Freie Universität Berlin

IoT Network Security

Untersuchung der verschieden Schutzmechanismen in Smart Home Netzwerken, SoSe23

Autoren: Zohreh Asadi, Aiman Al-Hazmi
Dozentin: Dr. Larissa Groth

31. Juli 2023

Inhaltsverzeichnis

1 Einführung

Die Bedenken hinsichtlich der Sicherheit und des Datenschutzes von IoT-Geräten nehmen zu, da Smart Homes immer beliebter und verbreiteter werden. Es gibt Hunderttausende Studien, die Sicherheitsrisiken und Gegenmaßnahmen in Smart-Home-Umgebungen identifizieren und bewerten. Dieser Artikel bietet einen Überblick über Sicherheits- und Datenschutzprobleme in Smart-Home-Netzwerken und konzentriert sich auf die Untersuchung verschiedener Schutzmechanismen. Die Rezension basiert auf einer Auswahl mehrerer Forschungsartikel und Bücher. Ziel des Beitrags ist es, einen detaillierten Überblick über die aktuellen Forschungsergebnisse zu geben und Empfehlungen und Ansichten zur Implementierung von Sicherheitsmechanismen in einer Smart-Home-Umgebung zu geben.

Der Artikel beginnt mit einer Einführung in verschiedene Netzwerkarchitekturen und Kommunikationsprotokolle. Das Smart-Home-Netzwerk basiert auf einem grundlegenden Schichtenmodell bestehend aus der Sensor- und Aktorschicht, der Netzwerkschicht und der Anwendungsschicht. Die Kommunikation zwischen Geräten erfolgt über verschiedene Protokolle wie Wi-Fi, Bluetooth, Zigbee, Z-Wave, MQTT und CoAP. Anschließend geht er auf die Gefahren und Risiken ein, die mit Smart-Home-Netzwerken verbunden sind, und betont die Bedeutung der Sicherheit in einer IoT-Umgebung. Darüber hinaus werden verschiedene Cyberangriffe besprochen, die häufig in IoT-Anwendungen auftreten, wie zum Beispiel: B. Channel-Angriffe, selektive Umleitungsangriffe, Sybil-Angriffe und Denial-of-Service-Angriffe (DoS).

Im weiteren Verlauf dieses Artikels werden die wichtigsten Sicherheitsmechanismen für Smart-Home-Netzwerke erörtert, wobei der Schwerpunkt auf der IoT-Authentifizierung und der Sicherheitsarchitektur liegt. Es werden Verschlüsselungs-, Authentifizierungs- und Zugriffskontrollmechanismen untersucht, darunter Technologien wie Advanced Encryption Standard (AES), Blowfish und RSA. Die Vorteile, Einschränkungen und Leistung dieser Verschlüsselungsalgorithmen werden diskutiert. Die Authentifizierung, insbesondere mittels CoAP, spielt eine wichtige Rolle bei der Gewährleistung der Sicherheit und Integrität der Kommunikation in einem Smart-Home-Netzwerk.

Insgesamt analysiert dieser Artikel Sicherheits- und Datenschutzprobleme in Smart-Home-Netzwerken und skizziert wichtige Sicherheitsmaßnahmen. Ziel ist es, das Verständnis für Cybersicherheit im Bereich IoT zu fördern und bei der Gestaltung einer sicheren Smart-Home-Umgebung zu helfen.

2 Grundlagen von Smart Home Netzwerken

2.1 Architektur von Smart Home-Netzwerken

In der Literatur wurden mehrere IoT-Architekturen vorgeschlagen. Die IoT-Architektur berücksichtigt wichtige Faktoren wie Dienstqualität (QoS), Datenschutz, Zuverlässigkeit, Integrität usw.(1). Die allgemeine IoT-Architektur besteht aus drei Hauptschichten: der Wahrnehmungs- oder physischen Schicht, der Netzwerkschicht und der Anwendungsschicht. In diesem Abschnitt befassen wir uns mit der grundlegenden Architektur und Serviceorientierung des IoT.

Basic layered architecture

Die Verwendung dieser Architektur ist weit verbreitet, um Komponenten zu strukturieren und mit dem System zu interagieren. Diese Architektur bietet einen modularen Ansatz, der Flexibilität, Skalierbarkeit und Interoperabilität bietet(1). Innerhalb dieser Architektur werden verschiedene Schichten verwendet, um spezifische Funktionen auszuführen.

1. **Die Wahrnehmungsschicht** ist dafür verantwortlich, Daten von verschiedenen Sensoren und Geräten in der physischen Welt zu erfassen. Physische Ereignisse werden in digitale Daten umgewandelt, die verarbeitet und übertragen werden können.

2. **Die Netzwerkschicht** sorgt für die Kommunikation zwischen Geräten/Sensoren in der Wahrnehmungsschicht und den höheren Schichten der Architektur. Es umfasst Protokolle und Technologien wie Wi-Fi, Bluetooth, Zigbee, Mobilfunknetze und andere Konnektivitätsoptionen. Diese Schicht gewährleistet eine zuverlässige Datenübertragung und verwaltet die Kommunikation zwischen Geräten.
3. **Die Anwendungsschicht** ist die oberste Schicht der Architektur und besteht aus verschiedenen IoT-Anwendungen und -Diensten. Diese Anwendungen nutzen verarbeitete Daten zur Durchführung spezifischer Funktionen wie Überwachung, Steuerung, Automatisierung, Analyse und Entscheidungsfindung. Beispiele für IoT-Anwendungen sind Smart-Home-Systeme, industrielle Überwachungssysteme sowie Gesundheits- und Umweltüberwachung.

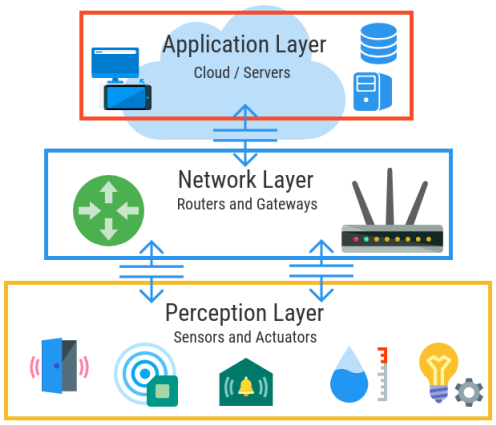

Abbildung 1: Basic layered architecture Quelle.

Serviceorientierte Architektur

Die IoT-Architektur kann dienstorientiert sein. Ihre vier Schichten sind in Abbildung 2 dargestellt. Die Erfassungsschicht (Sensing layer) integriert die Hardware. Die Netzwerkschicht unterstützt die Datenübertragung über das Netzwerk. Die Dienstschicht (Service layer) erstellt und verwaltet Dienste. Die Schnittstellenschicht (Interface layer) ermöglicht die Interaktion zwischen Benutzer und Anwendungen(1).

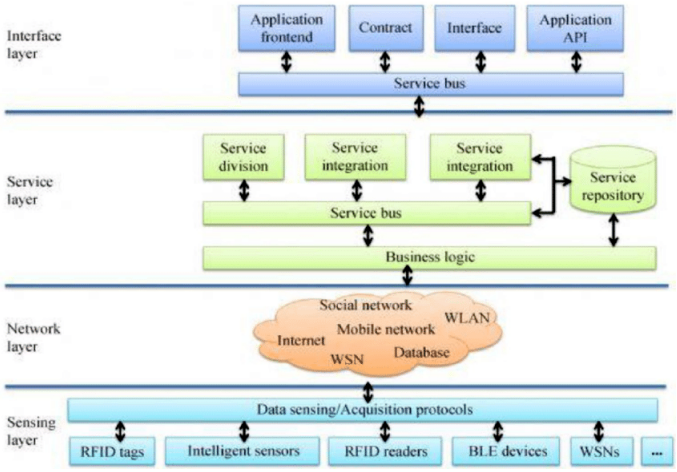

Abbildung 2: Serviceorientierte Architektur Quelle

Smart Homes Architektur

Abbildung 3 zeigt ein Beispiel einer Smart-Home Architektur. Ein Smart Home besteht aus verschiedenen Geräten, die mit einem lokalen Netzwerk (LAN) verbunden sind. Diese Geräte kommunizieren untereinander über Kommunikationsprotokolle wie ZigBee, Bluetooth, Wi-Fi oder andere RF-Technologien. Die Smart-Home-Architektur besteht aus einem lokalen Netzwerk, das Geräte, Sensoren und Aktoren im Haus verbindet. Ein mit dem lokalen Netzwerk verbundener Server verwaltet Geräte, protokolliert, generiert Berichte und reagiert auf Benutzerbefehle. Der Server kann auch über APIs mit Cloud-Diensten interagieren und so zusätzliche Funktionalität und Skalierbarkeit bieten. Darüber hinaus sind Smart-Home-Geräte mit dem Internet verbunden, sodass Benutzer über die Anwendung aus der Ferne auf das System zugreifen und es steuern können(2).

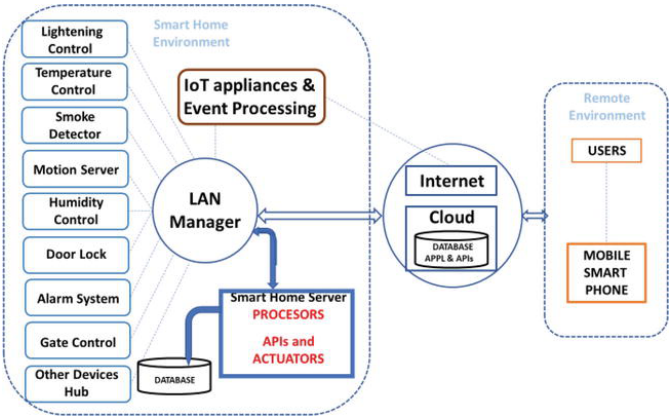

Abbildung 3: Smart Homes Architektur (2)

2.2 Kommunikationsprotokolle in Smart Home Netzwerken

Um in einem Smart Home effektiv kommunizieren zu können, müssen Geräte und Sensoren miteinander verbunden sein. Diese Verbindung basiert auf Kommunikationsprotokollen, die definieren, wie Daten zwischen diesen Geräten ausgetauscht werden. Diese Protokolle werden von verschiedenen Organisationen erstellt, um eine reibungslose und standardisierte Kommunikation in Smart-Home-Umgebungen sicherzustellen.

Die Wahl des Kommunikationsprotokolls für ein Smart-Home-System hängt von vielen Faktoren ab, unter anderem Verteilungsumgebung, Netzwerkgröße, Reichweite, Datenrate, Stromverbrauch und Netzwerkkonfiguration (2)(3). Kommunikationsprotokolle können in drei Hauptgruppen unterteilt werden: kabelgebunden, drahtlos und hybrid. Kurze Erläuterungen zu häufig genannten Technologien für Smart-Home-Anwendungen:

1. ZigBee: bietet eine robuste und skalierbare Lösung für die Verbindung und Steuerung verschiedener IoT-Geräte wie Sensoren, Aktoren, Beleuchtung, Thermostate und Smart-Home-Geräte(1)(3). Der ZigBee-Protokollstapel besteht aus vier Schichten: der Anwendungsschicht, der Netzwerk- und Sicherheitsschicht, der MAC-Schicht und der PHY-Schicht(1).
2. Bluetooth: ist ein drahtloses Medium mit kurzer Reichweite, das zum Datenaustausch zwischen zwei Geräten über eine kurze Distanz verwendet wird(3).
3. WiFi: ermöglicht drahtlose Kommunikation und ermöglicht den Zugriff auf das Internet in Form eines Funksignals (1).
4. 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks): ist ein Netzwerkprotokoll, das IPv6-Kommunikation über drahtlose Netze mit geringem Stromverbrauch ermöglicht(3). Es bietet eine Lösung für die Anbindung ressourcenbeschränkter IoT-Geräte an IPv6-Netzwerke, indem es

die Header-Größen reduziert und den Adressierungsraum optimiert, so dass diese Geräte auf IP-basierte Dienste und Kommunikation zugreifen können(3).

5. TLS (Transport Layer Security): ist ein Protokoll, das eine sichere und zuverlässige Kommunikation zwischen Anwendungen über das Internet gewährleistet(3)(4). TLS befindet sich zwischen der Anwendungsschicht (z. B. HTTP) und der Transportschicht (z. B. TCP) und ermöglicht die Server-Authentifizierung, eine sichere und zuverlässige Verbindung sowie die Client-Authentifizierung mit Hilfe der Public-Key-Kryptografie(3). TLS ist auch die verbesserte Version von SSL, mit der gleichen Architektur und den gleichen Protokollen (3). Die einzigen Unterschiede sind einige Änderungen bei den Sicherheitsparametern und der Berechnung von MAC, Schlüsselblock und digitaler Signatur. Außerdem werden einige neue Warnmeldungen und Pseudozufallsfunktionen eingeführt, um die Sicherheit im Vergleich zu SSL zu erhöhen(5)(3). Sowohl SSL als auch TLS unterstützen eine breite Palette von Verschlüsselungsalgorithmen. Die Wahl des Verschlüsselungsalgorithmus erfolgt während des Handshake-Prozesses auf der Grundlage der von Client und Server unterstützten Algorithmen.

6. Message Queuing Telemetry Transport (MQTT): funktioniert mit dem Übertragungs-/Internet -Steuerungsprotokoll (TCP/IP), der eine Verbesserung ermöglicht, die mit Verlusten behandelt wird. Es handelt sich um ein Client-Server-Nachrichtenprotokoll, das Nachrichten mit minimiertem Transport-Overhead übermittelt (1). Es gibt drei Dienstgüteklassen (QoS) für das MQTT-Protokoll, nämlich: 'at most once', das sicherstellt, dass Nachrichten je nach Verfügbarkeit der Betriebsumgebung zugestellt werden, 'at least once', das die Ankunft der Nachricht sicherstellt und 'genau einmal', das die Ankunft der Nachricht genau einmal sicherstellt(1)(3). MQTT verfügt über einen erstaunlichen Mechanismus zur Benachrichtigung über eine anormale Unterbrechung der Verbindung.

7. Constrained Application Protocol (CoAP): ist ein Protokoll der Anwendungsschicht, das speziell für eingeschränkte Geräte entwickelt wurde(1)(3). Es ermöglicht einen effizienten Datenaustausch mithilfe eines REST-Modells (Representative State Transition) unter Verwendung von Methoden wie GET, PUT, POST und DELETE. Die Kommunikation zwischen zwei Knoten wird nicht gesichert(6). Für eine sichere Kommunikation verwendet CoAP DTLS (Data Transport Layer Security), da die Implementierung von TLS (Transport Layer Security) bei begrenzten Geräten schwierig ist. UDP-basiertes DTLS eignet sich für beengte Umgebungen(3). Eingeschränkte Geräte verfügen oft über begrenzte Ressourcen, z.B 8-Bit-Mikrocontroller mit minimalem RAM und ROM. Darüber hinaus weisen begrenzte Netze oft eine hohe Paketfehlerrate auf. CoAP geht auf diese Einschränkungen ein, indem es Webanforderungen wie Multicasting erfüllt, den Overhead minimiert und Einfachheit bietet (1).

8. WebSocket: bietet eine zweiseitige Kommunikation mit dem Server, die aus einem Handshake und einem anschließenden Message Framing besteht. Die Interaktion zwischen Browser und Webserver nimmt zu, da sie die Übertragung von Echtzeitdaten vom und zum Server erleichtert. Websocket bietet Vollduplex-Kommunikation.(1)

2.3 Bedrohungen und Risiken für Smart Home-Netzwerke

Die Gewährleistung der Sicherheit in IoT-Umgebungen ist zu einer großen Herausforderung und einem wichtigen Barriers für den Fortschritt der Smart-Home-Automatisierung geworden (7) .Diese Anerkennung unterstreicht ihre Komplexität und Bedeutung. Je mehr Smart-Home- und IoT-Geräte entwickelt werden, desto schneller nehmen die mit deren Nutzung verbundenen Gefahren, Bedrohungen und Risiken zu. Hier überprüfen wir zwei verschiedene Studien. In (8) verwenden sie eine Methode namens 'Taxonomy'; welche ist ein System zur Inhaltsverwaltung, das Informationen basierend auf als Metadaten gespeicherten Begriffen gruppiert, und durch die Analyse der vorhandenen Literatur zu Smart Homes wurde eine Reihe von Bedrohungen identifiziert, die speziell für den Bereich Smart Home gelten. Basierend auf dieser Untersuchung gibt es 28 verschiedene Arten von Threads, die in drei verschiedene Gruppen eingeteilt werden können, nämlich: Unintentional, Malfunction und Intentional Threads.

2.3.1 Threads

Intentional Threads: Vorsätzlicher Missbrauch birgt Sicherheitsrisiken für Smart-Home-Geräte durch Aktivitäten wie Identitätsbetrug, Denial-of-Service-Angriffe und Datenmanipulation. Diese Aktionen können verschiedene zerstörerische Auswirkungen haben, darunter unbefugte Änderungen an Richtlinien, Identitätsdiebstahl und die Ausnutzung von Smart-Home-Diensten.

Malfunction Threads: Ausfälle im Internet oder Hardware- oder Softwarefehler, die zu internen Störungen und Kommunikationskanalausfällen führen können, sind die häufigsten Beispiele für Störungsthreads und -risiken in Smart Homes. Ein weiteres Beispiel sind Schäden durch Geräte oder Sensoren Dritter, die zu Fehlfunktionen des Geräts führen, die einen Austausch und einen vorübergehenden Funktionsverlust erfordern. Darüber hinaus beeinträchtigt die Installation nicht vertrauenswürdiger Hardware von Drittanbietern die Gerätesicherheit.

Unintentional Threads: Zu den unbeabsichtigten Threads gehören der Erhalt von Informationen aus unbekannten Quellen oder zufällige Änderungen von Daten oder Richtlinien. Dies geschieht normalerweise, wenn die Sicherheitsrichtlinien des Geräts nicht richtig eingestellt sind oder die Installation schwach ist. Die Nachteile dieser Art von Threads können Fehlfunktionen und Funktionsverluste sein.

In einer weiteren Studie (9) wurde die Risikoexposition von SHAS (sichere Hashing-Algorithmen) mithilfe der Methode der Information Security Risk Analysis (ISRA) untersucht. Laut dieser Studie ist die Risikoanalyse im Internet der Dinge das größte Sperre für die Weiterentwicklung dieser Technologie. Der Zweck der Studie dieser Gruppe besteht darin, die Risiken zu verstehen, die mit der Nutzung und dem Missbrauch von Informationen in Smart Homes und Gebäuden verbunden sind, was für die Gewährleistung der Sicherheit in IoT-Umgebungen sehr wichtig ist. Von den 32 von dieser Gruppe untersuchten Risiken wurden 9 Fälle als geringes Risiko und 4 Fälle als hohes Risiko eingestuft. Hochrisikofaktoren hängen entweder mit dem menschlichen Faktor oder mit den Softwarekomponenten des Systems zusammen. Wenn wir die Risiken in Bezug auf Software oder Hardware kategorisieren, hängt das wahrscheinlichste Risiko in der Liste der softwarebezogenen Risiken mit einer unzureichenden Reaktionsfähigkeit des internen Gateways zusammen. Beispielsweise werden Systemereignisse nicht aufgezeichnet, was die Situation erschweren kann, später nachvollziehen. Der höchste Risikowert steht im Zusammenhang mit unbefugten Änderungen an Systemfunktionen in mobilen Anwendungen. Beispielsweise können Endbenutzer ohne entsprechende Anmeldeinformationen auf Systemressourcen zugreifen. In der Kategorie der hardwarebezogenen Risiken sind die schwerwiegendsten Folgen hingegen mit unbefugten Änderungen bzw. Manipulationen an physischen Sensoren oder internen Gateways verbunden. Die höchsten Risikowerte beziehen sich auf unbefugte Änderungen/Manipulationen des internen Ports, beispielsweise das Zurücksetzen des Systems zur Wiederherstellung des Standardkennworts.

2.3.2 Cyber-attacks

Nachdem die verschiedenen Kategorien von Bedrohungen untersucht wurden und Einblicke in die Risiken und die Risikoexposition von Secure Hashing Algorithms (SHAS) gewonnen wurden, wird es Zeit, in die Welt der Cyberangriffe einzutauchen, die potenzielle Bedrohungen für Smart Homes und Gebäude im Internet der Dinge (IoT) darstellen. Das Verständnis und der Schutz vor diesen Angriffen sind entscheidende Aspekte, um eine robuste Sicherheit in IoT-Geräten zu gewährleisten.

Einige der häufig vorkommenden Cyberangriffe auf IoT-Anwendungen:

***Sinkhole − Angriff* :** Ein Sinkhole-Angriff ist eine Art Angriff auf Netzwerkebene, der auftritt, wenn Daten während der Übertragung weitergeleitet werden. Bei diesem Angriff werden alle durch das Netzwerk fließenden Daten an einen kompromittierten Knoten im Netzwerk umgeleitet (10) .Dieser Angriff reduziert den Datenverkehr und täuscht den Absender und das Netzwerk vor, dass das Paket sein eigentliches Ziel erreicht hat. Bei diesem Angriff handelt es sich um einen aktiven Angriff, der durch die Generierung von Datenverkehr und die Unterbrechung von Routing-Pfaden zu einem Denial-of-Service-Angriff (DoS) führen kann (11) .

***SelektiverWeiterleitungsangriff* :** Bei diesem Angriff werden ein oder mehrere Knoten im Netzwerk von den Hackers einnimmt. Wenn einer der Knoten ein Paket als selektiven Weiterleitungsangriff verwirft, helfen andere dabei, den Angriff zu verschleiern. Dies führt zu Paketverlusten aufgrund von Interferenzen und ist schwer zu erkennen. Dieser Angriff kann dazu führen, dass unvollständige Informationen übertragen werden und deren Integrität beeinträchtigt wird. In manchen Anwendungen können unvollständige Informationen sogar gefährlicher sein als keine Informationen (12) .

***Sybil − Angriff* :** Es ist ein weiterer Angriff auf Netzwerkebene, bei dem Angreifer Knoten manipulieren und mehrere Identitäten für einen einzelnen Knoten erstellen. Diese Art von Angriff gefährdet das gesamte System und kann zu Redundanz und falschen Informationen führen (13) .In IoT-Netzwerken werden Sensoren in Objekte eingebettet, die Sensorik und Kommunikation integrieren. Bei diesem Angriff wird eine gefälschte Identität generiert. Diese gefälschten Identitäten können falsche Berichte generieren, den Spam-Verkehr erhöhen und die Privatsphäre durch Malware und Phishing gefährden (14) .

***DoS − Angriff* :** Ein DoS-Angriff kann sowohl auf der Netzwerkebene als auch auf der Anwendungsebene erfolgen. Bei diesem Angriff überschwemmt der Angreifer das Netzwerk mit nutzlosem Datenverkehr (mehrere gleichzeitige Angriffe auf denselben Server) mit dem Ziel, Netzwerkressourcen zu belasten. Dieser Angriff macht das Netzwerk für legitime Benutzer unzugänglich (15) .Auf der Anwendungsebene ist der Angriff sogar noch ausgefeilter, da er die Netzwerkverteidigung umgeht und vertrauliche Informationen unter der Kontrolle des Angreifers stiehlt. Dieser Angriff kann auch dazu führen, dass das gesamte Netzwerk heruntergefahren wird und nicht mehr verfügbar ist(13).

2.4 Wichtige Schutzmechanismen zur Sicherung von Smart Home-Netzwerken

Um die Sicherheit eines mit dem Internet verbundenen Smart-Home-Systems zu gewährleisten, ist es wichtig, sieben Schlüsselkonzepte der Computersicherheit zu berücksichtigen und anzugehen (3)(16):

1. **Authentifizierung:** Der Prozess der Überprüfung der Identität einer Person, eines Systems oder eines Objekts, um diese Identität sicherzustellen.
2. **Autorisierung:** Der Prozess der Bestimmung, welche Berechtigungen, Zugriffe oder Berechtigungen einer authentifizierten Person, einem authentifizierten System oder einer authentifizierten Entität gewährt werden. Es definiert, auf welche Ressourcen oder Informationen sie Zugriff haben und welche Maßnahmen sie ergreifen können.
3. **Vertraulichkeit:** Die Sicherstellung, dass Informationen nur von autorisierten Personen eingesehen oder genutzt werden können.
4. **Integrität** der Daten/Nachrichten: Die Sicherstellung, dass Daten oder Nachrichten während der Speicherung, Übertragung oder Verarbeitung nicht unabsichtlich verändert oder manipuliert wurden.
5. **Accountability:** Die Fähigkeit, Handlungen und Aktivitäten einer Person, eines Systems oder einer Entität nachzuverfolgen und für diese verantwortlich zu sein. Es ermöglicht die Identifizierung von Personen, die für bestimmte Aktionen verantwortlich sind.
6. **Verfügbarkeit:** Informationen, Systeme oder Ressourcen sollten jederzeit und für autorisierte Benutzer zugänglich sein. Es beinhaltet die Gewährleistung, dass Dienste oder Ressourcen nicht durch Ausfälle, Angriffe oder andere Störungen beeinträchtigt werden.
7. **Nicht-Abstreitbarkeit:** Die Fähigkeit, die Authentizität und Integrität einer Kommunikation oder Transaktion nachzuweisen und sicherzustellen, dass die beteiligten Parteien später nicht leugnen können, dass sie an der Kommunikation oder Transaktion beteiligt waren.

Verschiedene Arten von Angriffen können die Sicherheit physischer Objekte oder die Sicherheit kognitiver Daten gefährden. Zum Schutz der Wahrnehmungsebene (perception layer) sollten verschiedene Sicherheitsmethoden eingesetzt werden, darunter B. Authentifizierungs- und Zugriffskontrollverfahren, Datenintegritäts- und Sicherheitsverfahren sowie Schlüsselverwaltungssysteme(17).

Die Netzwerkschicht leidet unter mehreren Sicherheitsproblemen und verschiedenen Arten von Angriffen, wie z. Man-in-the-Middle-Angriffe, Listening-Angriffe, DoS-Angriffe, Replay-Angriffe, Authentifizierungsprobleme, Datenintegrität und Datenschutz, Routing-Sicherheitsprobleme und Datenschutzrisiken(17)(3).

IoT-Sicherheitsarchitekturen

IoT-Sicherheitsarchitekturen Die IoT-Sicherheitsforschung hat zur Entwicklung verschiedener Architekturen geführt, darunter Cloud Computing (CoT), Fog Computing, Edge Computing und Gateway-Architektur. Für die Kommunikation mit IoT-Geräten werden üblicherweise Middleware-Systeme verwendet(3). Diese Architekturen zielen darauf ab, die Sicherheit zu verbessern und Lösungen für komplexe Verarbeitung, echtzeitnahe Reaktionen und zentralisierte Koordinationbereitzustellen. Die Gateway-Architektur bietet Proxy-Unterstützung auch dann, wenn keine Verbindung besteht(3).

Hier ist eine verkürzte Liste einiger sicherer IoT-Architekturen (3):

1. FIWARE (2011): ist eine IoT-Middleware mit einem Framework, das eine Vielzahl von Plugins unterstützt. Die Sicherheit wird durch Plugins für die Identitätsverwaltung, die Autorisierungspolitik und die Punkte zur Durchsetzung von Richtlinien implementiert.
2. IoT Cloud on CoAP(2014): Cloudbasierte IoT-Architektur basierend auf CoAP und unter Verwendung von DTLS für die Sicherheit.
3. IAGW(2016): Integrierte Gateway-Architektur mit Standardschnittstellen für Smart-Home-Umgebungen, inklusive Sicherheitsmodul zur Authentifizierung, Autorisierung und Verschlüsselung.
4. Smart-Home-Automatisierung mit WSN(2019): verwendet den Triangle Based Security Algorithm (TBSA), um eine energieeffiziente Datenverschlüsselung zu gewährleisten und so ein sicheres IoT-basiertes Smart-Home-Automatisierungssystem zu schaffen.
5. SH-BlockCC (2019): Dies ist eine sichere IoT-Smart-Home-Architektur, die auf Blockchain- und Cloud-Computing-Technologie basiert. Blockchain ist eine dezentrale und verteilte Datenbank, die Transparenz, Integrität und Sicherheit bieten kann(18)(19). Das Modell verwendet die MCA-Methode (Multivariate Correlation Analysis), um den Netzwerkverkehr zu analysieren und die Korrelation zwischen Verkehrsmerkmalen zu bestimmen.

Die folgende Tabelle 4 fasst die Sicherheitsarchitekturen und die Sicherheitsziele zusammen, die sie jeweils zu schützen versprechen (3).

security architectures	AuthN	AuthZ	Cf'ty	D In'ty	Acb'ty	Av'ty	N-R'ion
FIWARE (2011)	YES	YES	YES	YES	YES		
IoT Cloud on CoAP (2014)	YES		YES	YES			
IAGW (2016)	YES	YES	YES	YES			
TBSA for Smart Home (2019)			YES	YES			
SH-BlockCC (2019)	YES		YES	YES		YES	YES

Abbildung 4: Matrix of security architectures and security goals (Authentication (AuthN), Authorization (AuthZ), Confidentiality (Cf'ty), Data Integrity (D In'ty), Accountability (Acb'ty), Availability (Av'ty), Non-Repudiation (N-R'ion))

SH-BlockCC

Die Blockchain-Technologie kann als wichtiger Sicherheitsmechanismus zum Schutz von Smart-Home-Netzwerken eingesetzt werden. Unter Blockchain versteht man eine dezentrale Struktur, in der es keine einzige Autorität gibt, die Transaktionen genehmigt. Die Transaktionen beziehen sich in diesem Kontext auf die Interaktion und Kommunikation zwischen lokalen Smart-Home Geräten und den Overlay-Netzwerken. Abhängig von der Funktionalität können unterschiedliche Arten von Transaktionen zwischen Geräten und Overlay-Netzwerken stattfinden, beispielsweise Zugriffstransaktionen, Speichertransaktionen und Überwachungstransaktionen. Diese Transaktionen werden auf der lokalen Blockchain verwaltet und gespeichert. Um das Vertrauen in die Blockchain sicherzustellen, müssen die Knoten im Netzwerk einen Konsens zur Annahme von Transaktionen erzielen(18), (19), (3).

Nach dem Hinzufügen eines neuen Geräts zum Smart-Home Netzwerk führt das Netzwerk eine Genesis-Transaktion durch, um die Anwesenheit des Geräts zu registrieren und zu überprüfen. Diese Transaktion enthält wichtige Geräteinformationen wie z.B Seine ID und einzigartige Merkmale. Um eine sichere Kommunikation im Smart Home zu gewährleisten, wird bei der Genesis-Transaktion ein gemeinsamer Schlüssel mithilfe eines Verschlüsselungsalgorithmus wie Diffie-Hellman generiert. Mit diesem Schlüssel kann das neue Gerät sicher mit dem Netzwerk interagieren und sich authentifizieren. Miner, die für die Netzwerkwartung und -validierung verantwortlich sind, weisen Geräten basierend auf den von Smart-Home-Besitzern festgelegten Regeln gemeinsame Schlüssel zu. Auf diese Weise steuern Benutzer Netzwerktransaktionen und Geräte können sicher miteinander kommunizieren(18)(19).

Intrusion-Detection-Systeme (IDS)

Intrusion-Detection-Systeme (IDS) sind für die Netzwerk- und Datensicherheit von entscheidender Bedeutung. Sie erkennen und verhindern unbefugten Zugriff, und ein zentralisierter IDS-Ansatz kann den Einsatz bei begrenzter Stromversorgung vereinfachen. IDS arbeitet in drei Phasen: Überwachung, Analyse und Erkennung. Es verwendet Sensoren, um das Netzwerk oder den Host zu beobachten, Merkmale zu extrahieren und Muster während der Analyse zu identifizieren, und konzentriert sich bei der Erkennung auf die Identifizierung von Anomalien oder Eindringlingen. IDS entwickeln sich im Laufe der Zeit weiter und für verschiedene Systeme wurden unterschiedliche Methoden und Techniken vorgeschlagen. Durch die effiziente Verwaltung von Informationen, Diensten und Netzwerkverkehr verbessert IDS die Netzwerksicherheit, den Datenschutz und die Integrität(16)(20).

Verschlüsselung, Authentifizierung und Zugriffskontrolle

Bei der Authentifizierung wird die Identität einer Person oder eines Geräts überprüft, um sicherzustellen, dass es sich um das handelt, was es vorgibt zu sein. Dieser Sicherheitsmechanismus dient dazu, unbefugten Zugriff auf das System, Daten oder Ressourcen zu verhindern. Dies werden wir genauer später in diesem Artikel betrachten.

3 Verschlüsselung, Authentifizierung und Zugriffskontrolle in Smart Home-Netzwerken

3.1 Verschlüsselungstechnologien

Verschlüsselungstechnologien spielen eine entscheidende Rolle bei der Sicherung von sensiblen Informationen und der Gewährleistung der Privatsphäre. Für die Kryptographie wurden verschiedene Ziele aufgeführt, nämlich: Vertraulichkeit, Authentifizierung, Integrität, Bestreitbarkeit und Zugangskontrolle.

Verschlüsselungsalgorithmen können in zwei große Kategorien eingeteilt werden: Verschlüsselung mit symmetrischem Schlüssel (z. B. AES, Blowfish) und Verschlüsselung mit asymmetrischem Schlüssel (z. B. RSA).(21) . Symmetrische Verschlüsselung, auch als Geheimschlüsselverschlüsselung bekannt, ist eine grundlegende kryptografische Technik zur Sicherung sensibler Informationen. Dabei wird ein einziger geheimer Schlüssel für sowohl den Verschlüsselungs- als auch den Entschlüsselungsprozess verwendet. Auf der anderen Seite verwendet die asymmetrische Verschlüsselung, auch bekannt als Public-Key-Verschlüsselung, ein Schlüsselpaar: einen öffentlichen Schlüssel zum Verschlüsseln von Daten und einen privaten Schlüssel zum Entschlüsseln (3).

Im Folgenden besprechen wir zwei symmetrische Verschlüsselungsalgorithmen, nämlich AES (Advanced Encryption Standard) und Blowfish, sowie einen asymmetrischen Verschlüsselungsalgorithmus RSA.

3.1.1 AES:

AES (Advanced Encryption Standard) ist ein symmetrischer Verschlüsselungsalgorithmus, der mit Datenblöcken arbeitet. Es verwendet schlüsselbasierte Ersetzungs- und Permutationsoperationen, um Vertraulichkeit und Datenintegrität sicherzustellen. AES umfasst eine Schlüsselerweiterung zur Generierung von Rundenschlüsseln und durchläuft mehrere Runden von Substitutions-, Shift-, Shuffle- und XOR-Operationen. Der Verschlüsselungsprozess wandelt Klartext in Chiffretext um, während die Entschlüsselung die Schritte umkehrt, um den ursprünglichen Klartext abzurufen.(22) The AES design is based on a substitution-permutation network (SPN) and does not use the Data Encryption Standard (DES) Feistel network(23). Einerseits weist AES eine schlechte Leistung auf, da es mehr Rechenleistung benötigt und bei großen Datenblöcken mehr Ressourcen verbraucht. Andererseits ist es für kleine Pakete sogar besser als viele Algorithmen, einschließlich RC4, außerdem hat AES keine Wurmlöcher.(22) Wie in5 erwähnt, verfügt AES über eine schnelle Ver- und Entschlüsselung und verfügt über eine Schlüsselgröße von 128, 192 und 256 Bit. Seine Geschwindigkeit hängt vom Schlüssel ab, es ist aber sicher und funktioniert besser als viele Algorithmen, einschließlich DES und Triple-DES, und es ist sowohl in der Hardware als auch in der Software schnell(24), (22).

3.1.2 BLOWFISH:

Die Leistung des BLOWFISH-Algorithmus für unterschiedliche Paketgrößen mit und ohne Datentransformation ist besser als bei anderen Algorithmen und hat keine Wurmlöcher, aber er ist zeitaufwändig und verbraucht letztendlich mehr Strom(21). Die Ver- und Entschlüsselung im BLOWFISH-Algorithmus erfolgt schnell und die Schlüsselgröße liegt zwischen 32 und 448 Bit. Die Geschwindigkeit dieses Algorithmus hängt nicht vom Schlüssel ab. Es wird angenommen, dass es gesichert ist, aber es gibt weniger Versuche zur Kryptoanalyse. Dieser Algorithmus wurde nur für Software entwickelt(24), (21).

Um die Daten mit BLOWFISH-Algorithmus zu verschlüsseln, sind folgende Schritte erforderlich:

1. Speichern Sie die Schlüssel in einem Array; k1,k2,...kn und 1<=n<=14 und die Länge jedes Blocks beträgt 32 Bit.

$$32 * 14 = 448$$

2. Initialisieren Sie das Array mit der Nummer p; 1<=p<=18, die Länge jedes Wortes beträgt 32 Bit.

3. Substitutions-Boxen initialisieren, jede darf maximal 255 Bit lang sein

$$2^8 - 1 = 255$$

. 4. Initialisieren Sie jedes P-Array und S-Bixes mit Hexadezimalzahlen

5. Ausführen von XOR-Operationen zwischen k- und p-Arrays:

$$p1 = p1XORk1, ..., p14 = p14XORk14$$

$$p15 = p15XORk1, ..., p18 = p18XORk4.$$

6. Nehmen Sie 64-Bit-Klartexte und initialisieren Sie sie mit Null (000000).

Subkey wird generiert. Create figure for this algo

3.1.3 RSA:

RSA ist ein Asymmetric-Algorithm, dass auf Verschlüsselung mit öffentlichen Schlüsseln und digitalen Signaturen basiert hat. Bei RSA sind Verschlüsselungsschlüssel öffentlich verfügbar, während Entschlüsselungsschlüssel privat bleiben. Dadurch kann jeder eine Nachricht mit dem öffentlichen Verschlüsselungsschlüssel des Empfängers verschlüsseln, aber nur der beabsichtigte Empfänger kann sie mit dem entsprechenden privaten Entschlüsselungsschlüssel entschlüsseln. Seine Sicherheit basiert auf der Schwierigkeit von folgenden Zahlen:

„p“ und „q“: Dabei handelt es sich um unterschiedliche Primzahlen, die zufällig ausgewählt werden. Die Sicherheit von RSA beruht auf der Schwierigkeit, das Produkt dieser beiden Primzahlen zu faktorisieren. Diese Primzahlen werden geheim gehalten und zur Generierung des privaten Schlüssels verwendet.

„n“: Dies ist das Produkt der beiden Primzahlen „p“ und „q“ (n = p*q). Es ist Teil sowohl des öffentlichen als auch des privaten Schlüssels. Der Wert von „n“ ist öffentlich und kann jedem bekannt sein. Weitere in RSA verwendete Variablen sind:

„e“: Dies ist der öffentliche Exponent, der normalerweise eine kleine Primzahl ist. Es ist Teil des öffentlichen Schlüssels und wird so gewählt, dass es eine Primzahl zum Wert von (p-1)*(q-1) ist, wodurch sichergestellt wird, dass es eine modulare Umkehrung gibt.

„d“: Dies ist der private Exponent, der aus den Primfaktoren und dem öffentlichen Exponenten abgeleitet wird. Es wird geheim gehalten und zur Entschlüsselung verwendet. Der private Exponent wird mithilfe der modularen Arithmetik und des erweiterten euklidischen Algorithmus berechnet.

Die digitale Signatur ermöglicht die Authentifizierung und Nichtabstreitbarkeit von Nachrichten. Um die Authentizität einer gesendeten Nachricht zu überprüfen, verwendet der Empfänger den Entschlüsselungsschlüssel des Absenders, um die Signatur zu entschlüsseln, die durch Verschlüsselung des Nachrichten-Hashs mit dem privaten Schlüssel des Absenders erstellt wird. Der resultierende Hash wird dann mit dem neu berechneten Hash der empfangenen Nachricht unter Verwendung des öffentlichen Verschlüsselungsschlüssels des Absenders verglichen. Wenn die beiden Hashes übereinstimmen, bestätigt dies, dass die Nachricht nicht manipuliert wurde und vom angeblichen Absender stammt.

Die Stärke der RSA-Sicherheit zeigt sich darin, dass es keine erfolgreichen Versuche gibt, sie zu knacken. Allerdings ist dieser Algorithmus im Vergleich zu symmetrischer Verschlüsselung rechenintensiv. Daher wird RSA häufig in Verbindung mit schnelleren symmetrischen Algorithmen verwendet. Darüber hinaus schaffen Timing-Angriffe und Herausforderungen im Zusammenhang mit der Schlüsselverteilung potenzielle Schwachstellen, die Gegenmaßnahmen wie spezielle Hardware- und Softwareimplementierungen erfordern. (25)

	symmetric		asymmetric
	AES	BLOWFISH	RSA
Vorteil	•Schnelle Ver- und Entschlüsselung (HW,SW)	•Gute Leistung, keine Wurmlöcher	•Sichere Übertragung
	•Flexible Schlüsselgrößen (128, 192, 256 Bit)	•Effiziente Ver-/Entschlüsselung	•Digitale Signaturen
	•Bessere Sicherheit als DES und Triple-DES	•Variable Schlüsselgröße: 32-448 Bit	•Sicherer Algorithmus
	•Kein Wurmlöcher	•Weniger Kryptoanalyseversuche	
Nachteil	•Mehr Rechenleistung für große Datenblöcke	•Rechenintensiver als AES	•Rechenintensiver als symmetric-encryption
	•Nicht effizient für große Pakete	•Zeit- und Stromverbrauch	•Timing-Angriffe
		•Nicht Hardware-optimiert	•Spezielle HW- und SW-Implementierungen
		•Potenziell weniger analysiert	

Abbildung 5:

3.2 Authentifizierung

In einer Welt, in der alles miteinander verbunden ist, ist die Authentifizierung der erste Schritt zum Schutz unserer Daten. Die Authentifizierung ist ein häufiges Sicherheitsproblem, das sich auf die Art und Weise auswirkt, wie wir Apps anzeigen, verbinden und verwenden. Für das Internet der Dinge (IoT) wurden verschiedene Authentifizierungsarten entwickelt, darunter ressourcenbeschränkte Geräte und die Cloud. Es ist wichtig zu verstehen, dass sich die Authentifizierung im Laufe der Zeit kaum verändert hat, obwohl in der heutigen technologischen Welt ein einfaches Passwort nicht mehr ausreicht, um einen Benutzer zu verifizieren.

Es gibt mehrere Authentifizierungs- und Autorisierungsmethoden für smarte Geräte. Diese Methoden beruhen auf verschiedenen Faktoren, die sich in drei Kategorien einteilen lassen (3): 1. etwas, das eine Person weiß (z. B. ein Passwort oder ein Geheimcode). 2. etwas, das eine Person ist (z. B. ein Fingerabdruck). 3. etwas, das eine Person besitzt (z. B. eine Chipkarte oder ein digitaler Schlüssel). Diese Faktoren werden verwendet, um die Identität von Personen oder Systemen in Authentifizierungs- und Autorisierungsverfahren für smarte Geräte festzustellen.

Es lassen sich zwei grundlegende Arten von Authentifizierungsmechanismen unterscheiden: Intra-Domain (Authentifizierung innerhalb des Netzes) und Inter-Domain (Authentifizierung außerhalb des Netzes). Die Intra-Domain-Authentifizierung könnte durch einen der oben genannten Mechanismen erfolgen - Beweis durch Wissen, Beweis durch Besitz oder Beweis durch Eigentum. Bei der Inter-Domain-Authentifizierung sollten idealerweise zusätzliche Faktoren zum Tragen kommen, z. B. eine Kombination aus Wissens- und Besitznachweis.(3)

3.2.1 Taxonomie der IoT-Authentifizierungsschemas

Um die verschiedenen Authentifizierungsmethoden im IoT-Bereich besser zu verstehen, kann eine Taxonomie von IoT-Authentifizierungsschemas helfen. Eine klare und strukturierte Klassifizierung bietet einen systematischen Überblick über verfügbaren Methoden und Ansätze, die zur Authentifizierung verwendet werden können. Dies erleichtert den Vergleich, die Bewertung und die Auswahl der richtigen Lösung für bestimmte Anwendungsfälle und Sicherheitsanforderungen in einer IoT-Umgebung.

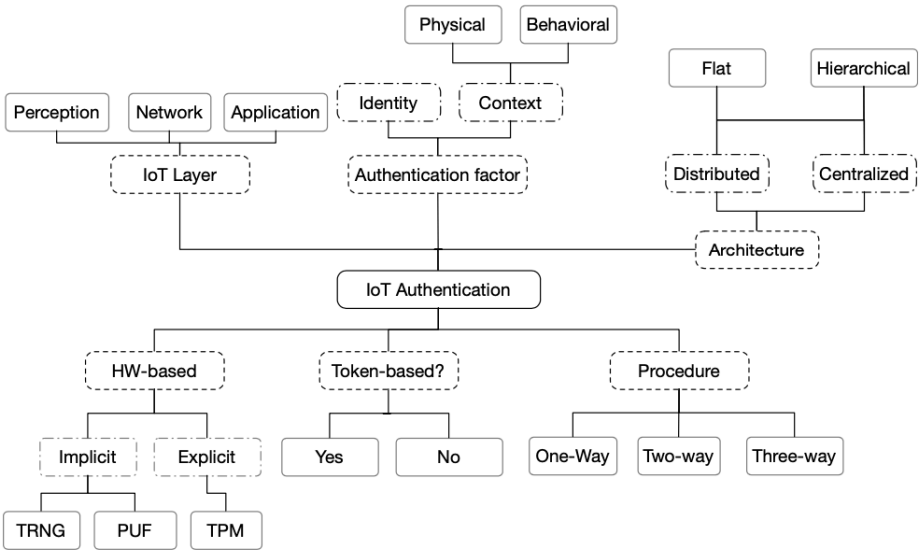

Abbildung 6: Taxonomie der IoT-Authentifizierungsschemas (26)

Hier sind einige Klassifizierungen, die in (3)(26) erwähnt und erläutert sind:

1. **Authentifizierungsfaktor:**
   a) Identität: Informationen, die eine Partei der anderen zum Zwecke der Authentifizierung zur Verfügung stellt. Identitätsbasierte Authentifizierungsschemas können einen Hash-Algorithmus (oder eine Kombination davon), einen symmetrischen Verschlüsselungsalgorithmus oder einen asymmetrischen Verschlüsselungsalgorithmus verwenden. Zum Beispiel Passwortbasiert Authentifizierung.
   b) Kontext: Der Kontext kann physisch oder verhaltensbezogen sein. Ein Beispiel für einen physischen Kontext ist die biometrische Authentifizierung.
2. **Token-based:**
   a) Token-basiert : Ein von einem Server generiertes Identitätstoken wird zur Authentifizierung verwendet, wie z. B. das OAuth2-Protokoll oder Open ID.
   b) Nicht-Token-basiert: Die Verwendung der Anmeldeinformationen(Benutzername/Passwort) bei jedem Datenaustausch.
3. **Authentifizierungsarchitektur:**
   a) Verteilt: Die Authentifizierung wird auf Knoten in der Nähe des Clients verteilt.
   b) Zentralisiert: Die Authentifizierung erfolgt über einen zentralen Server oder einen vertrauenswürdigen Dritten.
4. **Authentifizierungsverfahren:**
   a) Einweg-Authentifizierung (One-Way): Falls zwei Parteien miteinander kommunizieren möchten, wird nur eine Partei gegenüber der anderen authentifiziert und die andere Partei bleibt unauthentifiziert. Siehe Abbildung 7.

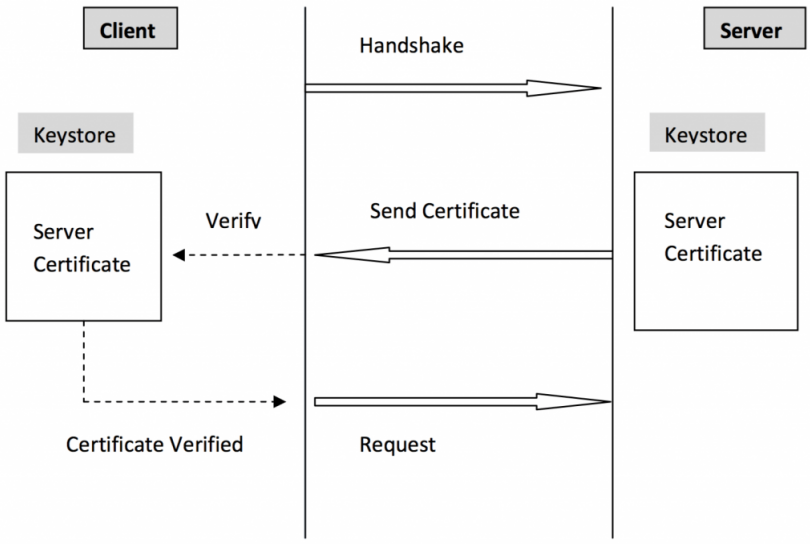

Abbildung 7: Einweg-Authentifizierung Quelle

b) Zwei-Wege-Authentifizierung (Mutual): Wird auch als gegenseitige Authentifizierung bezeichnet, wenn sich beide Parteien gegenseitig authentifizieren. Dies verhindert unbefugten Zugriff und Manipulation von Gerätezuständen. Es gibt verschiedene Methoden der gegenseitigen Authentifizierung:

   i. Gegenseitige Authentifizierung unter Verwendung eines gemeinsamen Schlüssels: Partei A und Partei B teilen sich vorher sicher einen Schlüssel. Partei A sendet eine Nachricht an Partei B, die mit einer zufälligen Herausforderung antwortet. Partei A verschlüsselt die Herausforderung mit dem gemeinsamen Schlüssel und sendet sie Partei B zurück. Sie tauschen also verschlüsselte Herausforderungen zur Authentifizierung aus.

   ii. Gegenseitige Authentifizierung mittels Public-Key-Kryptographie: Partei A und Partei B haben jeweils ein öffentlich-privates Schlüsselpaar. Wenn A mit B kommunizieren möchte, verschlüsselt A die Nachricht mit dem öffentlichen Schlüssel von B oder signiert sie mit seinem privaten Schlüssel. B entschlüsselt die Nachricht oder verifiziert die Signatur mit seinem privaten Schlüssel.

   iii. Gegenseitige Authentifizierung mit Zeitstempeln: Partei A sendet Partei B ihren aktuellen Benutzernamen und Zeitstempel mit einem gemeinsamen Schlüssel. B entschlüsselt den Zeitstempel, erhöht ihn, verschlüsselt ihn mit einem anderen gemeinsamen Schlüssel und sendet ihn zusammen mit dem Benutzernamen an Partei A zurück. Diese Techniken geben dem Kommunikationsprozess zusätzliche Sicherheit und stellen sicher, dass beide Parteien sich gegenseitig authentifizieren, bevor sie Nachrichten austauschen.

c) Drei-Wege-Authentifizierung(Three-Way): Hier authentifiziert die zentrale Autorität beide Parteien und hilft ihnen, sich gegenseitig zu authentifizieren.

3.2.2 Mutual TLS

Gegenseitiges TLS (mTLS), auch two-way TLS genannt, ist ein Sicherheitsmechanismus, der das herkömmliche TLS-Protokoll (Transport Layer Security) erweitert. Es bietet eine gegenseitige Authentifizierung zwischen Client und Server, indem beide Parteien ihre digitalen Zertifikate zur Überprüfung bereitstellen müssen.

Bei der gegenseitigen Authentifizierung authentifizieren sich Client und Server gegenseitig mit einem digitalen Zertifikat, z.B X.509-Zertifikatstandard, bevor sie die Kommunikation herstellen. Das X.509-Zertifikat ist der anerkannte Standard zur Authentifizierung und Zugangskontrolle in Netzwerken. Es bietet eine robuste Identitätsprüfung und sichere Kommunikation. Der weit verbreitete X.509-Zertifikatstandard umfasst einen öffentlichen Schlüssel, eine Identität, und kann von einer Zertifizierungsstelle (CA) signiert oder selbstsigniert werden.Der Prozess umfasst Root-CA-Zertifikate, Server-Zertifikate und Client-Zertifikate, die jeweils bestimmte Funktionen im Authentifizierungsprozess erfüllen (3) (27).

Allerdings können IoT-Geräte aufgrund ihrer begrenzten Rechenleistung Schwierigkeiten mit komplexen kryptografischen Algorithmen haben. Um dieses Problem zu lösen, wurden neuere IoT-Geräte mit mehr Standards und erhöhter Rechenleistung entwickelt, beispielsweise das Zertifikatformat IEEE 1609.2. Diese Geräte verwenden starke kryptografische Algorithmen, die speziell für IoT-Anwendungen optimiert sind. Dadurch wird die Authentifizierung und Sicherheit in IoT-Netzwerken verbessert, ohne die Ressourcen der Geräte übermäßig zu belasten.. (3)

Die gegenseitige TLS-Authentifizierung (mTLS) funktioniert folgendermaßen:

Wie die Abbildung 8 zeigt, der Client initiiert eine Verbindung zum Server. Der Server legt dem Client sein digitales Zertifikat vor, das vom Client mit Hilfe einer vertrauenswürdigen CA überprüft wird. Der Client legt dann sein eigenes Zertifikat zur gegenseitigen Authentifizierung vor, das der Server anhand einer vertrauenswürdigen CA verifiziert. Sobald beide Zertifikate erfolgreich verifiziert wurden, gewährt

der Server dem Client Zugang und stellt eine sichere Verbindung für eine verschlüsselte und vertrauenswürdige Kommunikation her (3) (27).

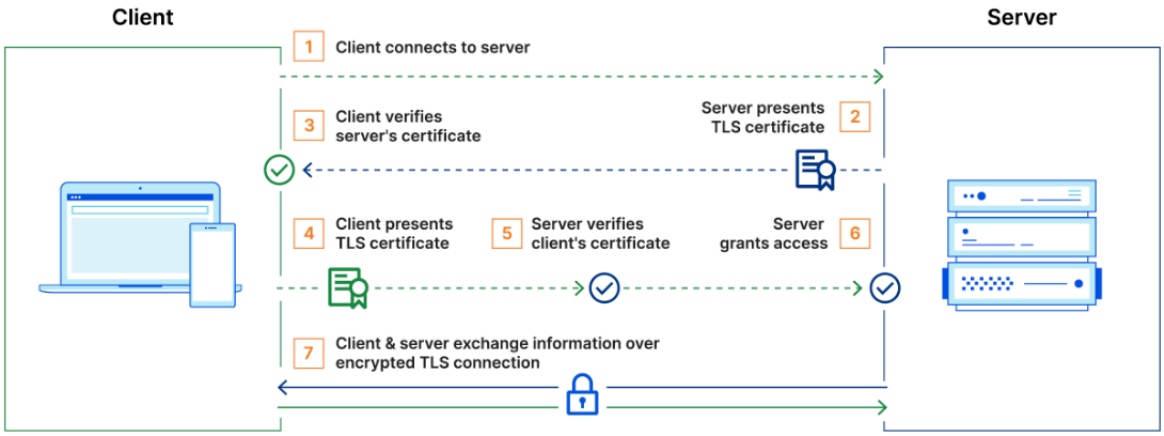

Abbildung 8: TLS-Authentifizierung Quelle

3.2.3 Lightweight CoAP-based Authentication

CoAP ist eine einfachere Variante von HTTP, die für Geräte mit eingeschränkten Ressourcen entwickelt wurde. In diesem Artikel(28) wird eine vereinfachte Authentifizierungsmethode basierend auf CoAP vorgeschlagen. Dabei handelt es sich um einen einfachen Algorithmus, der eine zuverlässige Alternative zur komplexeren Data Transport Layer Security (DTLS) sein kann. Die Authentifizierung besteht aus vier Handshake-Nachrichten zwischen dem Client und dem Server unter Verwendung des 128-Bit Advanced Encryption Standard (AES). Jeder Client sendet vor Beginn der Authentifizierung einen 128-Bit-Pre-Shared-Key Y_{i} an den Server. Jedem Gerät wird außerdem eine eindeutige Kennung zugewiesen, anhand derer der Server die Identität des Clients überprüft. Der Server verwaltet eine Pre-Shared-Key-Gruppe, die jeder Client-ID zugeordnet ist. Der Pre-Shared Key Y_{i} ist nur dem Server und dem Client bekannt, mit dem er kommunizieren möchte. Die Authentifizierung beginnt mit der Sitzungsinitiierung. Wenn die Client-ID nicht überprüft wird, schlägt die Sitzung fehl.

Hier sind die Schritte des Authentifizierungsschemas, wie in Abbildung 9 dargestellt:

1. Beginn der Sitzung: Der Client sendet eine Anforderungsnachricht an den Server unter Verwendung des Nachrichtentyps CON und der Methode POST. Die Nachricht enthält ein Token zur Identifizierung und eine eindeutige ID. Die Nachricht enthält auch Optionen, die die Art der Operation angeben, die an der Ressource des Servers (/authorize) durchgeführt wird. Mit dieser Anfrage wird der Server über den Beginn der Sitzung informiert.

2. Server Challenge: Der Server extrahiert die Geräte-ID aus der Anfrage und findet den entsprechenden Pre-Shared Key. Es antwortet dem Client mit einer verschlüsselten Payload mithilfe des AES-Algorithmus. Der Server generiert eine Pseudozufallszahl (nonceServer) und einen potenziellen Sitzungsschlüssel K_{s}. Die verschlüsselte Payload enthält das Ergebnis der XOR-Operation zwischen Y_{i} und K_{s} zusammen mit dem NonceServer.

   $\mathrm{AES}\{\mathrm{Y}_i, (Y_i \oplus K_s \mid \mathrm{nonceServer})\}$

Diese Challenge muss vom Client mit Y_{i} entschlüsselt werden, um eine Authentifizierungssitzung aufzubauen.

3. Client response and challenge: Der Client entschlüsselt die Challenge des Servers, um den potenziellen Sitzungsschlüssel K_s und die NonceServer zu erhalten. Ist dies erfolgreich, hat sich der Client authentifiziert. Anschließend generiert der Client eine neue verschlüsselte Payload. Zunächst führt der Client eine XOR-Operation zwischen nonceServer und Y_i durch. Anschließend wird das Ergebnis mit nonceClient kombiniert und mit K_s verschlüsselt, wie in folgender Gleichung dargestellt

   $$\mathrm{MP} = \mathrm{AES}\{\mathrm{K}_s, (nonceServer \oplus Y_i \mid \mathrm{nonceClient})\}$$

   Dabei wird die verschlüsselte Payload an den Server und die Pseudozufallszahl an den Client übertragen.

4. Server response: Der Server prüft die verschlüsselte Payload in der Client-Antwort. Es entschlüsselt die Payload, um nonClient und nonceServer abzurufen. Wenn die Client-Antwort keinen NonceServer enthält, führt dies zu einer nicht autorisierten Serverantwort. Andernfalls wird eine neue Payload erstellt, indem der NonceClient eingefügt und K_s hinzugefügt wird. Diese Payload wird mit Y_i verschlüsselt.

   $$\mathrm{M2P} = \mathrm{AES}\{Y_i\ , (\mathrm{nonceClient} \mid K_s)\}$$

   Nun sind sowohl Client als auch Server authentifiziert und können Datenpakete mit dem vereinbarten Sitzungsschlüssel K_s austauschen.

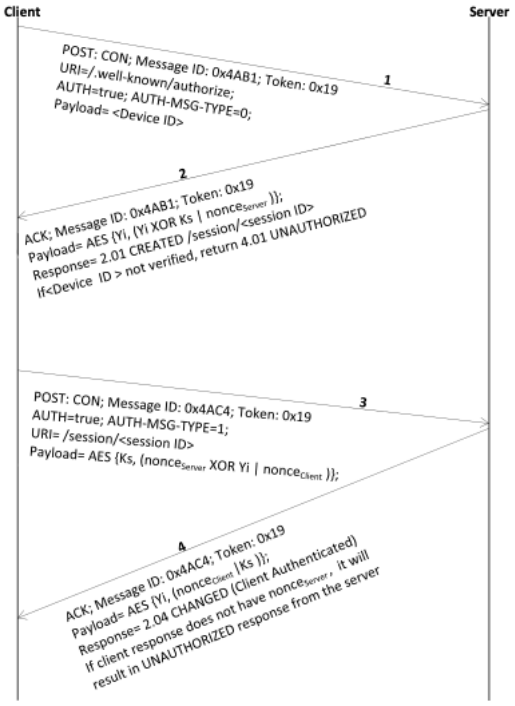

Abbildung 9: Lightweight CoAP-based Authentication (28)

Das vorgeschlagene Schema ist wirksam gegen Abhören, Schlüsselgenerierung, Ressourcenerschöpfung und Denial-of-Service-Angriffe, jedoch nicht gegen Sybil-Angriffe. Es gibt jedoch keine sichere Lösung, die alle Arten von Angriffen blockieren kann (28).

3.2.4 CoAP Payload Based Lightweight Authentication

In (6) wird eine neue Authentifizierungsmethode zwischen CoAP-Server und CoAP-Client vorgeschlagen. Diese Methode hat mehrere Vorteile: Sie erfordert keine zusätzlichen Protokolle, was die Komplexität

und den Ressourcenbedarf reduziert. Es ermöglicht die gegenseitige Authentifizierung zwischen Geräten mit nur zwei Nachrichten. Bei der Authentifizierung wird nur eine kleine Datenmenge gesendet, etwa 300 Byte. Diese Methode ist schnell und effizient, da auf beiden Geräten nur ein Roundtrip und eine Verarbeitungszeit erforderlich ist(6)

Das funktioniert so: Wenn ein Client auf eine Ressource zugreifen möchte, muss er dem Server gegenüber seine Identität nachweisen. Nachdem der Server die Identität des Clients überprüft hat, erstellt er eine Sitzung für diesen Client und antwortet darauf. Nach Erhalt der Antwort authentifiziert der Client den Server. Wenn sich sowohl der Server als auch der Client gegenseitig authentifizieren, können sie mithilfe eines speziell für diese Sitzung generierten Sitzungsschlüssels Daten austauschen. Bei dieser Methode teilen sich Client und Server einen 128-Bit-AES-Schlüssel. Diese Schlüssel werden zum Zeitpunkt der Herstellung in das Gerät einprogrammiert, wobei davon ausgegangen wird, dass die Geräte vor physischen Angriffen geschützt sind. Darüber hinaus wird jedem Gerät bei der Herstellung eine eindeutige 64-Bit-ID zugewiesen.

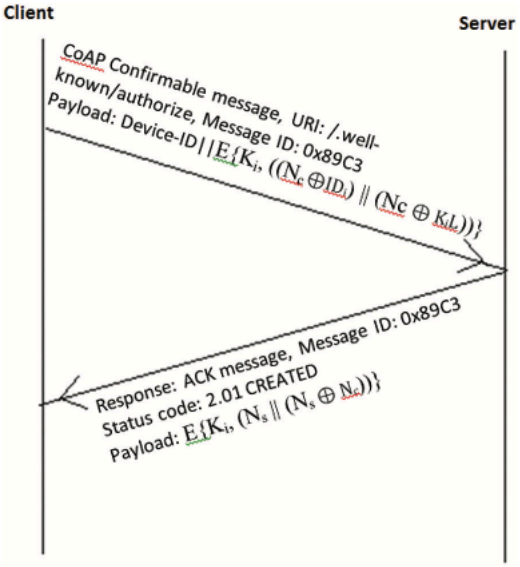

Abbildung 10: CoAP Payload Based Lightweight Authentication (6)

So funktioniert es im Detail: Der Authentifizierungsprozess besteht aus drei Schritten: einmalige Gerätekonfiguration, Client-Authentifizierung und Server-Authentifizierung.

1. **Die einmalige Gerätekonfiguration:** Während der einmaligen Geräteeinrichtungsphase werden der eindeutige AES-Schlüssel und die eindeutige AES-ID in jedem Gerät fest codiert. Das Servergerät ist mit einer Tabelle konfiguriert, in der die Gerätekennung (ID_{i}) und das gemeinsame Geheimnis (K_{i}) jedes Geräts gespeichert sind. Diese Tabelle wird verwendet, um das Gerät zu authentifizieren und den AES-Schlüssel während der ersten Servereinrichtung zu erhalten.

2. **Die Client-Authentifizierung:** Während des Client-Authentifizierungsschritts überprüft und authentifiziert der Server die Identität des Client-Geräts. Der Client generiert eine 64-Bit-Nonce (N_{c}) und XOR die Geräte-ID (ID_{i}). Nonce wird außerdem mit den niedrigstwertigen 64 Bits des gemeinsamen Clientgeheimnisses K_{iL} XOR-verknüpft.

   $C_1 = \mathrm{E}\{\ K_{\mathrm{i}}\ ,\ ((N_{\mathrm{c}}\ \mathrm{XOR}\ ID_{\mathrm{i}})\ ||\ (N_{\mathrm{c}}\ \mathrm{XOR}\ K_{\mathrm{iL}}\ ))\}$

   $\mathrm{M1p} = ID_{\mathrm{i}}\ ||\ C_1$

Der Client sendet dann eine CoAP-Nachricht (M_{1p}) an den Server.Die payload der Nachricht enthält die Geräte-ID und einen verschlüsselten Teil (C_1), der mit einem gemeinsamen Geheimnis generiert wird. Die Größe der Payload-Nachricht beträgt 192 Bit, einschließlich der unverschlüsselten Gerätekennung. Der Server empfängt diese Nachricht und liest daraus die Geräte-ID. Wenn die Geräte-ID in der Tabelle gefunden wird, wird der entsprechende Pre-Shared Key abgerufen. Anschließend entschlüsselt der Server den verschlüsselten Teil (C_1) mit dem gemeinsamen geheimen Schlüssel (K_i), um den Klartext (P_1) zu erhalten. P_1 =D$\{K_i, (C_1)\}$

Die höchstwertigen 64 Bits von P_1 werden mit der Gerätekennung IDi exklusiv-OR-verknüpft, was einen Wert N_c ergibt. Die niederwertigsten 64 Bits von P_1 werden mit diesem N_c exklusiv-OR-verknüpft, was einen WertR_1 ergibt.

Der Server vergleicht den erhaltenen Wert R_1 mit den niederwertigsten 64 Bits des geheimen Schlüssels, der mit der Geräteidentität IDi verknüpft ist. Wenn sie übereinstimmen, wird der Client authentifiziert und eine Sitzung zwischen Client und Server aufgebaut. Stimmen die Werte nicht überein, wird eine unautorisierte Nachricht an den Client gesendet, und es wird keine Sitzung aufgebaut.

3. **Die Server-Authentifizierung:** In der Server-Authentifizierungsphase erstellt der Server nach der Authentifizierung des Clients eine Antwortnachricht (M_{2p}) mit dem Statuscode "2.01 created". Der Server generiert eine Nonce (N_s) und führt eine exklusive OR-Operation mit der Nonce des Clients (N_c) durch.

   M_{2p} = E $\{K_i, (N_s \,||\, (N_s \text{ XOR } N_c))\}$

   Die daraus resultierende Nonce N_s wird dann mit der Verkettung der Nonce des Clients und der Nonce des Servers zu einem 128-Bit-Wert verkettet. Dieser Wert wird mit dem AES-Schlüssel verschlüsselt und als Payload der Antwortnachricht festgelegt.

   Der Client empfängt die Antwortnachricht und entschlüsselt sie mit dem gemeinsamen geheimen Schlüssel (K_i).

   P_2 = D$\{K_i, M_{2p}\}$

   Die entschlüsselte Payload (M_{2p}) wird dann verarbeitet, und die niedrigstwertigen 64 Bits des Klartextes (P_2) werden mit den höchstwertigen 64 Bits von P_2 exklusiv-OR-verknüpft, was zu einem Wert R_2 führt.

   Der Client vergleicht den Wert R_2 mit seiner eigenen Nonce (N_c), um den Server zu authentifizieren. Wenn sie übereinstimmen, ist der Server authentifiziert, was bedeutet, dass nur der beabsichtigte Server die Nachricht des Clients entschlüsselt und die verschlüsselte Nonce zurückgeschickt haben kann.

Sobald die gegenseitige Authentifizierung zwischen Client und Server erreicht ist, wird ein Sitzungsschlüssel (K_s) durch Verkettung der Nonce des Clients (N_c) mit der Nonce des Servers (N_s) erzeugt. Dieser Sitzungsschlüssel wird zum Ver- und Entschlüsseln der während der authentifizierten Sitzung übertragenen Daten verwendet. Der während der Herstellungsphase gemeinsam genutzte AES-Schlüssel wird ausschließlich zu Authentifizierungszwecken verwendet, was die Sicherheit des Systems erhöht.

3.3 Zugriffskontrolle und Berechtigungen

Die Integration der physischen Welt und des Cybersystems im Internet der Dinge stellt große Herausforderungen an die Gestaltung von Sicherheitslösungen. Die Zugangskontrolle gilt als kritische Systemkomponente zum Schutz von Daten und Cyber-Infrastruktur. Aufgrund der neuen Eigenschaften von IoT-Systemen, wie z. B. viele traditionelle Sicherheitslösungen, einschließlich vorhandener Zugriffskontrollmechanismen, sind jedoch möglicherweise nicht direkt in der IoT-Umgebung anwendbar, z. B. Ressourcenbeschränkung, großer Umfang und Geräteheterogenität.(29)(30)

3.3.1 Zugriffskontrollmechanismen

Ein vollständiges Zugangskontrollsystem besteht aus drei Funktionen: Authentifizierung, Autorisierung und Verantwortlichkeit. Der Zugriffskontrollmechanismus, der Hardware und Software umfasst, setzt Richtlinien durch und wertet Zugriffsanfragen aus. Die Konfiguration der Zugriffskontrollrichtlinien ist von entscheidender Bedeutung und sollte regelmäßig überprüft und verifiziert werden. Tools wie ACLs, Router, Verschlüsselung, Prüfprotokolle, IDS, Antivirensoftware, Firewalls, Smartcards und Warnungen sind Teil dieses Systems(30),(31),(32),(33),(34). Es gibt viele verschiedene Zugriffskontrollmechanismen wie Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC) , Mandatory Access Control (MAC), Discretionary Access Control (DAC), Multi-Factor Authentication (MFA) und Single Sign-On (SSO) und Secure Shell (SSH), aber wir konzentrieren uns nur auf RBAC und ABAC(30), (35), (36)(37)(38).

3.3.2 RBAC und ABAC

Obwohl ABAC und RBAC ähnlich sind, haben sie Vor- und Nachteile. Bei kluger Kombination kann die Kombination eine skalierbare, flexible, überprüfbare und verständliche Zugangskontrolle bieten. Durch die Kombination von Rollenzentrierung, dynamischen Rollenfunktionen und der feinkörnigen Autorisierung von ABAC demonstriert es die Praktikabilität des in ANSI/INCITS 494-2012 definierten Ansatzes und bietet eine Kombination der besten Funktionen von RBAC und ABAC für Unternehmen. RBAC ist weit verbreitet und bietet Verwaltungs- und Sicherheitsvorteile. Es ist jedoch veraltet, teuer in der Implementierung und nicht in der Lage, Echtzeit-Umgebungszustände als Zugangskontrollparameter zu berücksichtigen. ABAC hingegen ist neuer, einfacher zu implementieren und kann Echtzeit-Umgebungszustände als Zugangskontrollparameter berücksichtigen.

Einerseits basiert die Zugriffskontrolle in RBAC auf vordefinierten Rollen und Berechtigungen, und Benutzerzugriffsentscheidungen hängen in erster Linie von den ihnen zugewiesenen Rollen ab. RBAC vereinfacht die Verwaltung, indem es die Zugriffskontrolle nach Rollen organisiert. Andererseits basiert die Zugriffskontrolle in ABAC auf verschiedenen Merkmalen von Benutzern, Objekten und der Umgebung, und Zugriffsentscheidungen berücksichtigen Attribute wie Benutzerrolle, Abteilung, Standort usw. ABAC bietet eine feinkörnige und kontextbezogene Steuerung von Genehmigung.

Sowohl RBAC als auch ABAC können verwendet werden, indem Rollen als Benutzerattribute betrachtet werden. RBAC vereinfacht die Administration und eignet sich für Organisationen mit komplexen Benutzerstrukturen. ABAC bietet Flexibilität, Granularität und feinkörnige Zugriffskontrolle. Darüber hinaus können Attribute zu RBAC hinzugefügt werden, um die erforderliche Flexibilität bei der Zugriffskontrolle zu erreichen.

RBAC ist einfacher zu implementieren, während ABAC eine Eigenschafts- und Richtlinienverwaltung erfordert. ABAC eignet sich besonders für dynamische Umgebungen und Attribute wie Tageszeit und Standort und ermöglicht eine effektive Richtliniendurchsetzung(35)(36).

ABAC Eigenschaften	Beschreibung
Rollenzentrierung + Autorisierung	Zugriff wird basierend auf Benutzerattributen autorisiert, die in Rollen organisiert sind.
Echtzeit-Umgebungszustände	Aktuelle Umgebungsbedingungen werden in Echtzeit berücksichtigt.
Flexibilität, Granularität und Kontextbezogene Steuerung	Feinkörnige Steuerung von Zugriffsrechten basierend auf vielfältigen Attributen und Kontextfaktoren.
Erfordert Eigenschafts- und Richtlinienverwaltung	Die Verwaltung von Attributen und Richtlinien ist notwendig, um ABAC zu implementieren.
Für dynamische Umgebungen geeignet	Gut geeignet für dynamische und komplexe Umgebungen, in denen sich Zugriffsrechte ändern können.
Einfacher zu implementieren	Die Implementierung und Verwaltung von ABAC kann vergleichsweise einfacher sein.
Rollen als Benutzerattribute	Rollen dienen als eine der Attribute, die den Zugriff beeinflussen.
Flexibilität durch Attribute	Flexibilität wird durch die Vielfalt der verwendeten Attribute erreicht

Abbildung 11:

RBAC Eigenschaften	Beschreibung
Verwaltungs- und Sicherheitsvorteile	RBAC bietet Vorteile in Bezug auf Verwaltung und Sicherheit in großen Organisationen.
Basierend auf vordefinierten Rollen und Berechtigungen	Zugriffsrechte werden basierend auf vordefinierten Rollen und Berechtigungen vergeben.
Vereinfachte Administration	Die Verwaltung von Zugriffsrechten wird durch die Verwendung von Rollen vereinfacht.
Einfacher in der Implementierung	RBAC kann vergleichsweise leichter implementiert werden.
Rollen als Benutzerattribute	Rollen sind eine wichtige Grundlage für die Zuweisung von Zugriffsrechten.
Flexibilität durch Attribute	Flexibilität wird durch die Verwendung von Attributen in Verbindung mit Rollen erreicht.

Abbildung 12:

3.3.3 HyBACAC

Durch die vorteilhafte Kombination zweier Modelle, ABAC und R, als Hybridansatz steht eine umfassendere und wandelbarere Zutrittskontrolllösung zur Verfügung. Ein von RBAC bereitgestelltes rollenzentriertes Framework vereinfacht die Administration und Zugriffsverwaltung, indem Berechtigungen an Rollen statt an einzelne Benutzer vergeben werden. Dies verbesserten die Skalierbarkeit und Ausdauer. Andererseits kann ABAC eine feinkörnige Zugriffskontrolle unter Berücksichtigung der Attribute von Benutzern, Geräten und der Umgebung durchführen, was eine dynamische Entscheidungsfindung auf der Grundlage verschiedener Kontextfaktoren ermöglicht. HyBACAC kann von der rollenbasierten Einfachheit von RBAC und der attributbasierten Flexibilität von ABAC profitieren. Diese Kombination erleichtert Administratoren das Definieren von Rollen und das Zuweisen geeigneter Berechtigungen, indem dynamische Attribute berücksichtigt werden, um den Zugriff weiter einzuschränken. Es ermöglicht das Prinzip der geringsten Rechte und eine einfache Überprüfung der Benutzerberechtigungen. Das Hybridmodell ist in der Lage, plötzliche Attributänderungen und Kontextinformationen durch dynamische Attribute und Umgebungsrollen zu erfassen. Dadurch wird sichergestellt, dass Entscheidungen zur Zugangskontrolle, an die sich ändernden Bedingungen in intelligenten IoT-Systemen angepasst werden können(36).

HyBACAC Eigenschaften	Beschreibung
Verbesserte Skalierbarkeit und Ausdauer	HyBACAC bietet bessere Skalierbarkeit und Ausdauer in komplexen Umgebungen.
Dynamische Attribute und Umgebungsrollen für Zugriffsbeschränkungen	Zugriffsbeschränkungen werden basierend auf dynamischen Attributen und Umgebungsrollen festgelegt.
Anpassung an changing Bedingungen in intelligenten IoT-Systemen	Gut geeignet, um sich an sich ändernde Bedingungen in intelligenten IoT-Systemen anzupassen.
Vereinfacht Administration und Zugriffsverwaltung	HyBACAC erleichtert die Verwaltung von Zugriffsrechten und Administration.
Ermöglichung der Feinkörnige Zugriffskontrolle	HyBACAC ermöglicht eine detaillierte Kontrolle über Zugriffsrechte.
Dynamische Entscheidungsfindung basierend auf Kontextfaktoren	Zugriffsentscheidungen werden dynamisch basierend auf Kontextfaktoren getroffen.
Kombiniert Rollen- und attributbasierte Zugriffskontrolle	HyBACAC kombiniert sowohl Rollen- als auch attributbasierte Zugriffskontrollmechanismen.
Komplexität der Implementation	Die Implementierung von HyBACAC kann komplex sein.
Kosten viel	Die Kosten für die Implementierung und Verwaltung von HyBACAC können höher sein.
Abhängigkeit von Workload und Anwendungen	Die Leistung von HyBACAC kann von der Workload und spezifischen Anwendungen abhängen.

Abbildung 13:

4 Zusammenfassung

In diesem Artikel haben wir uns mit den grundlegenden Konzepten des Internet der Dinge (IoT) befasst, insbesondere im Zusammenhang mit Smart-Home Netzwerken. Dabei wurden Risiken und Bedrohungen beleuchtet, wie zum Beispiel absichtliche Cyberangriffe, unbeabsichtigte Fehler und Fehlfunktionen sowie Malware-Infektionen und Denial-of-Service-Angriffe. Um diese Netzwerke zu schützen, sind entsprechende Schutzmechanismen von großer Bedeutung und müssen deswegen eingesetzt werden. Verschiedene Verschlüsselungstechnologien wie AES, RSA und BLOWFISH wurden diskutiert, wobei sich AES als effizienter und sicherer als andere erwiesen hat.Zudem haben wir eine Reihe nützlicher Authentifizierungsmethoden wie Mutual TLS, Lightweight CoAP-based Authentication und CoAP Payload Based Lightweight

Authentication vorgestellt. Lightweight CoAP-based Authentication und CoAP Payload Based Authentication sind ideal für eingeschränkte Geräte. CoAP Payload Based Lightweight Authentication ist jedoch etwas schneller und erfordert nur zwei Handshakes für eine vollständige Authentifizierung. Außerdem ist eine effektive Zugangskontrolle durch ABAC, RBAC und insbesondere HybACAC erforderlich, um unbefugten Zugriff zu verhindern.

Keywords: Smart Home Netzwerke, Architektur, Kommunikationsprotokolle, Bedrohungen, Risiken, Schutzmechanismen, Sicherheit, Verschlüsselung, Authentifizierung, Zugriffskontrolle, AES, BLOWFISH, RSA, Mutual TLS, Lightweight CoAP-based Authentication, CoAP Payload Based Lightweight Authentication, Zugriffskontrollmechanismen, IoT, Netzwerksicherheit, Datenschutz, Netzwerkarchitektur, Schwachstellen, Sicherheitslücken, Netzwerkprotokolle, Cybersecurity, Sichere Kommunikation, Risikomanagement.

Was? Wer?

Architektur von Smart Home-Netzwerken: Aiman

Kommunikationsprotokolle in Smart Home-Netzwerken: Aiman

Bedrohungen und Risiken für Smart Home-Netzwerke: Zohreh

Wichtige Schutzmechanismen zur Sicherung von Smart Home-Netzwerken: Aiman

Verschlüsselungstechnologien: Zohreh

Authentifizierung: Aiman

Zugriffskontrolle und Berechtigungen: Zohreh

Literatur

[1] P. Datta and B. Sharma, "A survey on iot architectures, protocols, security and smart city based applications," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. Los Alamitos, CA, USA: IEEE Computer Society, jul 2017, pp. 1–5. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCCNT.2017.8203943

[2] O. Djumanazarov, A. Väänänen, K. Haataja, and P. Toivanen, "An overview of iot-based architecture model for smart home systems," in *International Conference on Intelligent Systems Design and Applications*. Springer, 2021, pp. 696–706.

[3] R. Khatoun, *Cybersecurity in Smart Homes: Architectures, Solutions and Technologies*. John Wiley & Sons, 2022.

[4] H. Lin and N. W. Bergmann, "Iot privacy and security challenges for smart home environments," *Information*, vol. 7, no. 3, 2016. [Online]. Available: https://www.mdpi.com/2078-2489/7/3/44

[5] A. Satapathy, J. Livingston *et al.*, "A comprehensive survey on ssl/tls and their vulnerabilities," *International Journal of Computer Applications*, vol. 153, no. 5, pp. 31–38, 2016.

[6] S. G. Oliver and T. Purusothaman, "Lightweight and secure mutual authentication scheme for iot devices using coap protocol." *Computer Systems Science & Engineering*, vol. 41, no. 2, 2022.

[7] A. B. Brush, B. Lee, R. Mahajan, S. Agarwal, S. Saroiu, and C. Dixon, "Home automation in the wild: Challenges and opportunities," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 2115–2124. [Online]. Available: https://doi.org/10.1145/1978942.1979249

[8] M. N. Anwar, M. Nazir, and K. Mustafa, "Security threats taxonomy: Smart-home perspective," in *2017 3rd International Conference on Advances in Computing,Communication Automation (ICACCA) (Fall)*, 2017, pp. 1–4.

[9] A. Jacobsson, M. Boldt, and B. Carlsson, "A risk analysis of a smart home automation system," *Future Generation Computer Systems*, vol. 56, pp. 719–733, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X15002812

[10] T. Xu, J. B. Wendt, and M. Potkonjak, "Security of iot systems: Design challenges and opportunities," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014, pp. 417–423.

[11] R. Stephen and L. Arockiam, "Intrusion detection system to detect sinkhole attack on rpl protocol in internet of things," *International Journal of Electrical Electronics and Computer Science*, vol. 4, no. 4, pp. 16–20, 2017.

[12] A. Mathur, T. Newe, and M. Rao, "Defence against black hole and selective forwarding attacks for medical wsns in the iot," *Sensors*, vol. 16, no. 1, 2016. [Online]. Available: https://www.mdpi.com/1424-8220/16/1/118

[13] S. Pawar and P. Vanwari, "Sybil attack in internet of things," *International Journal of Engineering and Innovative Technology (IJESIT)*, vol. 5, no. 4, pp. 96–105, 2016.

[14] I. Alqassem, "Privacy and security requirements framework for the internet of things (iot)," in *Companion Proceedings of the 36th International Conference on Software Engineering*, 2014, pp. 739–741.

[15] D. Kozlov, J. Veijalainen, and Y. Ali, "Security and privacy threats in iot architectures." in *BODYNETS*, 2012, pp. 256–262.

[16] S. Smys, A. Basar, H. Wang *et al.*, "Hybrid intrusion detection system for internet of things (iot)," *Journal of ISMAC*, vol. 2, no. 04, pp. 190–199, 2020.

[17] M. Saadeh, A. Sleit, M. Qatawneh, and W. Almobaideen, "Authentication techniques for the internet of things: A survey," in *2016 cybersecurity and cyberforensics conference (CCC)*. IEEE, 2016, pp. 28–34.

[18] S. Singh, I.-H. Ra, W. Meng, M. Kaur, and G. H. Cho, "Sh-blockcc: A secure and efficient internet of things smart home architecture based on cloud computing and blockchain technology," *International Journal of Distributed Sensor Networks*, vol. 15, no. 4, p. 1550147719844159, 2019.

[19] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*. IEEE, 2017, pp. 618–623.

[20] M. F. Elrawy, A. I. Awad, and H. F. Hamed, "Intrusion detection systems for iot-based smart environments: a survey," *Journal of Cloud Computing*, vol. 7, no. 1, pp. 1–20, 2018.

[21] E. Thambiraja, G. Ramesh, and D. R. Umarani, "A survey on various most common encryption techniques," *International journal of advanced research in computer science and software engineering*, vol. 2, no. 7, 2012.

[22] A. M. Abdullah *et al.*, "Advanced encryption standard (aes) algorithm to encrypt and decrypt data," *Cryptography and Network Security*, vol. 16, no. 1, p. 11, 2017.

[23] P. Raghav, R. Kumar, and R. Parashar, "Securing data in cloud using aes algorithm," *International Journal of Engineering Science and Computing*, pp. 3672–3675, 2016.

[24] A. Devi, A. Sharma, and A. Rangra, "A review on des, aes and blowfish for image encryption & decryption," *International Journal of Computer Science and Information Technologies*, vol. 6, no. 3, pp. 3034–3036, 2015.

[25] E. Milanov, "The rsa algorithm," *RSA laboratories*, pp. 1–11, 2009.

[26] M. El-Hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of internet of things (iot) authentication schemes," *Sensors*, vol. 19, no. 5, p. 1141, 2019.

[27] M. Barenkamp, "Iot security best practices," *HMD Praxis der Wirtschaftsinformatik*, vol. 58, no. 2, pp. 400–424, 2021. [Online]. Available: https://link.springer.com/article/10.1365/s40702-020-00637-4

[28] M. A. Jan, P. Nanda, X. He, Z. Tan, and R. P. Liu, "A robust authentication scheme for observing resources in the internet of things environment," in *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 2014, pp. 205–211.

[29] M. Alramadhan and K. Sha, "An overview of access control mechanisms for internet of things," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, 2017, pp. 1–6.

[30] A. Ouaddah, H. Mousannif, A. Abou Elkalam, and A. A. Ouahman, "Access control in the internet of things: Big challenges and new opportunities," *Computer Networks*, vol. 112, pp. 237–262, 2017.

[31] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. d. Bruijn, C. d. Laat, M. Holdrege, and D. Spence, "Rfc2904: Aaa authorization framework," 2000.

[32] D. Boyle and T. Newe, "A survey of authentication mechanisms: Authentication for ad-hoc wireless sensor networks," in *2007 IEEE Sensors Applications Symposium*. IEEE, 2007, pp. 1–6.

[33] H. Hu, G.-J. Ahn, and K. Kulkarni, "Anomaly discovery and resolution in web access control policies," in *Proceedings of the 16th ACM symposium on Access control models and technologies*, 2011, pp. 165–174.

[34] Y. Ledru, N. Qamar, A. Idani, J.-L. Richier, and M.-A. Labiadh, "Validation of security policies by the animation of z specifications," in *Proceedings of the 16th ACM symposium on Access control models and technologies*, 2011, pp. 155–164.

[35] E. Coyne and T. R. Weil, "Abac and rbac: scalable, flexible, and auditable access management," *IT professional*, vol. 15, no. 03, pp. 14–16, 2013.

[36] S. Ameer, J. Benson *et al.*, "Hybrid approaches (abac and rbac) toward secure access control in smart home iot," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[37] I. Ali, S. Sabir, and Z. Ullah, "Internet of things security, device authentication and access control: a review," *arXiv preprint arXiv:1901.07309*, 2019.

[38] S. Dutta, S. S. L. Chukkapalli, M. Sulgekar, S. Krithivasan, P. K. Das, and A. Joshi, "Context sensitive access control in smart home environments," in *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. IEEE, 2020, pp. 35–41.