Absicherung des Internets der Dinge

Artemij Olegovic Voskobojnikov
Freie Universität Berlin
Fakultät für Mathematik und Informatik
Berlin, Germany
Email: voskobojnikov.artemij@gmail.com

Benjamin Swiers
Freie Universität Berlin
Fakultät für Mathematik und Informatik
Berlin, Germany
Email: swiers.benjamin@googlemail.com

***Zusammenfassung*—Aktuellen Prognosen zufolge werden im Jahre 2020 rund 50 Milliarden "Dinge"mit dem Internet verbunden sein. Die Verknüpfung der virtuellen mit der dinglichen Welt stellt die Entwickler derartiger Systeme zukünftig vor diverse Herausforderungen, welche insbesondere die sichere technische Umsetzung von Datenübertragung und Datenspeicherung beinhaltet. In dieser Ausarbeitung wird das Konzept des *Internet of Things* (*IoT*, z.Dt. *Internet der Dinge*) erläutert und auf potentielle Risiken eingegangen, welche eine derart vernetze Welt mit sich bringt. Weiterhin werden Lösungsansätze für Sicherheitsmechanismen, sowohl im privaten als auch im industriellen Bereich, genannt. Juli, 2015**

I. Einleitung

Die Gegenwart von Computern hat Einzug in unseren Alltag gehalten. Während noch vor einigen Jahren ein stationärer Computer für das gelegentliche Surfen im Internet ausreichte, begleiten uns heute Smartphones, Tablets uns Smart Watches auf all unseren Wegen. Das Mitführen solcher Geräte erfüllt dabei nicht nur den Zweck, jederzeit erreichbar zu sein, sondern ermöglicht dem Nutzer weiterhin jederzeit auf das Internet zuzugreifen. Durch diese permanente Anbindung an das Internet entstehen eine Vielzahl von Gefahren, welchen jeder Nutzer potentiell ausgesetzt ist.
Unsere täglichen Begleiter beinhalten unzählige vertrauliche Daten, seien es Medien, Kontaktdaten oder die Inhalte eines Terminkalenders. All diese Informationen befinden sich oft nicht nur auf den Geräten selbst, sondern mitunter auf Cloud-Systemen, wie iCloud, Dropbox oder Google+. Dafür gibt es mehrere Gründe. Zum einen verfügen diese Geräte hinsichtlich Speicherkapazität, Rechenleistung und Energie über begrenzte Ressourcen. Darüber hinaus ermöglicht die Auslagerung der Daten auf die genannten Systeme, diese Daten mit anderen Geräten zu synchronisieren.
Betrachtet man hinsichtlich der zur Verfügung stehenden Ressourcen die Leistungsfähigkeit von kleinen Chips, wie sie in Bereichen der Hausautomation und der Industrie verwendet werden, so wird klar, dass diese oft über weitaus weniger Speicherkapazität, Rechenleistung und Energie verfügen.
Um ein möglichst hohes Maß an Sicherheit für *Iot*-Geräten zu gewährleisten, muss also ein Weg gefunden werden, der die permanente Verfügbarkeit von Information und die sichere Übertragung dieser Information auf Geräten mit begrenzten Ressourcen ermöglicht. Im Verlauf dieser Ausarbeitung werden Lösungsansätze aufgezeigt, welche zur Umsetzung dieser Anforderung im *Internet der Dinge* beitragen.

II. Das Internet der Dinge

Bevor auf Lösungen zur Optimierung der Sicherheit im *IoT* eingegangen wird, soll zunächst geklärt werden, was das *Internet der Dinge* genau ist und welche Eigenschaften sich hinter diesem Begriff verbergen. In [1] wird das *Internet der Dinge* wie folgt beschrieben: Das *Internet der Dinge* verfolgt "[...] das Ziel einer Unterstützung des Menschen sowie einer durchgängigen Optimierung wirtschaftlicher und sozialer Prozesse durch eine Vielzahl von in die Umgebung eingebrachten Mikroprozessoren und Sensoren."Weiterhin ergeben sich daraus laut [1] folgende Eigenschaften:

- *Dezentralität bzw. Modularität*:
  IoT-Geräte sind modular aufgebaut und lassen sich miteinander kombinieren, sodass sie miteinander kommunizieren und interagieren.

- *Einbettung*:
  Da die verwendeten Geräte zunehmend kleiner und portabler werden, können diese in Gegenstände des Alltags integriert werden, wodurch ein *IoT*-Gerät mitunter nicht mehr als solches Erkennbar ist (z.B. ein RFID-Chip in der Kleidung).

- *Mobilität*:
  IoT-Geräte müssen in der Lage sein, sich den Umgebungen des Nutzers anzupassen, um zu jeder Zeit an jedem Ort den Funktionsumfang zu gewährleisten, für den das entsprechende Gerät konzipiert wurde.

- *(Spontane) Vernetzung*:
  Ein wesentliches Merkmal des Internets der Dinge ist die Tatsache, dass die Geräte über das Internet oder anderen Netzwerktechnologien miteinander verbunden sind. Dabei ist es denkbar, dass die Geräte, je nach Einsatzgebiet, spontan (ad hoc) eine Verbinden zueinander aufnehmen können. Ein Beispielszenario wäre die in [2] beschrieben Kommunikation von Autos, die sich auf einer Straße entgegenkommen, um den Verkehrsstatus der kommenden Wegstrecke untereinander auszutauschen oder auf mögliche Gefahren hinzuweisen.

- *Kontextsensitivität*:
  IoT-Systeme sammeln Informationen ihrer Umgebung

und passen die angebotenen Dienste an die Bedürfnisse des Nutzersan den jeweiligen Kontext an.

- *Autonomie*:
  Das System reagiert eigenständig (ohne den Zugriff des Nutzers) auf bestimmte Ereignisse.

- *Energieautarkie*:
  Da sich das Einsatzgebiet eines *IoT*-Gerätes auf viele verschiedene Bereiche erstreckt, muss dieses über eine eigene Energiequelle verfügen und sollte nicht auf die stationäre Versorgung angewiesen sein.

Die Einsatzgebiete von *IoT*-Geräten Geräte sind vielfältig. Sie können in eine Infrastruktur (Gegenstände, Gebäude) und in andere mobile Geräte (Mobiltelefone, Wearables, Accessoires) eingebettet oder sogar dem Nutzer selbst implantiert werden (computerisierte Implantate)[1].

III. Sicherheit im IoT

A. Sicherheit von existierenden IoT-Geräten

Mit der wachsenden Zahl der *IoT*-Geräte steigt ebenfalls die Angriffsfläche für potentielle Angreifer. Da das *Internet der Dinge* noch vergleichsweise jung ist, sind viele Geräte, die bereits auf dem Markt erhältlich sind noch nicht bis ins Detail ausgereift. Viele Techniken und Methoden, die sich in der Informationstechnik bereits seit langem bewährt haben, können auf Grund der beschränkten Ressourcen in *IoT*-Geräten nicht verwendet werden. Einen erschreckenden Einblick in den Sicherheitsstand von im Handel erhältliche Geräten gewährt [3].
In dem Report wurden 10 führende *IoT*-Geräte, darunter Fernseher, Webcams, Thermostate, fernsteuerbare Steckdosen, Löschanlagen, Türschlösser, Waagen und Garagenöffner auf sicherheitsrelevante Schwachstellen untersucht. Bei den 10 Geräten wurden insgesamt 250 Schwachstellen gefunden. Die Abbildung 1 zeigt die prozentuale Verteilung verschiedener Sicherheitslücken in den Geräten.

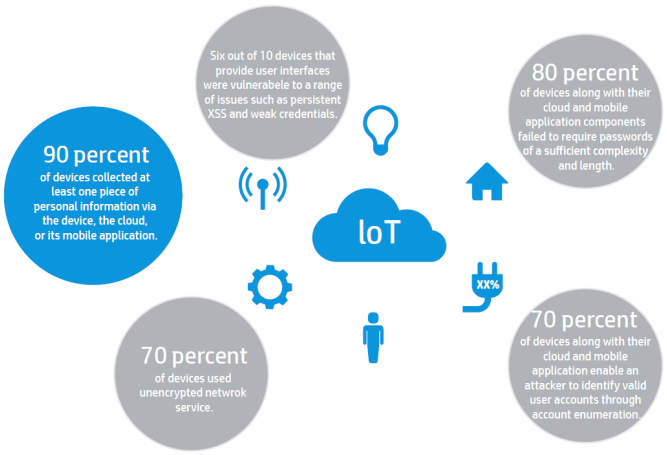

Abbildung 1: Reseach findings

B. Hauptgründe für Sicherheitsprobleme im IoT

Betrachtet man die Ergebnisse aus [3], so lassen sich einige Probleme kategorisieren und zu bestimmten Problemklassen zusammenführen. In [4] werden folgende 10 Hauptgründe für Sicherheitsprobleme im *IoT* genannt:

a *Unsichere Web Interfaces*
b *Unzureichende Authentifikation*
c *Unsichere Netzwerk Services*
d *Mangelnde Transportverschlüsselung*
e *Datenschutz/ Privatsphäre*
f *Unsichere Cloud-Schnittstellen*
g *Unsichere Mobile-Interfaces*
h *Unzureichendes Maß an Sicherheitskonfiguration*
i *Unsichere Software/ Firmware*
j *Mangelnde physische Sicherheit*

Im Folgenden werden diese Probleme erläutert, mögliche Angriffe aufgezeigt und Lösungen zur Reduzierung der Sicherheitsprobleme genannt.

1) Unsichere Benutzer-Schnittstellen: Die Probleme *a*, *f* und *g* betreffen die User-Interfaces, welche die Schnittstelle zwischen einem Nutzer und der Anwendung darstellen. Wenn Benutzer-Schnittstellen nicht ausreichend sicher implementiert werden, so sind Angriffe, wie *Account Enumeration* (Herausfinden, ob ein bestimmter Benutzername existiert), *Cross-site Scripting, XSS* (Ausführen von bösartigem Code in vertrauenswürdigem Context) oder *SQL-Injections* (Eingabe bösartiger Nutzereingaben, welche die Manipulation der Daten in der Datenbank zur Folge haben) möglich.
Weiterhin birgt die Verwendung von schwachen Anmeldedaten die Gefahr, dass ein Angreifer, mit einem zu geringen Aufwand Zugriff auf ein Benutzerkonto bekommt, für das dieser eigentlich nicht autorisiert ist.
Um diesen Angriffen zuvorzukommen sollten Entwickler die Eingaben der Schnittstellen validieren und einen robusten Passwort-Recovery-Mechanismus verwendet, durch den ein potentieller Angreifer keinen Hinweis auf einen gültigen Account bekommt[4]. Wenn ein Benutzer (eventuell Angreifer) zu oft falsche Anmeldedaten eingibt, so sollte dies eine Sperrung des Benutzerkontos zur Folge haben. Darüber hinaus dürfen die Anmeldedaten nicht offen dargelegt werden (Netzwerkverkehr).

2) Unzureichende Authentifikation: Ein wesentlicher Bestandteil für den sicheren Zugang zu einem System stellt der verwendete Authentifikationsmechanismus dar. Eine unzureichende Passwortkomplexität ermöglicht Angreifern den Zugang zum System, wodurch er dieses direkt manipulieren oder Zugang zu schutzbedürftigen Daten erlangen kann.
Durch die Verwendung von entsprechend komplexen Anmeldedaten und die Verschlüsselung dieser Daten kann ein weitaus höheres Maß an Sicherheit gewährleistet werden. Wenn möglich, sollte eine *Zwei-Faktor-Zwei-Faktor-*

Authentifizierung oder eine *Public Key Infrastructur (PKI)* genutzt werden[4].

3) Unsichere Netzwerk Services: Da die Bestandteile einer *IoT*-Infrastruktur alle miteinander vernetzt sind, bedeutet dies, dass ständig Daten miteinander ausgetauscht werden. Der Zugang zum System beinhaltet ein hohes Gefahrenpotential, denn gelingt es einem Angreifer sich über die Netzwerkservices einen Zugang zum System zu verschaffen, so kann dieser die übermittelten Daten abfangen, das System manipulieren oder funktionsunfähig zu machen (z.B. durch einen *Denial-of-Service*-Angriff). Die Schwachstelle bilden hier unnötig offene Ports oder angreifbare Services.
Um diesen Gefahren vorzubeugen, sollten die zu übermittelten Daten stets verschlüsselt übertragen werden. Weiterhin sollten nur diejenigen Ports erreichbar sein, die für die Aufrechterhaltung der Funktionsweise des Systems notwendig sind. Bestimmte Ports und Services sollten nicht über das Internet, z.B. mittels *Universal Plug and Play (UPnP)* erreichbar sein[4]. Die Herstellern und Entwicklern müssen in diesem Punkt einen Kompromiss finden zwischen Erreichbarkeit und Angreifbarkeit.

4) Mangelnde Transportverschlüsselung: Auch die Verwendung einer mangelnden Transportverschlüsselung führt potentiell zur Verletzung aller Schutzziele (Vertraulichkeit, Integrität, Verfügbarkeit). Ein wesentlicher Aspekt bildet hier die Nutzung geeigneter Protokolle(*Lightweight Kryptography*), welche die verschlüsselte Übertragung der Daten mit begrenzten Ressourcen ermöglichen. Beispiele für mögliche Protokolle und Verfahren sind das *Constrained Application Protocol (CoAP)* und *ZigBee*, aber auch Standards wie *Bluetooth* oder *Near Field Communication (NFC)*, welche durch ihre geringe Reichweite ein gewisses Maß an Sicherheit gewährleisten[5][6].

5) Datenschutz/ Privatsphäre: Test

6) Unzureichendes Maß an Sicherheitskonfiguration: Test

7) Unsichere Software/ Firmware: Test

8) Mangelnde physische Sicherheit:

IV. Anwendung des IoT in der Industrie

In Unternehmen wird meist zwischen zwei Informations- und Kommunikationstechnikbereichen unterschieden. Zum einen existiert in Unternehmen die *Produktions-IT*, welche jegliche Controller von Produktionsanlagen oder Logistikanbindungen umfasst. Zum anderen gibt es ebenfalls den Bereich der *Business-IT*. Darunter fallen jegliche Anwendungen, die von den Angestellten verwendet werden und in keinem direkten Zusammenhang mit der Produktion stehen. Ein Beispiel dafür wären *Content-Management-Systeme*.

V. Smart Home

Der Einsatz von *IoT*-Geräten zur Vernetzung verschiedener Komponenten in Wohngebäuden wird als *Smart Home* bezeichnet[7].

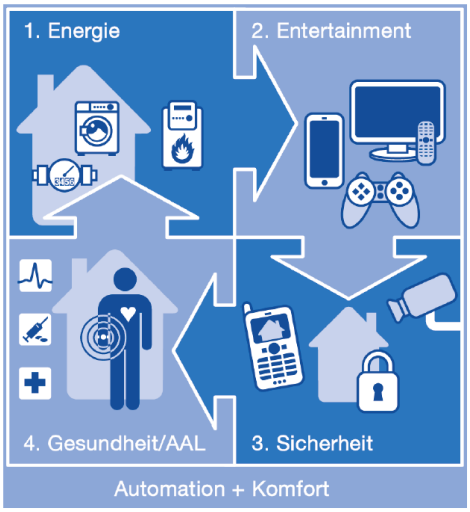

Abbildung 2: Anwendungsbereiche im Smart Home

Die Abbildung 2 zeigt die Anwendungsbereiche für einen vernetzen Haushalt. Um ein möglichst sicheres Umfeld für den Heimgebrauch zu schaffen, sollten ausschließlich geeignete Hardware verwendet werden. Zum einen sind dies Bauteile, die sogenannte *PUFS* (siehe Abschnitt Sichere Komponenten) verwenden und darüber hinaus Bauteile, welche auf äußere Einflüsse reagieren. Diese Bauteile erkennen Gefahren (unberechtigter Fremdzugang, Manipulation, Sabotage) und veranlassen beispielsweise die Trennung vom Internet[8].
Weiterhin sollte die leitungsgebundene Übertragung dem WLAN oder anderen Funk-Übertragungswegen vorzuziehen sein. Dieses Vorgehen schließt den Zugriff eines Angreifers nicht gänzlich aus, jedoch muss ein solcher einen höheren Aufwand betreiben, um Zugang zum System zu erhalten.
Auch im *Smart Home*-Bereich sollte besonderes Augenmerk auf die Verwendung von entsprechend komplexen Passwörtern für den Zugang zum System und die Nutzung von Verschlüsselungstechniken nach dem Stand der Technik (z.B. *AES*) gelegt werden. Weiterhin empfiehlt sich die Benutzung von Softwarekomponenten, welche laufende Angriffe erkennen (*Intrusion Detektion*). Dies kann durch Firewalls mit Application Filtern realisiert werden.

VI. Industrie 4.0

Die *Industrie 4.0* vernetzt beide Bereiche miteinander und als Resultat entstehen sogenannte *cyberphysikalische Systeme* (*CPS*).
Bei einem *CPS* handelt es sich meist um ein Gerät mit beschränkten Ressourcen. Dies bedeutet, dass diese Systeme in der Rechenleistung und im Energieverbrauch beschränkt

sind. Zusätzlich gibt es Anforderungen an solche Systeme, die unbedingt erfüllt werden müssen. So müssen *CPS* ständig verfügbar und ausfallsicher sein, sodass es im schlimmsten Fall nicht zum Produktionsstillstand kommen kann.

A. Anforderungen an derartige Systeme

Aufgrund der Vernetzung beider *IKT-Bereiche*, werden Anforderungen des jeweiligen Subsystems übernommen. Dies bedeutet, dass Sicherheitslösungen im Office-Bereich nicht ohne weiteres angewendet werden können, da in einem *CPS* auch Auswirkungen auf den anderen IKT-Bereich (in diesem Beispiel die *Produktions-IT*) abgeschätzt werden müssen. Aus dieser Problematik resultieren somit neue Probleme und Schwachstellen, die möglicherweise bei den anfänglich separierten Systemen nicht existierten.
Im Detail bedeutet das, dass sich insbesondere die Sicherheitsanforderungen unterscheiden und bekannte Sicherheitslösungen der *Business-IT* wie VPN oder SSL/TLS-Verschlüsselung nicht auf die *Produktions-IT* übertragen werden können.
Der Hauptgrund dafür ist die Tatsache, dass Komponenten der *Produktions-IT* zertifiziert sind und eine Verwendung von Verschlüsselungen einen Eingriff bedeuten würde, welcher im schlimmsten Fall zu Verlust der Zertifizierungen führen würde. Ebenfalls können Verschlüsselungen zu möglichen Latenzen führen, die im Office-Bereich verkraftbar sind, in einer Produktion aber eine Nichtfunktionalität bedeuten würden.
Ein weiterer Aspekt, der in der *Business-IT* nicht bedacht werden musste, sind physikalische Angriffe, die in der Produktion denkbar sind. So können Mitarbeiter Komponenten verändern oder gänzlich entfernen [9].

In den folgenden Abschnitten werden mögliche Probleme der *Industrie 4.0* aufgezeigt und es werden Lösungsansätze demonstriert, die aktuell im Einsatz sind.

B. Cloud-Computing

1) Unterarten von Clouds: Cloud-Computing ist nicht gleich Cloud-Computing. Es gibt verschiedene Services, die auf verschiedene Anwendergruppen zugeschnitten sind, wie die nachfolgende Grafik demonstriert.

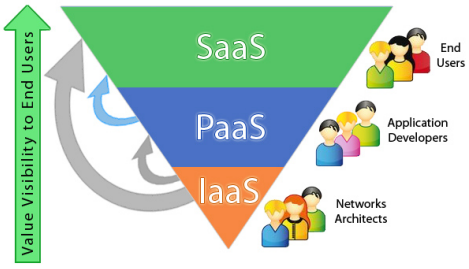

Abbildung 3: Unterarten von Clouds

Die umgekehrte Pyramide verdeutlicht dabei den unterschiedlichen Umfang des jeweiligen Dienstes. Dabei wird zwischen drei verschiedenen Services unterschieden. Bei *Software-as-a-Service* (SaaS) werden vollständige Systeme den Nutzern zur Verfügung gestellt. Der Zugriff auf diese erfolgt mittels Browser.
Product-as-a-Service (PaaS) ist in der Regel auf Anwendungsentwickler bzw. fortgeschrittene Nutzer zugeschnitten, welche möglicherweise eigene Applikationen auf den Systemen verwenden wollen. Die letzte Unterart *Infrastructure-as-a-Service* (IaaS) bietet dem Anwender lediglich ein Grundgerüst [10]. Es können personalisierte Dienste oder Betriebssysteme installiert werden. Große Unternehmen wählen häufig IaaS als Cloud-Lösung, weil diese eine komplette Plattform bieten und ein Drittanbieter für die Wartung dieser zuständig ist. Darüber hinaus ist der Drittanbieter im Falle eines Angriffs auch verantwortlich.

C. Industrie und die Cloud

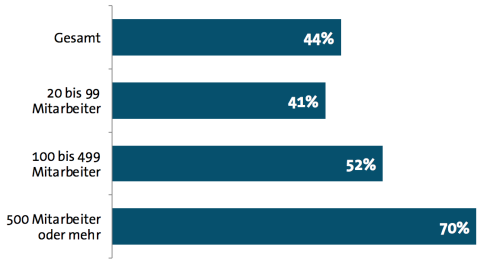

Abbildung 4: Clouds in der Industrie

Aufgrund der Verwendung von beschränkten Ressourcen wird oftmals auf *cloudbasierte Lösungen* industriellen im industriellen Bereich zurückgegriffen, wie die Abbildung 4 aufzeigt. Diese Verlagerung bedeutet, dass Rechenoperationen sowie die gesamte Datenspeicherung nicht auf dem limitierten Gerät selbst durchgeführt werden muss, es stattdessen vielmehr mit der Cloud in ständiger Kommunikation steht.
Trotz der verbreiteten Verwendung, sind Cloudlösungen nicht durchgehend akzeptiert. Dies ist der Statistik in Abbildung 5 zu entnehmen. Dabei handelt es sich um eine Statistik, die bei der *Pressekonferenz Cloud Monitor 2015* präsentiert wurde [11]. Die größten Bedenken beziehen sich auf die Sicherheit der Cloud, insbesondere auf Schutz der sensiblen (personenbezogenen) Daten. Aus diesem Grund wird im Folgenden auf die Absicherung dieser eingegangen.

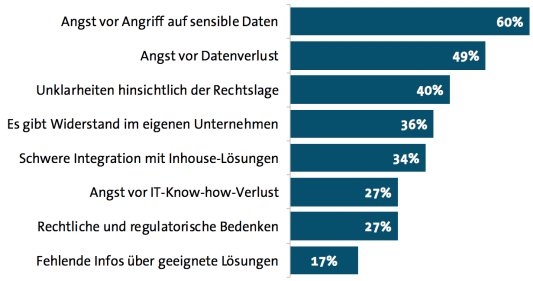

Abbildung 5: Sicherheitsbedenken der Nutzer

D. Trusted Cloud

Bei der *Trusted Cloud* handelt es sich um ein Programm des Bundesministeriums für Wirtschaft und Energie. Das Ziel dieses Projektes war es die Entwicklung einer sicheren Cloud-Infrastruktur, die böswilligen Administratoren den Zugriff auf sensible Daten verwehren soll. [12]
Das Grundkonzept sind speziell gesicherte Serverschränke, die bei unberechtigten Eingriffen sich herunterfahren. Zusätzlich werden jegliche Daten verschlüsselt auf den Festplatten gespeichert. Bei der Verarbeitung der Daten wird darüber hinaus zusätzlich nur flüchtiger Speicher verwendet, sodass nach Herunterfahren des Systems auf keinerlei Daten zugegriffen werden kann [9].

E. Zugriffskontrolle bei Cloud-Diensten

Aufgrund der Vielzahl von verschiedenen Nutzern bei einem Cloud-Dienst muss ebenfalls der Zugriff auf die Datensätze abgesichert werden. Dies könnte mittels einer *rollenbasierten Zugriffskontrolle* geschehen, doch besteht hierbei der Nachteil, dass ein möglicher Fehler in der Implementierung zur Aufhebung der Rollen führen würde. Dies ist nur möglich, weil solche Systeme im Regelfall vollständig in Software umgesetzt werden.
Eine bessere Möglichkeit bietet ein *attributbasiertes Verschlüsselungsystem* (s. Abbildung 6).

Zugrunde liegt hier ein *asymmetrisches Verschlüsselungsverfahren*. Die *Public Keys* sowie *Private Keys* werden dabei vom *Private Key Generator* ausgestellt.
In der Praxis würde also nur eine Person zugreifen können, die das nötige Attribut besitzt [9].
Bei mehreren Anwendern könnte man eine gruppenbasierte Zugriffskontrolle einführen. Dafür kann beispielsweise der Verzeichnisdienst *Active Directory* (AD) verwendet werden. Mit Hilfe von AD können nun Gruppen erstellt werden und digitale Zertifikate gruppenweit verteilt werden. Eine Zugriffskontrolle für mehrere Nutzer wäre damit gewährleistet.

F. Suchen auf verschlüsselten Daten

Aufgrund der Tatsache, dass die Daten verschlüsselt auf den Datenträgern vorliegen, müssen spezielle Algorithmen verwendet werden, die eine Suche auf verschlüsselten Datensätzen ermöglichen. Dies ist notwendig, da so die Daten zu jedem Zeitpunkt verschlüsselt auf dem Server vorliegen und nur clientseitig entschlüsselt werden. Böswillige Administratoren haben so im besten Fall keine Angriffsmöglichkeit (serverseitig).

G. Secure Indexes

In dieser Ausarbeitung beschränken wir uns auf *Symmetric Searchable Encryption* (SSE), also auf Verschlüsselungsverfahren, die auf einem symmetrischen Schlüssel/Schema basieren. Ein Index ist dabei eine Datenstruktur, die bei Eingabe eines Teilwortes die Pointer zu den Dokumenten wiedergibt, in denen das Teilwort enthalten ist [13]. Zusätzlich kann man einen Index als sicher ansehen, wenn die Suche nach einem Teilwort w nur mit Hilfe einer *Trapdoor* durchgeführt werden kann und ein Index alleine keinerlei Auskunft über den Inhalt gibt. Die Generierung einer sogenannten *Trapdoor* erfolgt dabei mittels des privaten Schlüssels des Nutzers. Im Folgenden wird eine beispielhafte *SSE* nach Goh [14] erläutert:
Zu Beginn muss der Begriff der *Bloom-Filter* eingeführt werden, da diese existenziell für die Erstellung der sicheren Indizes sind. Bei einem *Bloom-Filter* handelt es sich um eine Datenstruktur, welche eine Menge $S = \{s_1, ..., s_n\}$ mit n Elementen repräsentiert. Solch ein Filter besteht aus einem m-stelligen Bitarray, in welchem jede Stelle anfangs auf 0 gesetzt wird. Anschließend werden r unabhängige Hashfunktionen $h_1, ..., h_r$ gewählt, wobei $h_i : \{0,1\}^* \rightarrow [1, m]$ für $i \in [1, r]$ gilt. Zusätzlich werden für jedes Element $s \in S$ im m-stelligen Array die Positionen $h_1(s), ..., h_r(s)$ auf 1 gesetzt.
Die für die weitere Betrachtung relevante Operation ist nun das Prüfen, ob ein Element a in der Menge S ist. Hierfür werden die Positionen $h_1(a), ..., h_r(a)$ geprüft und falls all diese Positionen eine 1 sind, gehört a zur Menge S [15].
In dem folgenden Abschnitt wird die Konstruktion von *sicheren Indizes* nach [14] erklärt.

H. Konstruktion nach Goh [14]

- `Keygen(s)`: Ein Sicherheitsparameter s wird gegeben. Eine pseudozufällige Funktion $f : \{0,1\}^n \times \{0,1\}^s \rightarrow \{0,1\}^s$ wird gewählt sowie ein privater Schlüssel $K_{priv} = (k_1, ..., k_r) \leftarrow \{0,1\}^{sr}$
- `Trapdoor(`K_{priv}`, w)`: Eingabe für die *Trapdoor* sind der private Schlüssel K_{priv} und das Wort w, die Ausgabe für die beiden Parameter ist $T_w = (f(w, k_1), ..., f(w, k_r)) \in \{0,1\}^{sr}$
- `BuildIndex(`D`,` K_{priv}`)`: Eingabe für die Funktion `BuildIndex` ist das Dokument D mit einer einzigartigen ID $D_{id} \in \{0,1\}^n$ und einer Liste von Worten $(w_0, ..., w_t) \in \{0,1\}^{nt}$ sowie dem privaten Schlüssel $K_{priv} = (k_1, ..., k_n) \in \{0,1\}^{sr}$ Für jedes einzigartige Wort w_i, wobei $i \in [0, t]$ werden folgende drei Berechnungen angestellt
  1) Berechnung der *Trapdoor*: $(x_1 = f(w_i, k_i), ..., x_r = f(w_i, k_r)) \in \{0,1\}^{sr}$
  2) Berechnung eines eindeutigen Codewortes für jedes w_i: $(y_1 = f(D_{id}, x1), ..., y_r = f(D_{id}, xr)) \in \{0,1\}^{sr}$
  3) Abschließend wird das Codewort $y_1, ..., y_r$ in den *Bloom-Filter* des Dokumentes mit der D_{id} eingefügt

  Der Output ist der Index $I_{D_{id}} = (D_{id}, BF)$ für das Dokument mit der ID D_{id}
- `SearchIndex(`T_w`,` I_D`)`: Der Input ist die *Trapdoor* $T_w = (x_1, ..., x_r) \in \{0,1\}^{sr}$ für das jeweilige Wort w und ein Index I_D mit $I_{Did} = (D_{id}, BF)$ für das jeweilige Dokument mit der ID D_{id}. Es müssen die folgenden Berechnungen durchgeführt werden:

  1) Berechne das Codewort für das Wort w in D_{id} : $(y_1 = f(D_{id}, x_1), ..., y_r = f(D_{id}, xr)) \in \{0,1\}^{s,r}$

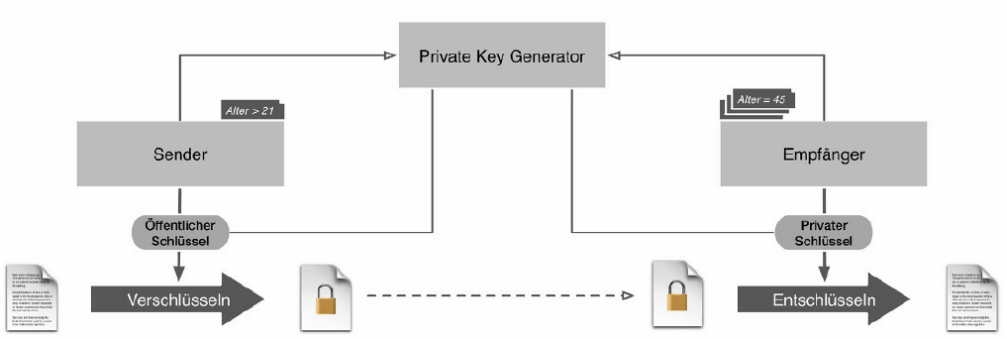

Abbildung 6: Attributbasiertes Verschlüsselungssystem

2) Wenn der *Bloom-Filter* nur Einsen an allen r Stellen mit $y_1, ..., y_r$ enthält
3) Falls dies der Fall ist, kam das Wort vor, gebe 1 aus. Ansonsten 0

Das obige Verfahren soll lediglich als Beispiel dienen. Es sind weitere Möglichkeiten denkbar. So können auch *asymmetrische Verfahren* angewendet werden ([16]).

I. Schutz von limitierten Geräten

Limitierte Geräte müssen ebenfalls geschützt werden. Dabei unterscheidet man zum einen zwischen der Kommunikation zwischen den jeweiligen Gerätschaften und dem eigentlichen Geräteschutz. Im Folgenden konzentrieren wir uns auf den internen Schutz limitierter Geräte.

J. Sichere Komponenten

Bei der Entwicklung limitierter Geräte werden häufig sichere Bauteile verwendet, welche beispielsweise dafür sorgen, dass die interne Schlüsselspeicherung sicher ist und die Schlüssel auch nicht ohne weiteres gelesen werden können. Die Abbildung 7 veranschaulicht einen beispielhaften Aufbau eines limitierten Gerätes.

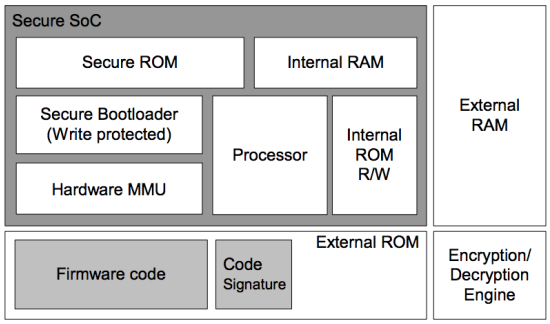

Abbildung 7: Diagramm eines System-on-a-Chip

1) Secure SoC: Das *Secure SoC* bietet Schutz für die geräteinternen Schlüssel. Diese werden verschlüsselt im *Secure Read-only-Memory* (Secure ROM) abgespeichert. Bei den Schlüsseln handelt es sich oftmals um Schlüssel, welche für digitale Signaturverfahren wie RSA, DSA oder auch ECDSA verwendet werden. Diese Schlüssel werden wie bereits gesagt verschlüsselt im ROM abgelegt. Dabei wird häufig auf Verschlüsselungsverfahren wie AES zurückgegriffen.[17]
Ein SoC bietet im besten Fall folgende Eigenschaften:

- Auf Secure ROM kann nicht physikalisch zugegriffen werden
- Die Buse innerhalb des Systems können nicht abgehört werden
- Das Ersetzen von Komponenten soll zu einer Nichtfunktionalität führen

Der letzte Punkt ist dabei besonders interessant, da das Gerät dadurch mögliche Angriffe verhindern kann. Der *Secure Bootloader* sorgt dabei, dass das Gerät nur mit einer korrekten Firmware hochfährt. Zusätzlich können *Physical Unclonable Functions* verwendet werden, deren Funktionsweise im Folgenden erläutert wird.

K. Physical Unclonable Functions

Keine zwei Stromkreise sind identisch. Diese Idee dient als Grundlage bei der Erstellung von *Physical Unclonable Functions* (PUFs). Durch Anwendung von PUFs sind Bauteile oder gesamte Geräte eindeutig identifizierbar.
In besten Fall sind PUFs eine Hardwareanalogon von Hashfunktionen. So gibt es im besten Fall sehr wenige Kollisionen und die Funktion selbst soll einfach durchführbar sein [18].
Formalisiert bedeutet das, dass es eine Challge c gibt und eine Funktion f, die diese Challenge als Eingabe nimmt. Die Ausgabe wäre dann $r = f(c)$. Dabei ist zu erwähnen, dass zwei Bedingungen gelten sollen:

- Für jede Challenge hat ein (integrierter) Stromkreis die gleiche Response
- Für jede Challenge haben unterschiedliche (integrierte) Schaltkreise unterschiedliche Responses

Die folgende Abbildung demonstriert eine beispielhafte PUF.

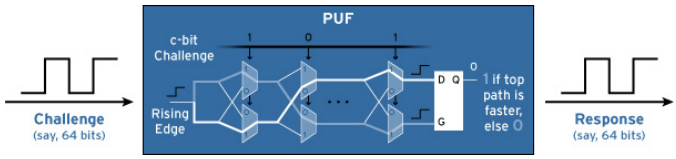

Abbildung 8: Multiplexer-PUF

Die PUF kann wie folgt formalisiert werden: $MUX - PUF : \{0,1\}^{64} \rightarrow \{0,1\}^{64}$.
Sie erwartet also eine 64bit-Folge als Eingabe und produziert eine dementsprechende 64-bit-Folge als Ausgabe.
Diese PUF besitzt zwei verschiedene Pfade und je nachdem auf welchem Pfad das Signal schneller war, wird eine 1 oder eine 0 ausgegeben. Die Anzahl der Challenge-Response-Paare ist bei dem obigen beispiel 2^{64}[19].

Dieses Verfahren ermöglicht eine Identifikation von Komponenten (Schaltkreisen). Als Folge daraus können limitierte Geräte in einem internen Speicher die Challenge-Response-Paare für die Komponenten speichern und sich selbstständig im Betrieb überprüfen. Falls nämlich für eine Komponente kein Challenge-Response-Paar, also keine eindeutige Identifikation vorliegt, kann sich das Gerät abschalten oder gar nicht erst hochfahren (*Secure Boot*)[20].
Darüber hinaus können PUFs auch zur sicheren Kommunikation zwischen Geräten verwendet werden. So können bestimmte Kommunikationspartner nur nach erfolgreicher Identifikation kommunizieren und werden andernfalls abgewiesen [9], [21].

VII. Fazit

In dieser Ausarbeitung wurden verschiedene Sicherheitsaspekte im *Internet of Things* untersucht. Zu Beginn wurde auf allgemeine Sicherheitsmechanismen eingegangen, es wurden mögliche Schwachstellen und Verbesserungen genannt. Im Anschluss folgte eine Sicherheitsbetrachtung im privaten und industriellen Bereich. Es wurde deutlich, dass die Kombination zweier IKT-Bereiche neue Herausforderungen mit sich bringt, die separat bei beiden System zuvor in dieser Form nicht existierten.
Besonders bemerkenswert sind hierbei die unterschiedlichen Möglichkeiten, Komponenten im industriellen Bereich abzusichern. Die Möglichkeiten reichen von hardwarebasierten Lösungen wie *PUFs* oder sicheren Bauteilen bis hin zu Softwarelösungen oder allgemeinen Verschlüsselungen auf der Transportebene.

Literatur

[1] M. Friedewald, O. Raabe, P. Georgieff et al., "Ubiquitäres Computing: Das „Internet der Dinge" – Grundlagen, Anwendungen, Folgen," Berlin 2010.
[2] Q. Sun, H. Garcia-Molina, "Using Ad-hoc Inter-vehicle Networks For Regional Alerts," 2004.
[3] Hewlett Packard Development Company, "Report, Internet of Things Research Study," September 2014. [Online]. Available: http://www8.hp.com/h20195/V2/GetPDF.aspx/4AA5-4759ENW.pdf
[4] Open Web Application Security Project, "Internet of Things Top Ten," 2014. [Online]. Available: https://www.owasp.org/index.php/OWASP_Internet_of_Things_Top_Ten_Project
[5] M. Kovatsch, S. Duquennoy, A. Dunkels, "A Low-Power CoAP for Contiki," 2011.
[6] G. Reiter, "Wireless connectivity for the Internet of Things, One size does not fi t all," Juni 2014.
[7] V. P. Andelfinger, T. Hänisch, "Internet der Dinge, Technik, Trends und Geschäftsmodelle," Wiesbaden 2015.
[8] Landeskriminalamt Nordrhein-Westfalen, Sachgebiet 32.2 - Technische Prävention, Prävention von Vermögens- u. Eigentumsdelikten, "Smart Home und Connected Home, Sicherheitsempfehlungen für Hersteller, Fachhändler und Handwerker," September 2014.
[9] C. Eckert, N. Fallenbeck, "Industrie 4.0 meets IT-Sicherheit: eine Herausforderung!" Heidelberg Berlin 2015.
[10] "Security Services für Industrie 4.0," June 2015. [Online]. Available: http://www.channelpartner.de/a/security-services-fuer-industrie-4-0,3044992
[11] Cloud-monitor. [Online]. Available: https://www.bitkom.org/Presse/Anh%C3%A4nge-an-PIs/2015/M%C3%A4rz/Cloud-Monitor.pdf
[12] Wind River Systems, "Security in the Internet of Things," 2015.
[13] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," *Proceedings of the 13th ACM conference on Computer and communications security, CCS '06, ACM*, 2006.
[14] E.-J. Goh, "Secure Indexes," *IACR Cryptology ePrint Archive 2003*, 2003.
[15] Broder A., Mitzenmacher M., "Network Applications of Bloom Filters: A Survey," 2002.
[16] Boneh D., Di Crescenzo G., Ostrovsky R., Persiano G., "Public Key Encryption with keyword Search," 2004.
[17] A. MS, "Security needs in embedded systems," *IACR Cryptology ePrint Archive*, 2008.
[18] S. Devadas, "Physical unclonable functions and applications."
[19] "Physical Unclonable Functions." [Online]. Available: http://studiopresence.com/client/verayo/technology
[20] G. E. Suh, S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE*, 2007.
[21] C. Eckert, "IT-Sicherheit und Industrie 4.0," *Fachzeitschrift für Innovation, Organisation und Management*, 2014.