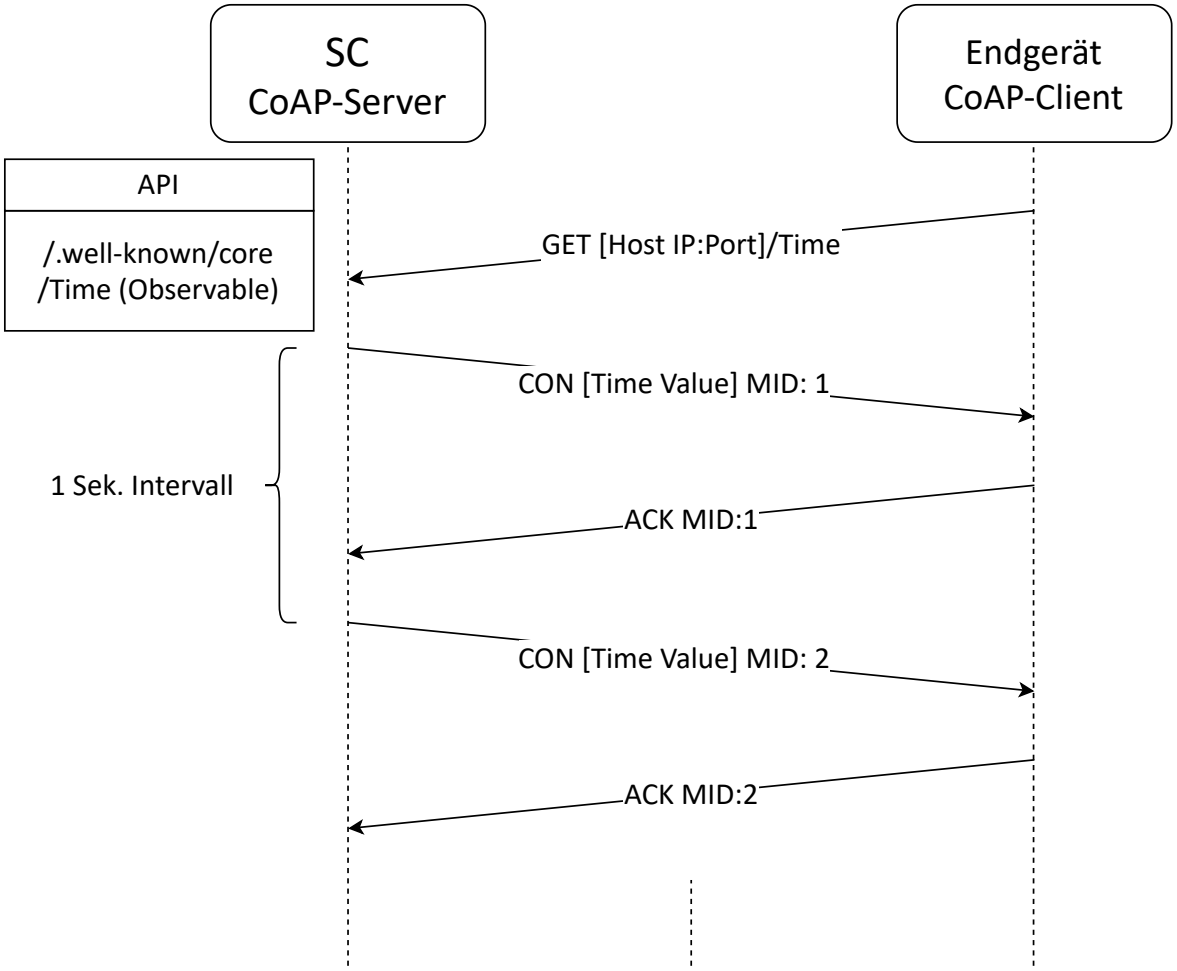

Abbildung 7.12: Nutzdatenübertragung zwischen SC und Endgerät

Ein Angreifer, der ein Gerät vollständig kontrolliert, kann nach Belieben WLAN-Frames senden. Folgende Annahme wird getroffen: Es ist dem Angreifer nicht möglich, die WLAN-Protokollimplementierung des physikalischen Interfaces zu verändern, um noch effektivere Störungen auf die Funkkommunikation auszuüben. In zwei Testfällen wird die Stabilität der WLAN-Kommunikation zwischen Server und Client während eines Angriffs evaluiert. Im ersten Testfall macht sich der Angreifer das Hidden-Station-Problem zunutze, um Frame-Kollisionen herbeizuführen. Bevor

jede Station nach IEEE 802.11 sendet, ”hört” sie auf physikalischer Ebene, ob das Kommunikationsmedium belegt ist. Ist das Medium frei, beginnt nach einer zufälligen Wartezeit der Sendevorgang. Beim Hidden-Station-Problem (Abbildung 7.13) wird das Medium vom Sender Alice fälschlicherweise als unbelegt angesehen. In Wirklichkeit sendet der Angreifer Mallory Frames, die von Empfänger Bob gehört werden. Die von Alice gesendeten Frames kollidieren somit bei Bob. Durch besondere Clear-to-send und Request-to-send Frames kann das Hidden-Station-Problem gelöst werden. Jedoch ist dieser Mechanismus häufig deaktiviert. Nach Ausführung des Experimentes über 12 Stunden wurden 43.058 Nachrichten bzw. Frames übermittelt. Davon wurden lediglich acht Frames ein zweites Mal nach einer zufälligen Wartezeit zwischen ein bis drei Sekunden (nach CoAP Default-Konfiguration) erneut gesendet. Diese Nachrichten wurden spätestens beim zweiten Sendeversuch erfolgreich übertragen.

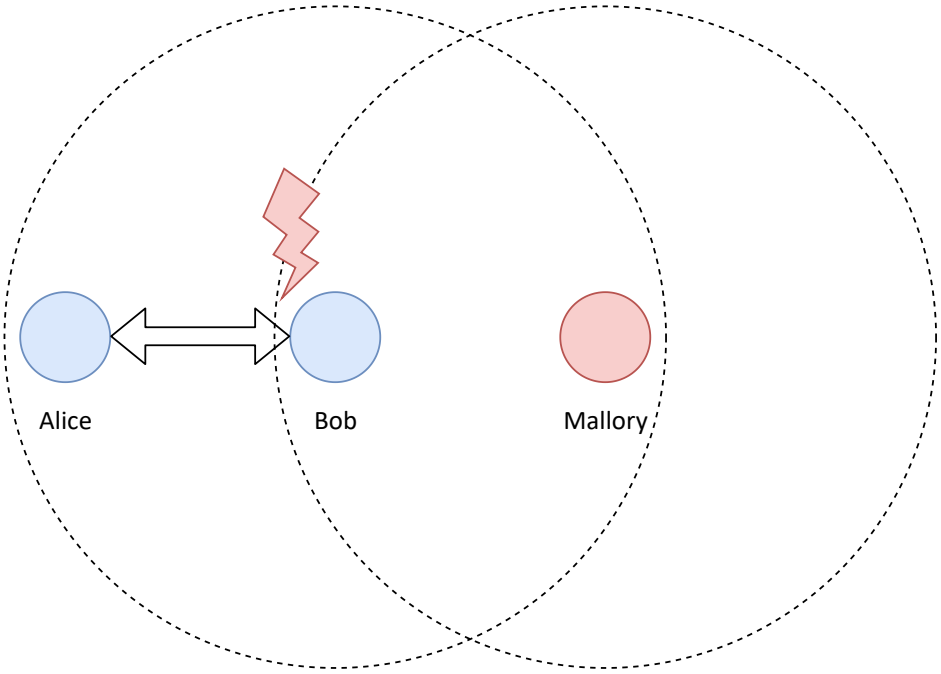

Abbildung 7.13: Mallory als Hidden Station aus Sicht von Alice

Der zweite Testfall (Abbildung 7.14) beschreibt einen Aufbau, in dem sich alle Teilnehmer (Alice, Bob und Mallory) in einer Kollisionsdomäne befinden. Hierbei sendet Bob auf einem Nachbarkanal von Alice und Bob, um maximale Störungen zu verursachen. Mallory sendet dabei mit maximaler Kanalbelegung, um Alice möglichst

wenig Air Time zu überlassen. Nach einer Experimentaldauer von 12 Stunden wurden 43.088 Nachrichten bzw. Frames übertragen. Im Gegensatz zum ersten Testfall wurde lediglich eine Nachricht neu übertragen. Keine der Nachrichten musste öfter als zweimal gesendet werden.

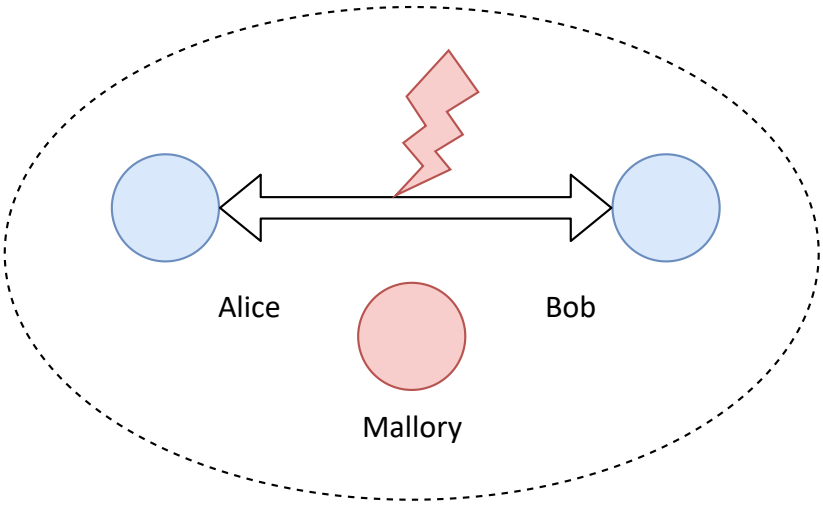

Abbildung 7.14: Kommunikation zwischen Alice und Bob durch Angreifer in Funkreichweite gestört

Der WLAN-Standard IEEE 802.11 sieht bereits wirksame Gegenmaßnahmen vor, um eine faire Kanalzuteilung zu ermöglichen. Insbesondere durch die Wartepausen jeder Station bleibt ein Zeitfenster für andere Stations Frames zu senden. Daher kann geschlussfolgert werden, dass es einem Angreifer nicht möglich ist, effektiv die Kommunikation zwischen SC und EDs zu stören, um die Verteilung von Konfigurationsdaten und Steuerbefehlen zu verhindern.

7.4.2 Seitenkanalangriffe

Werden EDs in mindestens zwei unabhängige Trust Zones gruppiert, kommunizieren die Anwendungen über separate virtuelle MAC-Interfaces mit den jeweiligen EDs der Trust Zone (Abbildung 7.15). Wenn die einzelnen Anwendungsprozesse so programmiert sind, dass sie keine Daten austauschen, existiert keine direkte Verbindung zwischen ihnen. In den vergangenen Jahren wurden verschiedene Hardware-Eigenheiten ausgenutzt, um einen Informationsaustausch zwischen abgeschotteten Prozessen oder sogar Virtuellen Maschinen zu ermöglichen [A 126], [A 127],

[A 128], [A 129], [A 130]. Der jeweilige Seitenkanal kann missbraucht werden, um sensible Daten wie Schlüsselmaterial abzugreifen. Außerdem besteht die Gefahr, dass aktiv Daten über den Seitenkanal manipuliert werden.

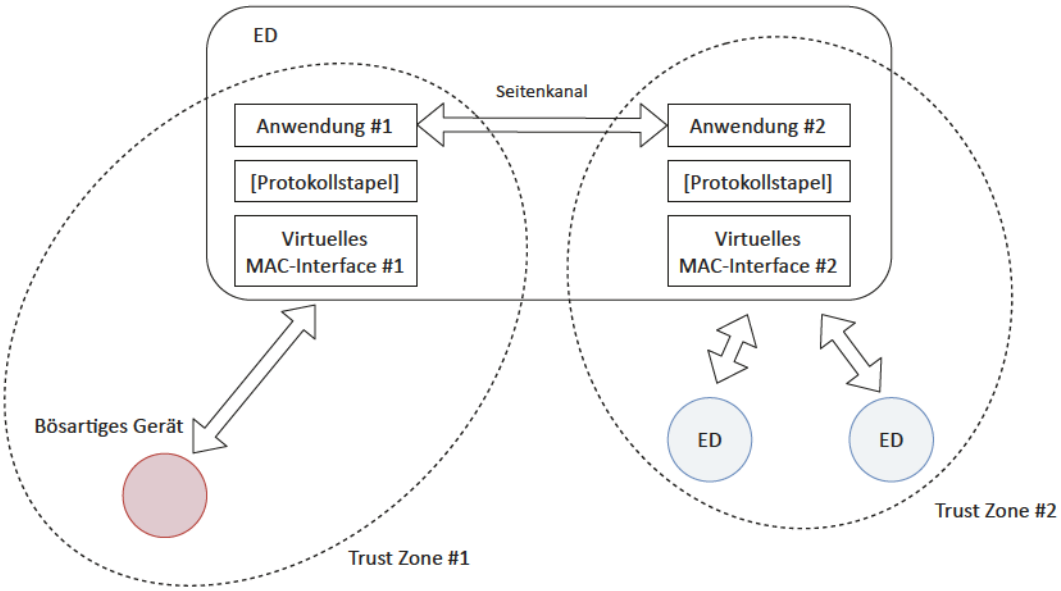

Abbildung 7.15: Seitenkanal eines ED, das sich in zwei Trust Zones befindet

Das Ausnutzen von Seitenkanälen stellt die größte Bedrohung für die vorgestellte Security Architektur dar. Kontrolliert ein Angreifer ein Gerät, das eine entsprechende Hardware-Spezifische Schwachstelle (Seitenkanal) aufweist, ist es möglich, aus einer Trust Zone auszubrechen. Wenn der Angreifer netzwerktechnische Konnektivität zu einer anderen Trust Zone erlangt hat, so ist der potentielle Schaden lediglich auf die weitere Trust Zone beschränkt. Eine Gegenmaßnahme ist, EDs bei Bekanntwerden einer Schwachstelle prophylaktisch in Quarantäne zu überführen. Alle anderen EDs erhalten bei diesem Vorgang neue Zugangsdaten für ihre jeweilige Trust Zone(s). Die technische Umsetzung dieses Verfahrens wurde in Abschnitt 7.3 beschrieben und experimentell evaluiert.

7.5 Zwischenfazit

Mit ANTs wurde die untere Ebene des GA-Systems betrachtet. Nach der Analyse des Standes der Technik zu verschiedenen Netzwerkpartitionierungsverfahren, fiel die Wahl auf eine Separation auf MAC-Ebene. Exemplarisch dient die IEEE 802.11s WLAN-Mesh Erweiterung als technologische Grundlage für experimentelle Evaluationen. Konzeptionell lassen sich alle Netzwerktechnologien verwenden, die eine gesicherte Kommunikation (Vertraulichkeit, Integrität und Authentizität sichergestellt) zwischen Teilnehmern ermöglichen. Dazu benötigt jedes ED Konfigurationsdaten. Die vom SC bestimmten Konfigurationsdaten werden über ein universelles Interface auf Anwendungsebene an jedes ED übertragen. Dazu muss jedes ED lediglich einen minimalen Funktionsumfang implementieren. In einer prototypischen Implementierung mit 25 EDs konnte die Konfiguration und Neukonfiguration zur Laufzeit durch den SC erfolgreich nachgewiesen werden. Je Gerät ist im Versuchsaufbau eine Nachricht mit maximal 1024 Byte Payload ausreichend gewesen, die essentiellen Konfigurationsparameter an ein Gerät zu senden. Die Konfigurationsdatenmenge steigt lediglich linear mit steigender Teilnehmerzahl. Außerdem wurde gezeigt, dass einzelne Geräte mit mehreren Trust Zones verbunden sein können. Dazu kommuniziert das ED über je ein virtuelles MAC-Interface mit einer Trust Zone. Experimente haben gezeigt, dass selbst verschiedene Anwendungen mit hoher Datenrate parallel über mehrere virtuelle MAC-Interfaces ausgeführt werden können. Dieser Extremfall tritt in der GA z.B. bei Überwachungskameras auf, die Audio-/Videodaten an Anzeigegeräte in verschiedenen Trust Zones senden. Des Weiteren wurde die Zuverlässigkeit der Gerätekonfiguration bei einem gestörten WLAN-Kanal getestet. Da nur eine Nachricht (ein Frame auf MAC-Ebene) genügt, konnten alle relevanten Konfigurationsdaten zuverlässig übertragen werden. Trotz verschiedener Störmuster, bei denen ein Angreifer per Remotezugriff Geräte der GA missbraucht, um WLAN-Traffic zu verursachen, waren 99,9 % aller Übertragungen beim ersten Versuch erfolgreich. Trust Zones können auf MAC-Ebene realisiert werden. Die Tauglichkeit für eingebettete Systeme und die Resilienz wurden experimentell nachgewiesen.

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

Protokoll- und Technologieauswahl GA-Systeme der Zukunft basieren auf offenen Internetstandards. Ein selbst zusammengestellter Anforderungskatalog dient zur Bewertung verschiedener Protokolle. Untersucht wurden die klassische GA-Protokolle KNX und BACnet/IP. Außerdem behandelt diese Arbeit verschiedene M2M-Prokokolle und Internetstandards. Durch IP-basierte Protokolle, die für die Anforderungen von eingebetteten Systemen optimiert sind, lassen sich die einzelnen Geräte herstellerunabhängig miteinander vernetzen. Die Anbindung an Cloud-Dienste zur Fernwartung oder zur Bereitstellung von Mehrwertdiensten wird deutlich erleichtert. Des Weiteren wurden die Eigenschaften unterschiedlicher Drahtloskommunikationsprotokolle miteinander verglichen und bewertet. Der WLAN-Standard IEEE 802.11s ermöglicht es vermaschte Netzwerktopologien aufzubauen. Dadurch können ohne zusätzliche Netzwerkinfrastruktur größere Bereiche mit einem Netzwerk abgedeckt werden.

Performance-Validierung für Streaminganwendungen Die theoretischen [B 1] und experimentellen [B 2] Untersuchungen haben gezeigt, dass sich CoAP eignet, um die verschiedenen Anwendungen von der sporadischen Übermittlung von Sensordaten bis hin zur Übertragung von Datenströmen zu realisieren. Dass sich CoAP für Streaminganwendungen eignet, konnte anhand eines Versuchsaufbaus mit verschiedenen eingebetteten Systemen nachgewiesen werden.

Bedrohungsanalyse Durch die steigende Zahl an Geräten im Netzwerk und stetige Änderungen am System, durch Entsorgung und Neuinstallation von Hardware, entstehen große Security-Risiken. Moderne Security-Konzepte müssen von Angreifern ausgehen, die sich im Netzwerk befinden. Sie kontrollieren die Firmware von Geräten, um Schaden durch Abgreifen von sensiblen Informationen und Steuern von Aktorik anzurichten. Zudem besteht die Gefahr, Ransomware auf weitere Netzwerkteilnehmer zu verteilen.

BIM-basierte Planung von Trust Zones Durch eine neuartige Security-Architektur, die Gebäudeinformationen und Anwendungsmodelle berücksichtigt, wird das gesamte GA-Netzwerk strukturiert, um den Einflussbereich und die Angriffsfläche für bösartige Geräte zu reduzieren [B 4], [B 5]. Die Verwaltung des Netzes wird auf

einem Security Controller ausgeführt. Es ist möglich, dieses Gerät auf Basis des Smart-Meter-Gateway-Schutzprofils zu entwickeln und zertifizieren zu lassen. Auf Basis der zertifizierten Hardware als Vertrauensanker werden die einzelnen Geräte mit Konfigurationsdaten versorgt. Damit Geräte mit dem vorgestellten Sicherheitskonzept kompatibel sind, müssen sie einen minimalen Funktionsumfang implementieren. Die Forschungsarbeit umfasst alle Abschnitte des Entwicklungsprozesses. Sie gibt Lösungen zur Geräteentwicklung für Hersteller, um Geräte generisch zu programmieren, damit sie in einer konkreten GA optimal eingebunden werden können. Anerkannte kryptographische Verfahren und Protokolle stellen die Grundlage für den Kommissionierungsprozess dar. In dieser Arbeit wird ein Modell-basierter Partitionierungsalgorithmus vorgestellt, um jedes Gerät in eine Trust Zone einzuteilen. Der Algorithmus wurde prototypisch implementiert und experimentell evaluiert. Es konnte nachgewiesen werden, dass die Laufzeit des Algorithmus linear mit steigender Netzwerkgröße steigt. Die absolute Ausführungszeit auf einer praxisnahen Hardware-/Software-Plattform liegt in einem vertretbaren Bereich von weniger als einer Sekunde. Neben der initialen Konfiguration (Anwendungslogik und Schlüsselmaterial) werden alle Geräte zur Laufzeit neu konfiguriert, um akuten Angriffen entgegenzuwirken oder kompromittiertes Schlüsselmaterial durch entsorgte Geräte auszutauschen.

ANTs In einem Testaufbau wurden Endgeräte vom SC aus mit Konfigurationsdaten für die MAC-Schicht versorgt [B 9]. Es konnte gezeigt werden, dass alle Geräte das Schlüsselmaterial empfangen und auf einem virtuellen MAC-Interface angewendet haben [B 8]. Außerdem konnte nachgewiesen werden, dass die Nutzung mehrerer virtueller MAC-Interfaces keine Performance-Einbußen nach sich zieht. Die entwickelte Architektur wurde prototypisch implementiert, um die Funktionen zu demonstrieren und Angriffe auf den Übertragungskanal des Konfigurationsnetzes zu untersuchen. Trotz Störung des Übertragungskanals durch einen Angreifer ist es nicht gelungen, die Übermittlung von Konfigurationsdaten zu verhindern. Durch die Verwendung einer vermaschten Netzwerkinfrastruktur nach IEEE 802.11s und einer dezentral ausgeführten Anwendungslogik wird eine hohe Dynamik erzielt. Es ist dadurch möglich, dynamisch zur Laufzeit Anwendungen auf andere Geräte abzubilden und Nachrichten über alternative, selbstorganisierte Routen auszutauschen. Außerdem ist es innerhalb der vom Planungsalgorithmus berechneten Trust Zones möglich, auf Security Protokolle wie DTLS und OSCORE zu verzichten. Daraus re-

sultieren für Endgeräte ein verringerter Energiebedarf und eine verringerte Übertragungslatenz. In Abschnitt 6.4 wurden die Verbesserungen quantitativ abgeschätzt.

8.2 Ausblick

Die Arbeit beschreibt ein Sicherheitskonzept, das den gesamten Lebenszyklus von Geräten und der GA betrachtet. Der Digital Twin modelliert das Gesamtsystem, sodass in Kombination mit allgemeingültigen Sicherheitsrichtlinien die Security durch individuelle Konfigurationsdaten verbessert wird. Dabei fließen bislang u.a. Standortdaten von nichtbeweglichen Geräten, die über ein vermaschtes Netzwerk verbunden sind, ein. Die erarbeiteten Konzepte wurden prototypisch mit Web-Technologien implementiert um ihre Tauglichkeit für eingebettete Systeme zu evaluieren und zu demonstrieren. Web-Technologien werden auch in Zukunft weiter Einzug in eingebettete Systeme halten.

Daher ergeben sich für zukünftige Arbeiten weitere Anwendungsfelder, die durch mobile Geräte charakterisiert sind. Nachdem jedes Gerät in eine abgeschottete Trust Zone platziert wurde, übernehmen weitere Geräte innerhalb derselben Mesh-basierten Trust Zone Aufgaben als Netzwerkinfrastrukturknoten zur Sicherung der Ausfallsicherheit durch Redundanz. Weitere Optimierungsgrößen, wie zum Beispiel die Latenz beim Ansprechen von Geräten sowie der Energieumsatz von batteriebetriebenen Geräten, können in den Planungsalgorithmus einfließen. Dabei sollten auch andere Drahtloskommunikationstechnologien und Anwendungsschichtprotokolle untersucht werden. Der Digital Twin lässt sich durch Netz- und Energiemodelle erweitern und an die Herausforderungen moderner GA-Systeme anpassen. Außerdem ist eine Verknüpfung mit reaktiven Security-Maßnahmen sinnvoll. Dabei wird das Netzwerk im Angriffsfall neu konfiguriert (z.B. betroffene Endgeräte in Quarantäne setzen), um den Schaden zu begrenzen. Der Security Controller kann zur Laufzeit Statusinformationen der einzelnen Trust Zones erfassen, protokollieren und auswerten um beispielweise Angriffe zu detektieren. Annomalitätsmuster können vom Hersteller an den Security Controller über eine sichere Internetverbindung verteilt werden. Die Erweiterung des Security Controller steht im Einklang zur erarbeiteten Security Architektur. Der vorgestellte Security Controller kann nach dieser Architektur durch weitere Funktionalitäten (Anwendungen) erweitert werden. Durch die dynamische Erweiterung und Anpassung von Security Funktionalitäten ist das Gebäudeautomationssystem auf zukünftige Anforderungen vorbereitet.

A Literaturverzeichnis

[1] „Webseite: KNX Association," KNX Association, Techn. Ber., 2021. Adresse: https://www.knx.org/knx-en/for-professionals/index.php.

[2] S. Liaisons, R. Hall, M. Modera u. a., „BACnet-A Data Communication Protocol for Building Automation and Control Networks," *ANSI/ASHRAE Standard*, Jg. 135, 2012.

[3] J. Luo, B. Qian, Y. Su und W. Yu, „Research on Building Lighting Intelligent Control System Based on Bus Technology," in *2021 9th International Conference on Orange Technology (ICOT)*, 2021, S. 1–4. DOI: 10.1109/ICOT54518.2021.9680662.

[4] J. Pan, R. Jain und S. Paul, „A Survey of Energy Efficiency in Buildings and Microgrids using Networking Technologies," *IEEE Communications Surveys & Tutorials*, Jg. 16, Nr. 3, S. 1709–1731, 2014. DOI: 10.1109/SURV.2014.060914.00089.

[5] N. Mishra und S. Pandya, „Internet of Things Applications, Security Challenges, Attacks, Intrusion Detection, and Future Visions: A Systematic Review," *IEEE Access*, Jg. 9, S. 59 353–59 377, 2021. DOI: 10.1109/ACCESS.2021.3073408.

[6] W. Iqbal, H. Abbas, M. Daneshmand, B. Rauf und Y. A. Bangash, „An In-Depth Analysis of IoT Security Requirements, Challenges, and Their Countermeasures via Software-Defined Security," *IEEE Internet of Things Journal*, Jg. 7, Nr. 10, S. 10 250–10 276, 2020. DOI: 10.1109/JIOT.2020.2997651.

[7] R. AL MOGBIL, M. AL ASQAH und S. EL KHEDIRI, „IoT: Security Challenges and Issues of Smart Homes/Cities," in *2020 International Conference on Computing and Information Technology (ICCIT-1441)*, 2020, S. 1–6. DOI: 10.1109/ICCIT-144147971.2020.9213827.

[8] S. Sinche, D. Raposo, N. Armando u. a., „A Survey of IoT Management Protocols and Frameworks," *IEEE Communications Surveys & Tutorials*, Jg. 22, Nr. 2, S. 1168–1190, 2020. DOI: 10.1109/COMST.2019.2943087.

[9] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari und M. Ayyash, „Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, Jg. 17, Nr. 4, S. 2347–2376, 2015. DOI: 10.1109/COMST.2015.2444095.

[10] V. Altmann, B. Butzin, R. Balla, F. Golatowski und D. Timmermann, „A BACnet gateway for embedded Web services," in *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, 2015, S. 1–6. DOI: 10.1109/ETFA.2015.7301563.

[11] R. Droms, *Dynamic Host Configuration Protocol*, RFC 2131, März 1997. DOI: 10.17487/RFC2131. Adresse: https://www.rfc-editor.org/info/rfc2131.

[12] A. Nugur, M. Pipattanasomporn, M. Kuzlu und S. Rahman, „Design and Development of an IoT Gateway for Smart Building Applications," *IEEE Internet of Things Journal*, Jg. 6, Nr. 5, S. 9020–9029, 2019. DOI: 10.1109/JIOT.2019.2926099.

[13] W. Jin, R. Xu, T. You, Y.-G. Hong und D. Kim, „Secure Edge Computing Management Based on Independent Microservices Providers for Gateway-Centric IoT Networks," *IEEE Access*, Jg. 8, S. 187 975–187 990, 2020. DOI: 10.1109/ACCESS.2020.3030297.

[14] B. Kang, D. Kim und H. Choo, „Internet of Everything: A Large-Scale Autonomic IoT Gateway," *IEEE Transactions on Multi-Scale Computing Systems*, Jg. 3, Nr. 3, S. 206–214, 2017. DOI: 10.1109/TMSCS.2017.2705683.

[15] M. Aminul Hoque, M. Hossain und R. Hasan, „IGaaS: An IoT Gateway-as-a-Service for On-demand Provisioning of IoT Gateways," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, 2020, S. 1–6. DOI: 10.1109/WF-IoT48130.2020.9221225.

[16] B. Schneier und P. Sutherland, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd. USA: John Wiley und Sons, Inc., 1995, ISBN: 0471128457.

[17] C. (.-). Eckert, *IT-Sicherheit: Konzepte - Verfahren - Protokolle*, 8., aktualisierte und korr. Aufl. München: Oldenbourg, 2013, Verfasserangabe: von Claudia Eckert ; Quelldatenbank: FHBK-x ; Format:marcform: print ; Umfang: XIV, 1016 S : Ill., graph. Darst. ; 978-3-486-72138-6 Kunst. : EUR 69.80 (DE), ISBN: 978-3-486-72138-6. Adresse: http://digitool.hbz-nrw.de:1801/webclient/DeliveryManager?pid=5082642%5C&custom%5C_att%5C_2=simple%5C_viewer;%20http://digitool.hbz-nrw.de:1801/webclient/DeliveryManager?pid=5082643%5C&custom%5C_att%5C_2=simple%5C_viewer.

[18] E. Rescorla, *Diffie-Hellman Key Agreement Method*, RFC 2631, Juni 1999. DOI: 10.17487/RFC2631. Adresse: https://rfc-editor.org/rfc/rfc2631.txt.

[19] National Institute of Standards and Technology (NIST), *Federal Information Processing Standards Publication 197: Announcing the ADVANCED ENCRYPTION STANDARD (AES)*, 2001. Adresse: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[20] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, RFC 8446, Aug. 2018. DOI: 10.17487/RFC8446. Adresse: https://rfc-editor.org/rfc/rfc8446.txt.

[21] D. H. Krawczyk, M. Bellare und R. Canetti, *HMAC: Keyed-Hashing for Message Authentication*, RFC 2104, Feb. 1997. DOI: 10.17487/RFC2104. Adresse: https://rfc-editor.org/rfc/rfc2104.txt.

[22] J. Jonsson, B. Kaliski und RSA Laboratories, „RFC 3447: Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1," Internet Engineering Task Force, Techn. Ber., 2003. Adresse: https://tools.ietf.org/html/rfc3447.

[23] K. Igoe, D. McGrew und M. Salter, *Fundamental Elliptic Curve Cryptography Algorithms*, RFC 6090, Feb. 2011. DOI: 10.17487/RFC6090. Adresse: https://rfc-editor.org/rfc/rfc6090.txt.

[24] F. G. Björn Butzin, „Abbau von Nutzbarkeitshürden für den Einsatz von effektiven Sicherheitsmechanismen in der Gebäudeautomation der Zukunft," Universität Rostock, Forschungsinitiative Zukunft Bau, Bundesinstitut für Bau, Stadt und Raumfoschung, Techn. Ber., 2019. Adresse: https://www.irbnet.de/daten/rswb/20019000398.pdf.

[25] A. Borrmann, J. Beetz, C. Koch, T. Liebich und S. Muhic, „Industry Foundation Classes: A Standardized Data Model for the Vendor-Neutral Exchange of Digital Building Models," in Sep. 2018, S. 81–126, ISBN: 978-3-319-92861-6. DOI: 10.1007/978-3-319-92862-3_5.

[26] *Webseite: buildingSMART*. Adresse: https://www.buildingsmart.de/.

[27] „Systeme der Gebäudeautomation - Teil 5: Datenkommunikationsprotokoll," DIN EN ISO 16484-5, Techn. Ber., 2016. Adresse: https://www.knx.org/knx-en/for-professionals/index.php.

[28] Z. M.-l. Zhou Ning und X. Yi-ping, „BACnet® for Video Surveillancey," *ASHRAE Journal*, Jg. 46, S. 18–23, 2004.

[29] M. O. (1. I. W. G. David Fisher Bernhard Isler, *Whitepaper: BACnet Secure Connect - A Secure Infrastructure for Building Automation*. Adresse: http://www.bacnet.org/Bibliography/B-SC-Whitepaper-v15_Final_20190521.pdf.

[30] A. Banks und R. Gupta, *MQTT Version 3.1.1*, OASIS Standard, Okt. 2014. Adresse: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.

[31] S. Hartman, K. Raeburn, T. Yu, C. Neumann und Network Working Group, *RFC 4120: The Kerberos Network Authentication Service (V5)*, 2005. Adresse: https://tools.ietf.org/html/rfc4120.

[32] J. Sermersheim, *Lightweight Directory Access Protocol (LDAP): The Protocol*, RFC 4511, Juni 2006. DOI: 10.17487/RFC4511. Adresse: https://rfc-editor.org/rfc/rfc4511.txt.

[33] C. Rigney, S. Willens, A. Rubens, Merit, W. Simpson und Daydreamer, „RFC 2138: Remote Authentication Dial In User Service (RADIUS)," Internet Engineering Task Force, Techn. Ber., 2000. Adresse: https://tools.ietf.org/pdf/rfc2865.pdf.

[34] R. Fielding, U. C. Irvine und J. Gettys, *RFC 2616: Hypertext Transfer Protocol – HTTP/1.1*, 1999. Adresse: http://tools.ietf.org/pdf/rfc2616.pdf.

[35] D. Winer, *RSS 2.0 Specification*, Juli 2003.

[36] A. Melnikov und I. Fette, *The WebSocket Protocol*, RFC 6455, Dez. 2011. DOI: 10.17487/RFC6455. Adresse: https://www.rfc-editor.org/info/rfc6455.

[37] T. Dierks und E. Rescorla, *RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2*, 2008. Adresse: http://tools.ietf.org/pdf/rfc5246.pdf.

[38] Z. Shelby, K. Hartke und C. Bormann, *The Constrained Application Protocol (CoAP)*, RFC 7252, Juni 2014. DOI: 10.17487/RFC7252. Adresse: https://rfc-editor.org/rfc/rfc7252.

[39] G. Selander, J. Mattsson, F. Palombini und L. Seitz, „Object Security for Constrained RESTful Environments (OSCORE)," Internet Engineering Task Force, Internet-Draft draft-ietf-core-object-security-14, Juli 2018, Work in

Progress, 81 S. Adresse: https://datatracker.ietf.org/doc/html/draft-ietf-core-object-security-14.

[40] C. Bormann und P. E. Hoffman, *Concise Binary Object Representation (CBOR)*, RFC 7049, Okt. 2013. DOI: 10.17487/RFC7049. Adresse: https://rfc-editor.org/rfc/rfc7049.txt.

[41] J. Schaad, *CBOR Object Signing and Encryption (COSE)*, RFC 8152, Juli 2017. DOI: 10.17487/RFC8152. Adresse: https://rfc-editor.org/rfc/rfc8152.txt.

[42] R. Barnes, *Use Cases and Requirements for JSON Object Signing and Encryption (JOSE)*, RFC 7165, Apr. 2014. DOI: 10.17487/RFC7165. Adresse: https://rfc-editor.org/rfc/rfc7165.txt.

[43] G. Selander, J. Mattsson, F. Palombini und L. Seitz, *Object Security for Constrained RESTful Environments (OSCORE)*, RFC 8613, Juli 2019. DOI: 10.17487/RFC8613. Adresse: https://rfc-editor.org/rfc/rfc8613.txt.

[44] G. Choi, D. Kim und I. Yeom, „Efficient streaming over CoAP," in *IEEE 2016 International Conference on Information Networking (ICOIN)*, IEEE, 2016).

[45] P. Krawiec, M. Sosnowski, J. Mongay Batalla, C. X. Mavromoustakis und G. Mastorakis, „DASCo: dynamic adaptive streaming over CoAP," *Multimedia Tools and Applications*, Jg. 77, Nr. 4, S. 4641–4660, Feb. 2018, ISSN: 1573-7721. DOI: 10.1007/s11042-017-4854-z. Adresse: https://doi.org/10.1007/s11042-017-4854-z.

[46] *Webseite: Open Mobile Alliance (OMA) - OMA Specifications*. Adresse: http://openmobilealliance.org/wp/index.html.

[47] IEEE, „IEEE Standard for Local and metropolitan area networks, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2012*, DOI: 10.1109/IEEESTD.2012.6178212.

[48] G. R. Hiertz, D. Denteneer, S. Max u. a., „IEEE 802.11s: The WLAN Mesh Standard," *IEEE Wireless Communications*, Jg. 17, Nr. 1, S. 104–111, 2010. DOI: 10.1109/MWC.2010.5416357.

[49] IEEE, *IEEE Standard for Low-Rate Wireless Networks*. Adresse: https://standards.ieee.org/ieee/802.15.4/7029/.

[50] G. Montenegro, J. Hui, D. Culler und N. Kushalnagar, *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*, RFC 4944, Sep. 2007. DOI: 10.17487/RFC4944. Adresse: https://www.rfc-editor.org/info/rfc4944.

[51] zigbee alliance, *Zigbee Specification Rev. 22 1.0*. Adresse: https://csa-iot.org/wp-content/uploads/2022/01/docs-05-3474-22-0csg-zigbee-specification-1.pdf.

[52] Silicon Labs, *Z-Wave Specification*. Adresse: https://www.silabs.com/wireless/z-wave/specification.

[53] M. Woolley, *Standard: Bluetooth® Core Specification Version 5.0 Feature Enhancements*. Adresse: https://www.bluetooth.com/wp-content/uploads/2019/03/Bluetooth_5-FINAL.pdf.

[54] U. F. Khan, S. Hameed und T. Macintyre, „TCP/IP Over Bluetooth," in *Advances in Computer and Information Sciences and Engineering*, T. Sobh, Hrsg., Dordrecht: Springer Netherlands, 2008, S. 479–484, ISBN: 978-1-4020-8741-7.

[55] L. Alliance™, *LoRaWAN™ 1.1 Specification*. Adresse: https://lora-alliance.org/wp-content/uploads/2020/11/lorawantm_specification_-v1.1.pdf.

[56] G. Association, *LTE-M Deployment Guide to Basic Feature Set Requirements*. Adresse: https://www.gsma.com/iot/wp-content/uploads/2018/04/LTE-M_Deployment_Guide_v2_5Apr2018.pdf.

[57] 3GPP, *Study on Narrow-Band Internet of Things (NB-IoT) / enhanced Machine Type Communication (eMTC)*. Adresse: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3747.

[58] C. G. C. Carducci, A. Monti, M. H. Schraven, M. Schumacher und D. Mueller, „Enabling esp32-based iot applications in building automation systems," in *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4. 0&IoT)*, IEEE, 2019, S. 306–311.

[59] „IEEE Standard for Ethernet," *IEEE Std 802.3-2018 (Revision of IEEE Std 802.3-2015)*, S. 1–5600, 2018. DOI: 10.1109/IEEESTD.2018.8457469.

[60] C. Bormann und Z. Shelby, *Block-Wise Transfers in the Constrained Application Protocol (CoAP)*, RFC 7959, Aug. 2016. DOI: 10.17487/RFC7959. Adresse: https://rfc-editor.org/rfc/rfc7959.txt.

[61] I. Sodagar, „The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MultiMedia*, Jg. 18, Nr. 4, S. 62–67, Apr. 2011, ISSN: 1070-986X. DOI: 10.1109/MMUL.2011.71.

[62] ITU-T, „Standard: H.264 Advanced video coding for generic audiovisual services,“ Techn. Ber., 2016.

[63] ITU-T, „Standard: H.265 High efficiency video coding,“ Techn. Ber., 2016.

[64] O. Diaz, S. Loreto und H. Back, *Method and server for sending a data stream to a client and method and client for receiving a data stream from a server*, US Patent 9,621,617, 2017. Adresse: https://www.google.com/patents/US9621617.

[65] Raspberry Pi Foundation, *Webpage: Raspberry Pi Documentation*. Adresse: http://www.raspberrypi.org/documentation/hardware/README.md.

[66] *Web Page: Californium Repository*. Adresse: https://github.com/eclipse/californium.

[67] W. Group, *Git Repository: jCoAP*, 2016. Adresse: https://gitlab.amd.e-technik.uni-rostock.de/ws4d/jcoap.

[68] M. Kasparick, B. Beichler, B. Konieczek u. a., „Measuring latencies of IEEE 11073 compliant service-oriented medical device stacks,“ in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, S. 8640–8647. DOI: 10.1109/IECON.2017.8217518.

[69] *Web Page: Libcoap Repository*. Adresse: https://github.com/obgm/libcoap.

[70] *ZedBoard documentations*, 2013. Adresse: http://www.zedboard.org/documentation.

[71] S. Tomar, „Converting video formats with FFmpeg,“ *Linux Journal*, Jg. 2006, Nr. 146, S. 10, 2006.

[72] International Organization for Standardization, *ISO/IEC 10918-1:1994: Information technology — Digital compression and coding of continuous-tone still images: Requirements and guidelines*. Geneva, Switzerland: International Organization for Standardization, 1994, S. 182. Adresse: http://www.iso.ch/cate/d18902.html.

[73] J. S. (
bibinitperiod T. H. (Oracle), „VisualVM - All-in-One Java Troubleshooting Tool,“ Techn. Ber., 2021. Adresse: https://visualvm.github.io/.

[74] S. Unger, *Secure Web Services für ambiente eingebettete Systeme*, German. Rostock: Universität, 2015, Dissertation Universität Rostock 2016.

Adresse: http://rosdok.uni-rostock.de/resolve/urn/urn:nbn:de:gbv:28-diss2016-0024-0.

[75] D. Dolev und A. Yao, „On the security of public key protocols," *IEEE Transactions on Information Theory*, Jg. 29, Nr. 2, S. 198–208, 1983. DOI: 10.1109/TIT.1983.1056650.

[76] Z. Shelby und C. Bormann, *6LoWPAN: The wireless embedded Internet*. John Wiley & Sons, 2011, Bd. 43.

[77] T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. medi und K. Wehrle, „Security Challenges in the IP-Based Internet of Things," *Wireless Personal Communications*, Jg. 61, Dez. 2011. DOI: 10.1007/s11277-011-0385-5.

[78] R. Roman, J. Zhou und J. Lopez, „On the features and challenges of security and privacy in distributed internet of things," *Computer Networks*, Jg. 57, Nr. 10, S. 2266–2279, 2013, Towards a Science of Cyber Security Security and Identity Architecture for the Future Internet, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2012.12.018. Adresse: https://www.sciencedirect.com/science/article/pii/S1389128613000054.

[79] OWASP, „OWASP Top 10 - 2013, The Ten Most Critical Web Application Security Risks," Techn. Ber., 2013, S. 1–36. Adresse: https://www.owasp.org/images/f/f8/OWASP_Top_10_-_2013.pdf.

[80] G. Zhang, X. Qiu und Y. Gao, „Software Defined Security Architecture with Deep Learning-Based Network Anomaly Detection Module," in *2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN)*, 2019, S. 784–788. DOI: 10.1109/ICCSN.2019.8905304.

[81] X. Qiu, F. Cheng, W. Wang, G. Zhang und Y. Qiu, „A security controller-based software defined security architecture," in *Proc. of ICIN '17*.

[82] A. Darabseh, M. Al-Ayyoub, Y. Jararweh, E. Benkhelifa, M. Vouk und A. Rindos, „SDSecurity: A Software Defined Security experimental framework," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2015, S. 1871–1876. DOI: 10.1109/ICCW.2015.7247453.

[83] G. Steindl und W. Kastner, „Semantic Microservice Framework for Digital Twins," *Applied Sciences*, Jg. 11, Nr. 12, 2021, ISSN: 2076-3417. DOI: 10.3390/app11125633. Adresse: https://www.mdpi.com/2076-3417/11/12/5633.

[84] A. Fernbach und W. Kastner, „Gebäudemanagement durch Wissensbasierte Systeme,“ in Jan. 2018, S. 97–106, ISBN: 978-3-662-55231-5. DOI: 10.1007/978-3-662-55232-2_8.

[85] M. Jung, „Security by Delegation für Industrie 4.0,“ in Jan. 2018, S. 73–84, ISBN: 978-3-662-55231-5. DOI: 10.1007/978-3-662-55232-2_6.

[86] S. Gerdes, O. Bergmann und C. Bormann, „Delegated CoAP Authentication and Authorization Framework (DCAF),“ 2015.

[87] T. Hardjono und N. Smith, „Fluffy: Simplified Key Exchange for Constrained Environments,“ Internet Engineering Task Force, Internet-Draft draft-hardjono-ace-fluffy-03, Juli 2016, Work in Progress, 46 S. Adresse: https://datatracker.ietf.org/doc/html/draft-hardjono-ace-fluffy-03.

[88] N. Kang, J. Park, H. Kwon und S. Jung, „ESSE: Efficient Secure Session Establishment for Internet-Integrated Wireless Sensor Networks,“ *International Journal of Distributed Sensor Networks*, Jg. 11, Nr. 8, S. 393 754, 2015. DOI: 10.1155/2015/393754. eprint: https://doi.org/10.1155/2015/393754. Adresse: https://doi.org/10.1155/2015/393754.

[89] R. Hummen, H. Shafagh, S. Raza, T. Voig und K. Wehrle, „Delegation-based authentication and authorization for the IP-based Internet of Things,“ in *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2014, S. 284–292. DOI: 10.1109/SAHCN.2014.6990364.

[90] S. R. Moosavi, T. N. Gia, E. Nigussie u. a., „Session Resumption-Based End-to-End Security for Healthcare Internet-of-Things,“ in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, S. 581–588. DOI: 10.1109/CIT/IUCC/DASC/PICOM.2015.83.

[91] G. Lessa dos Santos, V. T. Guimarães, G. da Cunha Rodrigues, L. Z. Granville und L. M. R. Tarouco, „A DTLS-based security architecture for the Internet of Things,“ in *2015 IEEE Symposium on Computers and Communication (ISCC)*, 2015, S. 809–815. DOI: 10.1109/ISCC.2015.7405613.

[92] J. Granjal, E. Monteiro und J. S. Silva, „End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC

public-key authentication," in *2013 IFIP Networking Conference*, 2013, S. 1–9.

[93] H. Yu, J. He, T. Zhang, P. Xiao und Y. Zhang, „Enabling End-to-End Secure Communication between Wireless Sensor Networks and the Internet," *World Wide Web*, Jg. 16, Juli 2013. DOI: 10.1007/s11280-012-0194-0.

[94] S. Unger und D. Timmermann, „Bridging the UI gap for authentication in smart environments," in *Proc. of ISCC '14*.

[95] J. M. Ho, „A versatile suite of strong authenticated key agreement protocols for body area networks," in *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2012, S. 683–688. DOI: 10.1109/IWCMC.2012.6314287.

[96] „IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security," *IEEE Std 802.1AE-2006*, S. 1–150, 2006.

[97] D. Harkins, „Simultaneous Authentication of Equals: A Secure, Password-Based Key Exchange for Mesh Networks," in *Proc. of SENSORCOMM '08*.

[98] J. Case, M. Fedor, M. Schoffstall und J. Davin, „RFC 1157: A Simple Network Management Protocol (SNMP)," Techn. Ber., 1990, S. 1–36. Adresse: http://tools.ietf.org/pdf/rfc1157.pdf.

[99] S. M. S. Bari, F. Anwar und M. H. Masud, „Performance study of hybrid Wireless Mesh Protocol (HWMP) for IEEE 802.11s WLAN mesh networks," in *2012 International Conference on Computer and Communication Engineering (ICCCE)*, 2012, S. 712–716.

[100] C. Perkins, E. Belding-Royer und S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*, RFC 3561, 2003. Adresse: http://www.ietf.org/rfc/rfc3561.txt.

[101] R. Garroppo, „A joint experimental and simulation study of the IEEE 802.11 s HWMP protocol and airtime link metric," *International Journal of ...*, Nr. March 2011, S. 92–110, 2012. DOI: 10.1002/dac. Adresse: http://onlinelibrary.wiley.com/doi/10.1002/dac.1255/full.

[102] Z. Laaroussi und O. Novo, „A Performance Analysis of the Security Communication in CoAP and MQTT," in *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, 2021, S. 1–6. DOI: 10.1109/CCNC49032.2021.9369565.

[103] Y. Guamán, G. Ninahualpa, G. Salazar und T. Guarda, „Comparative Performance Analysis between MQTT and CoAP Protocols for IoT with Raspberry PI 3 in IEEE 802.11 Environments," in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, 2020, S. 1–6. DOI: 10.23919/CISTI49556.2020.9140905.

[104] S. S. Prayogo, Y. Mukhlis und B. K. Yakti, „The Use and Performance of MQTT and CoAP as Internet of Things Application Protocol using NodeMCU ESP8266," in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, 2019, S. 1–5. DOI: 10.1109/ICIC47613.2019.8985850.

[105] T. L. A. Lagerqvist, „Dissertation: IoT Latency and Power consumption: Measuring the performance impact of MQTT and CoAP," 2018. Adresse: http://urn.kb.se/resolve?urn=urn:nbn:se:hj:diva-39392.

[106] M. Gunnarsson, J. Brorsson, F. Palombini, L. Seitz und M. Tiloca, „Evaluating the performance of the OSCORE security protocol in constrained IoT environments," *Internet of Things*, Jg. 13, S. 100 333, 2021, ISSN: 2542-6605. DOI: https://doi.org/10.1016/j.iot.2020.100333. Adresse: https://www.sciencedirect.com/science/article/pii/S2542660520301645.

[107] „Webseite: Firefly Board," Zolertia, Techn. Ber., 2021. Adresse: https://zolertia.io/product/firefly/.

[108] P. N. Bideh, J. Sönnerup und M. Hell, „Energy Consumption for Securing Lightweight IoT Protocols," in *Proceedings of the 10th International Conference on the Internet of Things*, Ser. IoT '20, Malmö, Sweden: Association for Computing Machinery, 2020, ISBN: 9781450387583. DOI: 10.1145/3410992.3411008. Adresse: https://doi.org/10.1145/3410992.3411008.

[109] I. Ud Din, M. Guizani, B.-S. Kim, S. Hassan und M. Khurram Khan, „Trust Management Techniques for the Internet of Things: A Survey," *IEEE Access*, Jg. 7, S. 29 763–29 787, 2019. DOI: 10.1109/ACCESS.2018.2880838.

[110] D. Cooper, S. Farrell, S. Boeyen, R. Housley und W. Polk, *RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, 2008. Adresse: http://tools.ietf.org/pdf/rfc5280.pdf.

[111] N. S. Nizamkari, „A graph-based trust-enhanced recommender system for service selection in IOT," in *2017 International Conference on Inventive Systems and Control (ICISC)*, 2017, S. 1–5. DOI: 10.1109/ICISC.2017.8068714.

[112] R. Talreja, S. Sathish, K. Nenwani und K. Saxena, „Trust and behavior based system to prevent collision in IoT enabled VANET," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 2016, S. 1588–1591. DOI: 10.1109/SCOPES.2016.7955707.

[113] U. S. Premarathne, „MAG-SIoT: A multiplicative attributes graph model based trust computation method for social Internet of Things," in *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, 2017, S. 1–6. DOI: 10.1109/ICIINFS.2017.8300344.

[114] H. Son, N. Kang, B. Gwak und D. Lee, „An adaptive IoT trust estimation scheme combining interaction history and stereotypical reputation," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2017, S. 349–352. DOI: 10.1109/CCNC.2017.7983132.

[115] C. V. L. Mendoza und J. H. Kleinschmidt, „Defense for selective attacks in the IoT with a distributed trust management scheme," in *2016 IEEE International Symposium on Consumer Electronics (ISCE)*, 2016, S. 53–54. DOI: 10.1109/ISCE.2016.7797367.

[116] J. Caminha, A. Perkusich und M. Perkusich, „A smart middleware to perform semantic discovery and trust evaluation for the Internet of Things," in *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2018, S. 1–2. DOI: 10.1109/CCNC.2018.8319287.

[117] M. Dorodchi, M. Abedi und B. Cukic, „Trust-Based Development Framework for Distributed Systems and IoT," Juni 2016, S. 437–442. DOI: 10.1109/COMPSAC.2016.213.

[118] S. Asiri und A. Miri, „An IoT trust and reputation model based on recommender systems," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, 2016, S. 561–568. DOI: 10.1109/PST.2016.7907017.

[119] H. Hellaoui, A. Bouabdallah und M. Koudil, „TAS-IoT: Trust-Based Adaptive Security in the IoT," in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, 2016, S. 599–602. DOI: 10.1109/LCN.2016.101.

[120] M. Rethfeldt, H. Raddatz, B. Beichler u. a., „ViPMesh: A virtual prototyping framework for IEEE 802.11s wireless mesh networks," in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016, S. 1–7.

[121] *Linux Driver Backports*, 2017. Adresse: https://backports.wiki.kernel.org.

[122] *Linux Wireless Team*, 2017. Adresse: https://wireless.wiki.kernel.org.

[123] M. Rethfeldt, B. Beichler, H. Raddatz u. a., „Mini-Mesh: Practical Assessment of a Miniaturized IEEE 802.11n/s Mesh Testbed," in *IEEE 16th Wireless Communications and Networking Conference*, IEEE, 2018.

[124] cozybit, *Git Repository: cozybit/authsae*, 2017. Adresse: https://github.com/cozybit/authsae.

[125] K. Hartke, *Observing Resources in the Constrained Application Protocol (CoAP)*, RFC 7641, Sep. 2015. DOI: 10.17487/RFC7641. Adresse: https://rfc-editor.org/rfc/rfc7641.txt.

[126] R. Spreitzer, V. Moonsamy, T. Korak und S. Mangard, „Systematic Classification of Side-Channel Attacks: A Case Study for Mobile Devices," *IEEE Communications Surveys Tutorials*, Jg. 20, Nr. 1, S. 465–488, 2018. DOI: 10.1109/COMST.2017.2779824.

[127] X. Lou, T. Zhang, J. Jiang und Y. Zhang, „A Survey of Microarchitectural Side-channel Vulnerabilities, Attacks and Defenses in Cryptography," *CoRR*, Jg. abs/2103.14244, 2021. arXiv: 2103.14244. Adresse: https://arxiv.org/abs/2103.14244.

[128] K. S. Yim, „The Rowhammer Attack Injection Methodology," in *2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS)*, 2016, S. 1–10. DOI: 10.1109/SRDS.2016.012.

[129] R. Qiao und M. Seaborn, „A new approach for rowhammer attacks," in *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2016, S. 161–166. DOI: 10.1109/HST.2016.7495576.

[130] D. Gruss, M. Lipp, M. Schwarz u. a., „Another Flip in the Wall of Rowhammer Defenses," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, S. 245–261. DOI: 10.1109/SP.2018.00031.

B Liste der Veröffentlichungen und Fachvorträge auf Tagungen

[1] H. Raddatz, A. Wall und D. Timmermann, „SafeBase: A Security Framework for Smart Home Systems Based on Smart Metering Infrastructure," in *Proceedings of the 2018 International Conference on Embedded Wireless Systems and Networks*, Ser. EWSN '18, Madrid, Spain: Junction Publishing, 2018, S. 276–277, ISBN: 9780994988621.

[2] A. Wall, H. Raddatz, B. V. Reddy Gopu und D. Timmermann, „Evaluation of CoAP Implementations for Live Streaming using CoAP-Observe," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, S. 468–473. DOI: 10.1109/wf-iot.2019.8767189.

[3] A. Wall, V. Altmann, J. Müller, H. Raddatz und D. Timmermann, „Decentralized configuration of embedded web services for smart home applications," in *2017 Annual IEEE International Systems Conference (SysCon)*, 2017, S. 1–6. DOI: 10.1109/SYSCON.2017.7934759.

[4] A. Wall, B. Butzin, F. Golatowski, M. Rethfeldt und D. Timmermann, „Software-Defined Security Architecture for Smart Buildings using the Building Information Model," in *2019 IEEE Global Conference on Internet of Things (GCIoT)*, 2019, S. 1–5. DOI: 10.1109/GCIoT47977.2019.9058404.

[5] A. Wall, B. Butzin und D. Timmermann, „Trust Zone Formation for Building Automation Networks Using Building Information Modeling," in *2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*, 2020, S. 1–7. DOI: 10.1109/GCAIoT51063.2020.9345857.

[6] H. Raddatz, M. Rethfeldt, M. Kasparick, A. Wall und D. Timmermann, „Absicherung der Gerätekommunikation im Smart Home unter Verwendung des Schutzprofils für Smart Meter Gateways," in *Abschlussbericht*, 2018, S. 84, ISBN: 978-3-7388-0229-0.

[7] M. Rethfeldt, A. Wall, P. Danielis, B. Konieczek und D. Timmermann, „AKadeMesh: Software-defined overlay adaptation for the management of IEEE 802.11s networks," in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2016, S. 477–482. DOI: 10.1109/CCNC.2016.7444826.

[8] A. Wall, H. Raddatz, M. Rethfeldt, P. Danielis und D. Timmermann, „Performance evaluation of MAC-layer trust zones over virtual network interfaces,“ in *2018 Fourth International Conference on Mobile and Secure Services (MobiSecServ)*, 2018, S. 1–5. DOI: 10.1109/MOBISECSERV.2018.8311442.

[9] A. Wall, H. Raddatz, M. Rethfeldt, P. Danielis und D. Timmermann, „ANTs: Application-driven network trust zones on MAC layer in smart buildings,“ in *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2018, S. 1–2. DOI: 10.1109/CCNC.2018.8319304.