Säkerhet inom sakernas internet

Peter Eriksson 36657

Kandidatavhandling i datateknik, inbyggda datorsystem

Åbo Akademi

Fakultet för naturvetenskap och teknik

Handledare: Annamari Soini, Johan Lilius

Innehåll

Referat

Då världen började koppla små apparater med olika sensorer till internet för att reglera och övervaka, uppstod sakernas internet. Antalet olika apparater inom sakernas internet har vuxit väldigt mycket under senaste tiden och med denna ökning måste även utvecklingen av säkerheten inom sakernas internet hålla ikapp. I denna avhandling analyseras först säkerhetsriskerna och sedan undersöks hur säkra apparaterna verkligen är samt den säkerhet som implementeras av dessa apparater. Fokus ligger på nätverkssäkerhet men går även igenom säkerheten på den fysiska nivån. Avhandlingen tar dessutom upp de områden inom säkerheten som ännu borde utvecklas samt nya säkerhetsprotokoll som håller på att utvecklas. Syftet med avhandlingen är att få en bättre bild av den säkerhet som finns eller inte finns i de apparater som vi köper och installerar i våra hem eller i våra företag.

Sökord: Sakernas internet, säkerhet, IoT,

1. Inledning

Sakernas internet (eng. *Internet of Things, IoT*) är en relativt ny term som myntades år 1999 av Kevin Ashton, en brittisk entreprenör [1]. Det är ett mycket brett begrepp som omfattar otroligt många olika apparater och generellt anses som inbyggda apparater som har mjukvara och sensorer samt är kopplade till internet.

Antalet apparater som går under definitionen sakernas internet har ökat otroligt mycket på sistone. Gartner [2], ett stort it-forsknings- och konsultföretag, förutspådde att det totala antalet apparater som är i användning år 2016 uppgår till 6,4 miljarder, och år 2020 finns det över 20 miljarder apparater. På grund av denna ökning är det otroligt viktigt att apparaterna är säkra på alla nivåer, så att den information som apparaterna samlar inte hamnar i fel händer eller modifieras, och så att ingenting annat händer med apparaterna. De flesta sakernas internet-apparater som konsumenter köper brukar ha relativt dålig säkerhet vilket beror på både användarens dåliga kunskap i hur man använder apparaten och själva apparatens säkerhet.

Vissa sårbarheter som har förekommit har varit relativt ofarliga medan andra har varit mera allvarliga. Man måste ändå komma ihåg att säkerhet utan en enda sårbarhet är mycket dyrt att uppnå med tanke på forskning och utveckling, men nära total säkerhet måste vi försöka sträva till. Apparater implementerar olika protokoll, en uppsättning regler och rutiner, för att kunna kommunicera med varandra. Protokoll inom kommunikationssäkerhet används för att till exempel en apparat skall veta vilken krypteringsalgoritm den ska använda för att kryptera ett meddelande. Dessutom skall den apparat som mottar meddelandet också vet vilken krypteringsalgoritm den skall använda för att dekryptera meddelandet. Följande protokoll är bara en del av alla protokoll som finns att använda inom sakernas internet, men de är troligen de mest populära som används för tillfället.

2. Vad är sakernas internet?

I dag finns sakernas internet-apparater överallt och de kan användas till att övervaka och mäta nästan vad som helst, bara det finns en sensor som kan användas. Dessutom kan man spara och analysera de data som apparaterna och sensorerna övervakar för att exempelvis hitta mönster eller oregelbundenheter. Listan på sakernas internet-apparater är enorm och nya apparater kommer ut varje vecka, om inte varje dag. En ny trend av sakernas internet-apparater som har uppkommit relativt nyligen är träningsarmband och andra hälsoapparater. Dessa apparater använder sensorer som kan mäta till exempel ens puls, steg, antalet trappor man har gått upp och ifall man sitter eller står. Alla data skickas antingen till ens telefon eller till ens dator där det analyseras och en översikt kan exempelvis presenteras med hjälp av olika grafer eller andra sätt att visa data.

Hemautomation har också blivit populärt på sistone, apparaternas mål är att göra livet hemma lättare. Exempel på apparater som används i hemautomation är lampor som kan styras från ens telefon, kylskåp som är kopplade till internet och har inbyggda kameror som tar bilder av innehållet i kylskåpet, och rullgardiner samt fönster som styrs antingen enligt rumstemperaturen eller från en applikation på ens telefon.

En annan kategori med apparater som har vuxit enormt på sistone är små datorer som Raspberry Pi. Med dessa små datorer kan man själv lätt koppla olika sensorer och programmera dem så som man själv vill. Hemautomation går lätt att själv bygga och förverkliga med en Raspberry Pi och olika sensorer. Då man själv kan koppla och bygga sin egen sakernas internet-apparat är möjligheterna att uppfinna nya apparater obegränsade.

Sakernas internet-apparater används även inom industrin för att övervaka olika maskiners slitage vilket gör det lättare att veta när en maskin måste servas. Detta minimerar tiden en maskin är ur bruk, då man kan lättare planera en servicetid i god tid före maskinen får problem istället för just när maskinen råkar gå sönder eller får problem. [1]

Apparater används dessutom i logistikföretag där de har samlat data på hur ekonomiskt lastbilschaufförerna kör och enligt det kan företaget vidareutbilda de som kör mindre

ekonomiskt och på så sätt spara på bränsle. Med samma data kan man också se om det finns bättre rutter som antingen skulle vara kortare eller mindre trafikerade. Liknande mål finns inom avfallshantering där man har installerat apparater i rosklådor som kan mäta när lådan är full och skall tömmas. Denna information skickas till en avfallscentral som kan planera den kortaste och mest effektiva rutten sopbilen skall ta och därmed sparar man både tid och pengar samt på naturen då man kör mindre i onödan. Systemet är redan i användning i åtminstone en plats i Finland enligt [14].

Som sades innan är sakernas internet-apparater i grund och botten ett inbyggt system med antingen inbyggda sensorer eller sensorer som är kopplade till systemet, samt en uppkoppling till internet som kan antingen vara trådlösa eller trådbunden. Den trådlösa uppkopplingen kan dessutom delas in i lokala nätverksprotokoll som till exempel NFC, Wi-Fi, ZigBee och 6LoWPAN, eller breda nätverksprotokoll som till exempel GPRS, 3G, 4G och LTE [1]. Skillnaden mellan dessa nätverksprotokoll är bland annat räckvidden, storleken på meddelanden, användningsområde och hastigheten. Dessutom kan man, med de lokala nätverksprotokollen, välja om man vill vara kopplad till internet eller om de lokala apparaterna är kopplade tillsammans.

3. Olika sårbarheter som har förekommit

För att visa hur illa det kan gå då man inte har säkerheten i skick undersöks kort först ett par exempel. Det första exemplet hände nyligen i USA då Dr. Charlie Miller och Chris Valasek [3] lyckades bryta sig in i en bils dator och genom fjärråtkomst kunde ta över hela bilen. Männen lyckades koppla av bromsarna, vända på ratten, koppla på radion och kunde i stor utsträckning styra allt som var kopplat till bilens CAN-buss. Allt detta kunde de göra medan de själva satt i en bil åtminstone 32 meter borta. Sårbarheten som gjorde detta möjligt var ett lätt Wi-Fi-lösenord samt port nummer 6667 som var öppen på det cellulära nätet som bilarna hade i användning. Efter att Miller och Valasek publicerade sin undersökning och sitt resultat var biltillverkarna tvungna att återkalla totalt 1,4 miljoner fordon och göra ändringar i det mobila nätet som bilen använde.

Det andra exemplet på sårbarheter i sakernas internet-apparater kommer från en fallstudie som publicerades 7 september 2015 [4]. I fallstudien undersökte man olika babyvakter för sårbarheter. Dessa apparater använder föräldrar för att övervaka om deras barn mår bra eller lyssna ifall de börjar skrika. I fallstudien kom man fram till att flera apparater som undersöktes hade bakdörrar med hårdkodade lösenord, vissa hade osäker dataöverföring över nätet och andra hade mindre farliga sårbarheter. Genom dessa sårbarheter kunde någon ta kontroll över apparaten och till exempel titta på videoströmningen, höra på allt som man sade nära apparaten eller spela upp ljud från apparaten så att barnet vaknar. Alla sårbarheter som hittades rapporterades till tillverkaren så att de kunde åtgärda dem så snabbt som möjligt. Dessa exempel omfattade bara en handfull sakernas internet-apparater av totalt flera miljarder apparater som är i användning i dag. Detta ger ändå en bra bild av hur viktigt det är att satsa på att säkra alla apparater så bra som möjligt och försöka sträva efter standardiserade säkerhetsåtgärder.

4. Riskanalys för sakernas internet

För att kunna veta hur man skall gå tillväga för att säkra sakernas internet måste man göra en riskanalys. Riskanalysen delas här in i två delar, riskbedömning samt riskhantering. I riskbedömningen mäter man sannolikheten och allvaret i riskerna samt identifierar olika risker och utvärderar riskerna. I riskhanteringen går man igenom hur man borde hantera risken ifall man stöter på en.

4.1 Riskbedömning

Genom att först identifiera riskerna får man en bild av hur många sårbarheter det potentiellt finns. I sakernas internet-apparater är den farligaste risken att någon tar total kontroll över apparaten, antingen över nätverket eller fysiskt, som bevisades möjligt i exemplet ovan. En annan risk är att någon avlyssnar den information som apparaten skickar eller att någon ändrar på den information som skickas. Det finns även risken att någon vill förstöra apparater. Sannolikheten för olika risker är svår att estimera men då man utvärderar riskerna kan man komma fram till om de är mera eller mindre sannolika.

Då det gäller risken av att någon utomstående får total kontroll över en apparat är sannolikheten mindre, ty det tar mera tid och möjligtvis pengar att åstadkomma en sådan attack. Sannolikheten att någon skulle avlyssna den information som man skickar från en apparat är större på grund av att det är potentiellt lättare.
Dessa sannolikheter kan dock variera mycket på grund av att alla apparater har olika sårbarheter som är antingen lättare eller svårare att använda då man attackerar apparaten.

HP i [15] undersökte de tio mest populära sakernas internet-apparaterna för sårbarheter och risken att de kunde hackas. Av dessa apparater kom HP fram till att 70% av dem använde sig av icke-krypterad kommunikation. Detta gör det mycket lätt för attackerare att avlyssna den information som skickas. HP kom också fram till att 60% av apparaterna de undersökte använde inte kryptering då de skulle uppdatera till en nyare version av den inbyggda programvaran. Detta innebar att vissa nedladdningar av den nya programvaran kunde fångas upp och montera på en annan dator där man sedan kunde modifiera programvaran.

4.2 Riskhantering

Hur man hanterar olika risker som kan uppkomma beror på hur mycket tid och pengar man vill sätta på forskning och utveckling av apparaternas säkerhet. Antingen kan man göra en grundlig analys över alla risker som kan uppkomma då apparaten används eller så låter man det bli och löser problemen då de uppkommer senare, om de alls uppkommer. Om man ser på de tidigare exemplen på sårbarheter som har uppkommit och på undersökningen som HP gjorde [15] låter de flesta företag bli att analysera, utvärdera och hantera riskerna. Detta leder till ett enormt antal apparater som är fulla av sårbarheter som bara väntar på att bli utnyttjade.
I kommande kapitel tas det upp olika protokoll som säkerställer sakernas internet-apparater i dag samt kommande protokoll som förbättrar olika aspekter av de nuvarande protokollen.

5. Nätverksstandarder som är i användning och deras säkerhet

Valet av protokoll ligger i huvudsak hos tillverkaren av apparaterna då apparaten planeras. Tillverkaren måst tänka på användningsområde och miljö, priset på apparaten, operativsystemet och många andra faktorer. Användaren kan dock även välja protokoll ifall tillverkaren har implementerat flera olika protokoll som används inom samma område.

5.1 Auktorisering och autentisering

Autentisering behövs inom sakernas internet då antalet apparater ökar och varje apparat har en egen identitet. En av orsakerna till varför man vill att sina apparater har en identitet är att man vill att de data som apparaten skickar eller använder identifieras som ens eget, att data är inte anonyma. Auktorisering används då till exempel en apparat vill ha access till en sensor eller till data som den behöver. Enhetsautentisering innebär en säker länk mellan två apparater då de kommunicerar med varandra. Detta kan ske genom till exempel att båda använder en delad nyckel för att kryptera meddelanden.

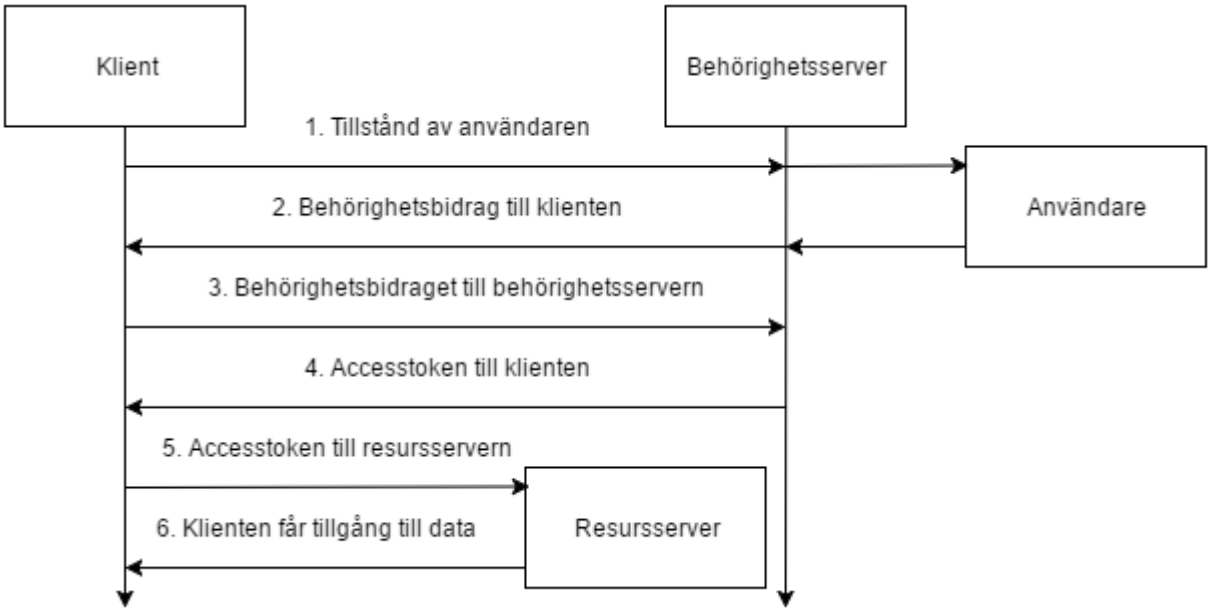
Auktorisering och autentisering sker på många nivåer då det kommer till sakernas internet. Som exempel kan man ta en termometer som kopplas till nätet. Först måste termometern autentiseras med förmedlingsnoden (eng. *gateway*), sedan måste förmedlingsnoden autentisera sig med molnklienten då den ska skicka data till molnet och till slut måste även programmet som vill analysera de data som termometern har samlat autentisera sig med molnklienten.

Enligt Paul Madsen [5] finns det två möjliga standardiserade ramverk som går att använda för denna typ av autentisering och auktorisering. De är OAuth 2.0, som används till auktorisering, och OpenID Connect 1.0 som används, tillsammans med OAuth 2.0, för autentisering. Dessutom används olika sätt att autentisera apparater då de kopplas till olika nätverk, där krypteringsnycklar även beräknas.

5.1.1 OAuth 2.0

Enligt dokumentationen på OAuth 2.0 [7] använder protokollet sig av ett accesstoken över HTTP för att tillåta en tredje part att använda resurser utan att slutanvändaren måste ge sin

information till den tredje parten. Autentiseringsprocessen skulle i praktiken ske på följande sätt: 1. Klienten, en applikation, ber genom en behörighetsserver om tillstånd av användaren, en person, att få använda en viss resurs som användaren har tillgång till. 2. Användaren godkänner begäran och skickar ett behörighetsbidrag tillbaka till klienten. Bidraget kan antingen vara ett av fyra färdigt definierade bidrag eller ett eget bidrag. 3. Klienten skickar behörighetsbidraget till behörighetsservern. 4. Behörighetsservern kontrollerar att behörighetsbidraget är korrekt samt ännu i kraft och skickar sedan tillbaka ett accesstoken. 5. Klienten skickar nu accesstokenet till servern där användarens resurser som klienten vill ha tillgång till finns. 6. Resursservern bekräftar att den token som togs emot är korrekt och ger sedan klienten tillgång till användarens resurser. Med att använda detta sätt för att auktorisera, finns det ingen risk att användarens användarnamn och lösenord hamnar i fel händer.



Figur 1. OAuth 2.0 protokollflöde

5.1.2 OpenID Connect 1.0

OpenID Connect 1.0 [6] används ovanpå OAuth 2.0-protokollet då man behöver både auktorisering och autentisering, för att få information om slutanvändaren samt för att verifiera identiteten av slutanvändaren med hjälp av en autentiseringsserver. Slutanvändaren i detta fall kan antingen vara en apparat eller en mänsklig användare. OpenID Connect

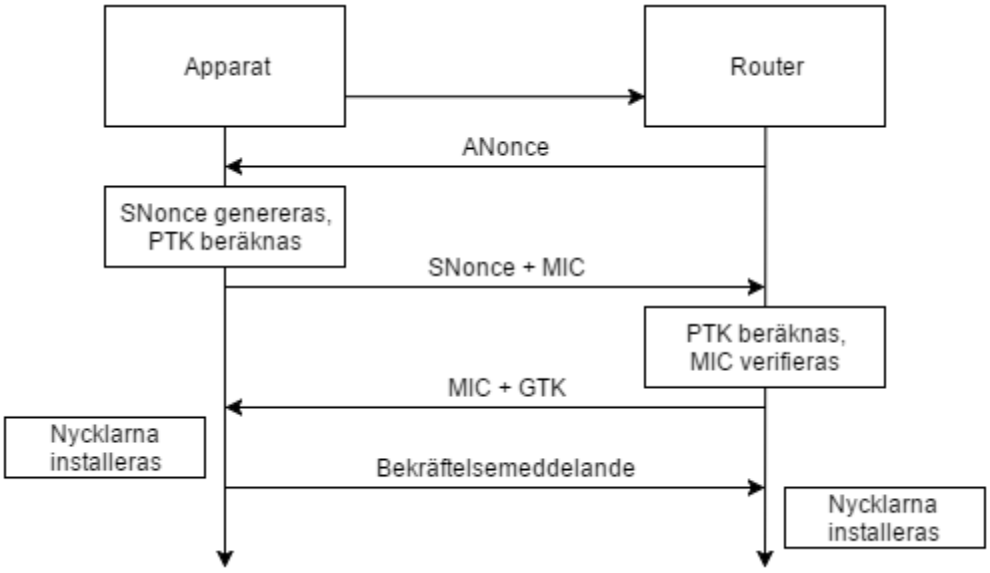
använder ID-token som innehåller information om användaren och kan begäras av en klient. Innehållet i ett ID-token kan till exempel vara identiteten eller namnet på användaren, vem som har utfärdat ett token, vilken klient som skall få ett token, en digital signatur och ett ID-token kan vara krypterat om det behövs. Det finns tre olika sätt att erhålla ID-token, behörighetskodflöde, implicitflöde och hybridflöde.
Enligt OpenID connect core specifikationer [20] har alla flöden samma fyra startsteg, vilka är; 1. Klienten förbereder en begäran som innehåller de parametrar som behövs. 2. Klienten skickar begäran till tillståndsservern. 3. Tillståndsservern autentiserar användaren. 4. Tillståndsservern erhåller användarens tillstånd. En av skillnaderna mellan de olika flöden är steg fem där behörighetskodflöde skickar slutanvändaren tillbaka till klienten med en behörighetskod. Implicit kod skickar användaren tillbaka till klienten med en ID-token och ett accesstoken om ett sådant begärdes. Hybridflöde kommer att skicka användaren tillbaka med en behörighetskod och beroende på responstypen kan även andra parametrar skickas. Vid steg sex i implicita flödet, det sista steget, kommer klienten att bekräfta det ID-token som den fick och sedan hämta slutanvändarens unika identifierare. Både behörighetsflöde och hybridflöde har samma steg 6-7 som är; 6. Klienten skickar behörighetskoden till token-klienten. 7. Klienten får tillbaka ett ID-token och ett accesstoken. Sista steget i både behörighetsflöde och hybridflöde är samma som sista steget i implicitflödet.

5.1.3 Autentisering i 802.11 trådlösa nätverk

Då man kopplar en apparat till en trådlös accesspunkt måste apparaten autentisera sig med wi-fi-stationen. Det finns flera protokoll som används för autentisering och autentiseringsinställningarna ställs in på routern. Om man väljer det säkraste protokollet för autentisering på en router hemma ska man välja WPA2-Personal (eng. *Wi-Fi Protected Access 2, WPA*) och om man har ett företag borde man välja WPA2-Enterprise [21].

WPA2-Personal autentiseringsprocessen använder en gemensam statisk nyckel (eng. *Pre-Shared Key, PSK*) och en 4-vägs handskakning. Den gemensamma statiska nyckeln bildas från nätverkets lösenord samt namnet på nätverket (SSID), dessa sätts in i en funktion, PBKDF2 (eng. *Password-Based Key Derivation Function 2*), och som utdata kommer PSK som är 256-bitar lång

Enligt IEEE 802.11-standardens dokumentation [17] sker en 4-vägars handskakning på följande sätt: Först skickar accesspunkten en nonce[1] (ANonce), ett slumptal som används endast en gång, till apparaten som skall kopplas till nätet. Till näst genererar apparaten en nonce (SNonce) som den använder tillsammans med ANonce som den fick, apparatens MAC-adress, accesspunktens MAC-adress och PSK-nyckeln att beräkna en parvis övergående nyckel (eng. *Pairwise Transient Key*, PTK). PTK består av tre olika nycklar, nyckelbekräftelsenyckel (eng. *key confirmation key*), nyckelkrypteringsnyckel (eng. *key encryption key*) och en temporal nyckel (eng. *Temporal Key*). Efter detta skickar apparaten dess SNonce tillsammans med en MIC (eng. message integrity code) till accesspunkten. Accesspunkten beräknar, på samma sätt som apparaten, PTK och verifierar den MIC som apparaten skickade. Då accesspunkten har beräknat PTK och verifierat MIC, skickar den ett meddelande med sin egen MIC, ifall PTK-nyckeln skall installeras på apparaten eller inte och en grupp övergående nyckel (eng. *group trancient key, GTK*) till apparaten, utan att själv installera nycklarna ännu. Då apparaten får det tredje meddelandet, installerar den nycklarna och skickar tillbaka ett bekräftelsemeddelande att nycklarna har installerats. Till sist, när accesspunkten har mottagit bekräftelsemeddelandet kommer accesspunkten att själv installera och använda nycklarna.



Figur 2. 4-vägshandskakning

[1]nonce: slumptal som bara används en gång [25]

Skillnaden mellan WPS2-Personal och WPS2-Enterprise är användningen av IEEE 802.1X standarden, som baserar sig på utdragbart autentiseringsprotokoll (eng. *Extensible Authentication Protocol*), och RADIUS-protokollet. Istället för att ha ett lösenord som alla använder för att koppla sig till nätverket, har varje användare ett eget användarnamn och lösenord. Detta gör det lättare att byta en användares lösenord om hen tappade till exempel bort sin telefon som innehåller både användarnamnet och lösenordet till nätverket. Dessutom används en RADIUS-autentiseringsserver före 4-vägshandskakningen utförs. För att autentisera användare samt bilda en parvis primärnyckel (eng. *pairwise master key, PMK*). 4-vägs handskakningen fungerar på samma sätt som för WPS2-Personal men PMK används istället för PSK. [17]

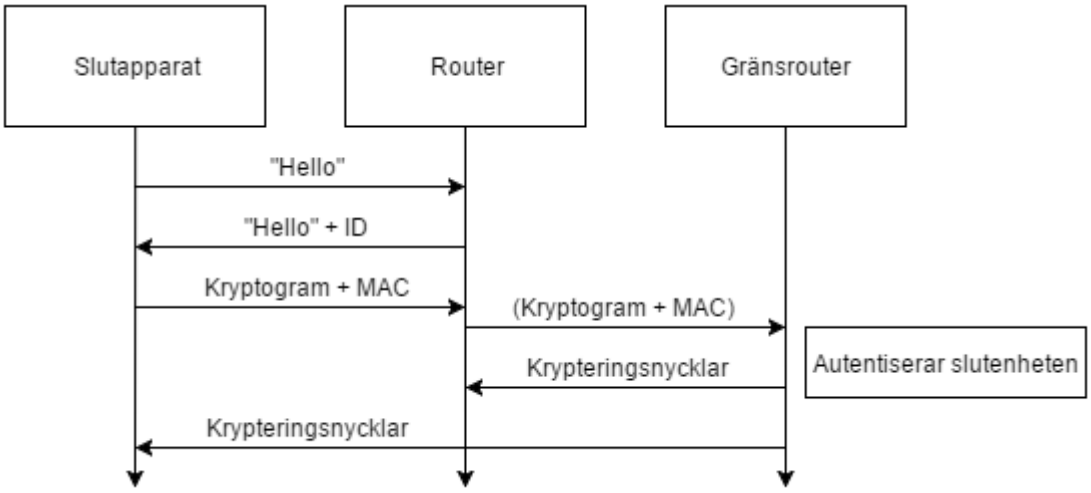
5.1.4 Autentisering i 6LoWPAN-nätverk

6LoWPAN står för IPv6 over Low Power Wireless Personal Area Networks och används nuförtiden i små och energisnåla sakernas internet-apparater [9]. Varje apparat har en egen IPv6-adress och kan bygga upp ett trådlöst mesh-nätverk av 6LowPAN-apparater som antingen kan kommunicera en-till-många- eller många-till-en-ruttning. Ett mesh-nätverk byggs upp av flera noder som kommunicerar med varandra antingen direkt eller via en annan nod. Varje apparat i ZigBee-nätverket fungerar som en nod och alla noder kommunicera, antingen direkt eller genom en annan nod, med en gränsrouter som i sin tur är kopplad till Internet. Gränsroutern är en enkel förmedlingsnod och behövs på grund av att 6LoWPAN-apparater använder en komprimerad IPv6 adress som måste konverteras till en IP adress som kan användas på internet. Systemet tillåter även flera gränsroutrar som kan kommunicera med varandra och mesh-nätverket och är själva förstås kopplade till internet. Det finns dock flera olika sätt att sköta autentiseringen i 6LoWPAN-nätverk, nya förslag och scheman dyker upp hela tiden. Följande autentiseringssätt, SAKAS, ser ut att vara ett lovande och säker protokoll för autentisering i ett 6LoWPAN-nätverk.

Det finns flera olika förslag om hur man skall autentisera apparater i 6LoWPAN nätverk. Ett förslag som Amira Barki et al. hänvisar till i [8] är SAKES [11], föreslagen av Hassen Redwan Hussen et al. SAKES står för Secure Authentication and Key Establishment Scheme (säker autentisering och nyckeletableringschema). SAKES-schemat går ut på att först

autentisera slutapparaten (eng. *end device*) och sedan ifall autentiseringen lyckades beräkna en sessionsnyckel (eng. *Session Key*) som används för att kryptera kommunikationen.

Autentiseringsprocessen i SAKES går ut på att 6LoWPAN-gränsroutern kontrollerar att 6LoWPAN-slutapparat har korrekt 6LoWPAN-router granne som den skickar meddelandet igenom. Slutapparaten börjar med att ta reda på vilken dess närmaste router är genom att sända en ”HELLO”-begäran. Begäran besvaras av 6LoWPAN-routern med ett annat ”HELLO”-meddelande. ”HELLO”-meddelandet från routern innehåller 6LowPAN-routerns ID-nummer. Efter att slutapparaten har fått routerns ID-nummer använder den det, tillsammans med andra variabler, för att bilda ett kryptogram och ett meddelandeautentiseringskod som skickas till routern. Routern börjar med att först modifiera meddelandet med sin egen autentiseringskod som den sedan skickar vidare till autentiseringsmodulen i gränsrouter. Gränsroutern kontrollerar att meddelandet är äkta och sedan kontrollerar om slutapparaten är registrerad hos autentiseringsmodulen. Ifall den är det skickar den ett meddelande tillbaka till routern och ett meddelande till slutapparaten. Om slutapparaten inte är registrerad hos autentiseringsmodulen kommer den att kontrollera att den routern som är närmast slutapparaten är den som skickar meddelandet och sedan besluta om slutapparaten är äkta eller inte.



Figur 3. SAKES autentiseringsprocess

5.2 Kryptering och säkerhet av olika sakernas internet-nätverk

Enligt Amira Barki et al. [8] behöver man använda en infrastruktur för kryptering med öppen nyckel för att säkerställa en säker överföring av data antingen från apparat till apparat eller från apparat till en förmedlingsnod. Infrastrukturen för kryptering med öppen nyckel använder skilda nycklar för kryptering och dekryptering. Då man vill motta krypterad data skickar man den offentliga nyckeln till sändaren som använder den för att kryptera data. Då man vill dekryptera data använder man en privat nyckel som bara man själv känner till. Med mindre sakernas internet-apparater som har begränsade resurser, till exempel mycket lite minne eller långsam processor, kan det vara svårt att implementera kryptering med öppen nyckel medan det med större apparater går att använda då de har mera resurser som kan användas. Amira Barki et al. i [8] behandlar flera olika nyckelhanteringssystem som andra forskare har använt för att lösa problemet med hur man använder sig av kryptering med öppen nyckel på mindre apparater. De flesta sakernas internet kommunikationsprotokoll använder en variation av AES, *Advanced Encryption Standard,* för att kryptera kommunikationen men använder olika implementationer för att transportera krypteringsnycklarna som används för att kryptera data som skickas mellan apparaterna.

5.2.1 6LoWPAN-nätverk

Problemet enligt Sumit Goswami et al. i [10] är att noderna inte har tillräckligt med processorkraft för att kunna implementera en full infrastruktur med alla dess delar för kryptering med öppen nyckel. Delarna som hör bland annat till infrastrukturen är en certifikatutfärdare, en registraturfunktion, en certifikatdatabas och en certifikatbutik [22]. Deras idé är att man låter gränsroutern göra största delen av jobbet då man vet att den inte har några resursbegränsningar. Systemet går ut på att varje nod har två nycklar från den tid den kopplas in i nätverket, en egen hemlig nyckel och gränsnodens öppna nyckel, båda sparade i apparatens eller nodens minne. Gränsroutern har en lista på alla noder samt vilka hemliga och öppna nyckelpar noden använder. Ifall en nod vill skicka data till internet måste det gå via gränsroutern på så sätt att noden frågar gränsroutern efter destinationsnodens öppna nyckel och med den kan kryptera data som skickas.

I ett 6LoWPAN-nätverk som använder SAKES skulle krypteringsnycklarna fastställas av routern med information som den fick av gränsrouterns autentiseringsmodul. Routern skulle använda en-vägs Diffie-Hellman nyckelöverenskommelsemetoden. Krypteringsmetoden är baserad på elliptisk kryptering, som är ett slags kryptering med öppen nyckel. Elliptisk kryptering är en krypteringsmetod som kräver mindre minnesutrymme och processorkraft jämfört med andra asymmetriska då dess nycklar som den använder är mycket mindre [23]. Krypteringen använder de matematiska egenskaperna hos elliptiska kurvor.

5.2.2 ZigBee-nätverk

ZigBee, lika som 6LoWPAN, är baserad på IEEE 802.15.4 protokollet och används för kommunikation mellan apparater med låg effekt, en centralprocessor, låg komplexitet och dessa apparater är kostnadseffektiva. Exempel på ZigBee-apparater är olika sensorer och trådlösa strömbrytare som har små batterier som skall hålla länge. Dessa apparater används inom hemautomation. Dessutom har ZigBee en maximal hastighet på 250 Kbps och använder sig av 2,4 GHz trådlösa frekvensen. På grund av dessa låga resurser som ZigBee-apparaterna har är det utmanande att implementera komplexa och resurskrävande säkerhetsprotokoll. [16]

ZigBee-protokollet definierar högre lager som IEEE 802.15.4 inte definierar, IEEE 802.15.4 definierar *media access control* (MAC) skiktet samt fysiska lagret (PHY). Det innebär att ZigBee-protokollet definierar säkerheten på nätverksskiktet samt applikationsskiktet, och båda mekanismer som används för säkerheten baserar sig på 128-bit AES [16]. Krypteringen som ZigBee implementerar använder symmetriska nycklar vilket innebär att samma nyckel används för både kryptering och dekryptering. För att hålla ZigBee-nätverket och apparater inom nätverket säkra måste man hålla de symmetriska nycklarna säkra och omöjliga att gissa. Totalt använder ZigBee tre olika symmetriska nycklar, en primärnyckel (eng. *Master key*), en nätverksnyckel samt en länknyckel. Primärnyckeln används bara då två apparater bestämmer länknyckeln som de skall använda, länknyckeln används för kommunikation mellan två apparater på applikationsnivån

Nycklarna som används för kryptering kan antingen vara färdigt hårdkodade från fabriken, genom nyckeletableringsproceduren, genom nyckeltransport eller användaren bestämmer

själv nyckeln. Metoden som används beroende på vilken nyckel det handlar om. Primärnyckeln används då man börjar nyckeletableringsproceduren (eng. *Key establishment procedure*) som en inledande delad hemlighet mellan två apparater. Denna procedur används för att generar länknycklar. Primärnyckeln kan även skapas av användaren eller förtroendecentern. Länknyckeln är antingen skapad med nyckeletableringsproceduren, installerad färdig i fabriken eller så kan den fås från en nyckeltransport. Denna nyckel används för att kryptera enkelsändningsmeddelanden på applikationsnivån mellan två apparater. Sista nyckeln som används är nätverksnyckeln som används på nätverksnivån och alla apparater som är i samma ZigBee nätverk använder samma nätverksnyckel.

Hongwei Li et al. i [16] lyfter fram sex säkerhetsfunktioner som ZigBee har implementerat för att göra det säkrare. Dessa funktioner är: krypterade data, enhetsautentisering, ramintegritetskontroll, sekventiell färskhet, användningen av en Trust Center och CCM* krypterings algoritm.

Ramintegritetskontrollfunktionen fungerar så att varje meddelande innehåller en integritets kod (eng. *message integrity code*) som säkerställer att de data som kommer från en annan apparat har använt samma krypteringsnyckel som mottagande apparat har. Detta säkerställer dessutom att de data som skickats har modifierats på vägen. Sekventiell färskhet är en funktion som blockerar meddelanden som har skickats tidigare genom att använda en ordnad sekvens av inputs. Genom att använda denna funktion kan man blockera attacker eller meddelanden från att skickas vidare till alla noder i ett nätverk. Då en ny nyckel skapas kommer ZigBee-apparaten att återställa sekvensen.

Förtroendecenter (eng. *Trust Center*) fungerar som ett centrum för alla ZigBee-apparater. Dess viktigaste säkerhetsuppgifter inom nätverket är: autentisering och antingen tillåta eller förkasta nya apparater som vill ansluta sig till nätverket, hålla reda på nätverks nycklar och förnya dem ifall det behövs samt fungera som konfigurationshanterare (eng. *configuration manager*) som sköter om aktiveringen av säkerheten på apparaterna som är kopplade till det. Då en förtroendecenter vill byta en krypteringsnyckel använder det först den gamla nyckeln för att kryptera nya nyckeln och skickar sedan den krypterade nyckeln ut till apparaterna.

CCM* är ett blockkrypto och autentiseringsmetod som har samma egenskaper som CCM men har modifierats och fått till funktioner och förmågan att välja om man vill endast använda kryptering eller endast integritetkontroll. Blockkrypto är ett krypteringssätt som delar upp meddelanden i lika stora block som sedan krypteras och skickas till mottagaren. Mottagaren kan dekryptera blocken i oordning och dessutom motta blocken i oordning. de CCM är en kombination av autentisering och kryptering.

5.2.3 802.11-nätverk

Sakernas internet-apparater som har mera resurser till sitt förfogande kan använda 802.11-nätverk samt vanligare och lättare sätt att kryptera kommunikationen. De flesta större och kraftigare sakernas internet-apparater, till exempel en Raspberry Pi, använder Wi-Fi för att koppla sig till internet. Kryptering av Wi-Fi-kommunikationen väljs på routern och det finns flera olika protokoll som kunder kan välja mellan, WEP, WPA-PSK (TKIP/AES) eller WPA2-PSK (AES). Skillnaden mellan dessa protokoll är nyare och bättre säkerhet. [24]

WEP används mest i världen men är den minst säkra av alternativen. WEP algoritmen innehåller flera sårbarheter och kan hackas på minuter. WPA-PSK förbättrar säkerheten genom att använda TKIP (eng. *Temporal Key Integrity Protocol*) men på grund av att WPA återanvänt delar av WEP tog det inte länge för hackare att hitta sårbarheter i WPA. WPA2 måste används AES för kryptering och CCMP istället för TKIP. Det finns också WPA2-EAP men det finns på routers som är avsedda för företag. Skillnaden mellan PSK och EAP är att i PSK använder alla användare samma lösenord för autentisering medan i EAP har varje användare ett eget användarnamn samt lösenord.

WEP anses vara osäker att använda, samma gäller TKIP kryptering. Det säkraste protokollet som man kan använda av dessa är WPA2-PSK AES, som använder PSK och AES-CCMP för att autentisera användare samt kryptera de data som skickas över Wi-Fi. Efter att man har autentiserat sig med routern med sitt lösenord sker det en 4-vägshandskakning. CCMP till skillnad från CCM är ett specifikt protokoll, CCM är bara en metod som kan implementeras på flera olika sätt.

6. Säkerhet på den fysiska nivån

Med tanke på att en stor del av alla sakernas internet-apparater kommer att installeras på lättåtkomliga platser, till exempel vägskyltar, är det viktigt att skydda apparaten från fysiska attacker. Med fysiska attacker menas att man har fysisk åtkomst till apparaten och antingen försöker få rotaccess till apparaten eller tillgång till sensorerna för att manipulera dem. Med rotaccess har man tillgång till alla data i apparaten och även kontroll över hela apparaten.

6.1 Sätt att hindra fysiska attacker

Det finns tre sätt att förhindra att någon får tillgång till apparaten; antingen har apparaten ett hölje som inte går att öppna utan att söndra apparaten, eller så har apparaten inga uttag som kan användas, eller så gör man programvaran tillräckligt säker att det inte skulle vara värt för en hackare att bryta sig in i apparaten. För att kunna göra service på eller uppdatera apparaterna måste man ha tillgång till apparatens olika portar, ifall den har portar. Detta innebär att användningen av ett hölje samt att låta bli att ha portar skulle vara en dålig idé, då återstår endast alternativet att man skriver säkert fast program det vill säga ett säkert inbyggt operativsystem (eng. *firmware*).

Lösningen till många allmänna fel som görs då man utvecklar sakernas internet-apparater presenteras av Arijit Ukil et al. i [13] där de utgår ifrån att nätverket och kommunikationen mellan apparaten och nätverket är säkert. De anser att det finns två olika mål med attacker. Det första målet är att få tag på hemlig information och det andra målet är att få apparaten att sluta fungera. Attackerna går bland annat ut på att observera hur apparaten reagerar då man ändrar på spänningen och tillför antingen överspänning eller underspänning till apparaten, eller genom att manipulera klocksignalen och hoppas på att processorn låter bli att köra kod.

För att kunna skydda sig från attacker måste man undersöka riskerna och definiera olika attacker. Enligt Dorottya Papp et al. i [12] identifieras och definieras de attacker som kan ske mot inbyggda datorsystem. Dessutom ger Papp et al. flera exempel på olika attacker som har utförts på olika apparater. Några av sårbarheterna som Papp et al. observerade var dåligt implementerad kryptering av data och autentisering, programmeringsfel och svaga autentiseringsprocesser.

Problemet med krypteringen i vissa apparater var att de använde en dålig slumptalsgenerator som används då man genererar krypteringsnycklar, vilket ledde till att man lätt kunde gissa krypteringsnyckeln. Andra apparater hade hårdkodade strängar samt apparatens serienummer som används för att generera lösenordet till administratöranvändaren vilket gör det lätt att gissa lösenordet. Programmeringsfelen som förekom i apparaterna kunde leda till minneshanteringsfel och buffertöverfyllning då man attackerade exekveringsflödet. Alla dessa sårbarheter kunde förhindras men det skulle ha krävt mycket tid och pengar vilket alla företag inte är beredda att spendera.

Lösningen till dessa attacker går ut på att man bygger en säker exekveringsmiljö (eng. *Secure execution environment*) eller betrodd exekveringsmiljö (eng. *Trusted execution environment*) där kod och data är först verifierade och sedan tillåten att exekveras.

6.2 Säker exekveringsmiljö i sakernas internet-apparater

Den säkra exekveringsmiljön baserar sig på användningen av en speciell säker processor som minskar belastningen på huvudprocessorn. Denna processor kan antingen vara separat från huvudprocessorn eller en del av den, båda fallen har sina för- och nackdelar. Dessutom hör säkert minne för kod och data i apparaten till den säkra exekveringsmiljön. Minnet placeras på chippet i RAM-minnet. Kod och data som är utanför den säkra exekveringsmiljön ska inte lämnas oskyddade utan ska både krypteras och skyddas mot modifiering.

Arijit Ukil et al. nämner två olika chipp som använder olika plattform som används kommersiellt för att kunna ha en säker exekveringsmiljö. Första chippet, som tillverkas av Atmel, heter Trusted Platform Module och fungerar så att chippet innehåller ett ID-nummer som används för identifiering, är unikt för varje chipp och inte går att ändra. Det andra chippet tillverkas av ARM och heter TrustZone, och den kan man, enligt egna behov, programmera med C-kod och plattformen fungerar som ett sätt att isolera de data, mjukvara och hårdvara som är betrott från det som är icke-betrodd. Dessa chipp och säkerhetsplattform möjliggör en viktig funktion, säker start.
Säker start går ut på att före en exekverbar avbild (eng. *executable image*), det vill säga mjukvara, körs och får kontroll över data samt hårdvara, verifieras integriteten av avbilden.

För att dessutom säkerställa att säker start-koden inte har modifierats verifieras den med att använda digital signaturverifiering. Efter att säker start-koden har verifierats och exekverats börjar den verifiera resten av systemets delar och ifall ett lager inte kan verifieras eller är icke-betrott kommer systemet att stanna.

6.3 Kryptering av data

För att hålla de data som lagras på apparaten säkra och ur fel personers händer, används kryptering. Det finns flera olika krypteringsalgoritmer som används i sakernas internet-apparater. Bland dessa är RSA, ECC, AES och 3DES de mest populära. RSA och ECC är asymmetrisk kryptering och AES samt 3DES är symmetrisk kryptering. På grund av att vissa krypteringsalgoritmer är beräkningsintensiva kan man använda skild hårdvara som tar hand om krypteringen. Det kan till exempel vara en signalprocessor som har optimerats för att köra en viss krypteringsalgoritm.

Dessa var bara en del av alla möjligheter som finns för att säkra sakernas internet-apparater på den fysiska nivån. Det kommer ut nya protokoll och gamla protokoll korrigeras hela tiden då man hittar nya sårbarheter i protokollen.

7. Framtida utvecklingsområden

Då processorer blir snabbare för varje år och utvecklingen av kvantdatorer framskrider med snabb fart måste man utveckla nya säkrare protokoll och algoritmer för att säkerställa apparater och de data som är lagrade på dem. Problemet med framtiden är att man aldrig vet vilka teknologier som kommer att bli populära och hurdana technologier som kommer att uppfinnas. Dessutom är säkerhet ett område som alltid kommer att utvecklas och förbättras i och med att sårbarheter hittas nästan varje dag.

Sathish Alampalayam Kumar et al. i [18] nämner sju områden som borde utvecklas och förbättras. Av dessa sju områden är infrastrukturen för kryptering med öppen nyckel ett mycket viktigt område som borde utvecklas. Ett annat område som är viktigt att utveckla är system som OpenID connect som auktoriserar och autentiserar apparater. En effektivare och snabbare krypteringsmetod borde dessutom utvecklas. Ett exempel på en framtida

krypteringsmetod är homomorphic kryptering, som exempel finns Enigma [19]. Homomorphic krypteringen går ut på att alla data hålls krypterade och man inte behöver dekryptera de data som man skall hantera. Enigma är ett projekt som försöker utveckla ett krypteringsprotokoll som implementerar total homomorfism, det vill säga ingen dekryptering behövs och data hålls privat, men tills vidare är det bara en idé.

Behovet av en separat apparat i till exempel ett 6LoWPan eller ZigBee nätverk som sköter om distributionen av krypteringsnycklar borde slopas så att apparaterna själv kan hantera alla nycklar som behövs. Detta skulle minimera risken att alla noder i ett nätverk blir utsatta då en säkerhetsnod attackeras.

8. Sammanfattande diskussion

Sakernas internet-apparater har flera olika säkerhetsprotokoll till sitt förfogande och många av apparaterna har möjligheten att implementera dem. Problemet med säkerheten inom sakernas internet idag är att företagen inte spenderar tillräckligt med tid och pengar på implementationen av säkerhetsprotokollen eller att användaren inte vet hur man tar i bruk apparatens säkerhetsfunktioner. I dag verkar det vara användare och andra personer som har det som hobby att hitta sårbarheter i apparater och faktiskt varnar allmänheten om dem. Detta är bättre än att sårbarheterna hålls gömda ifrån allmänheten och används för olagligheter.

Fast vi alla har dessa säkra säkerhetsprotokoll som vi kan implementera är det ändå värt att poängtera att det alltid kommer att finnas sårbarheter i apparater, på grund av att det skulle bli alltför höga forskning- och utvecklingskostnader att hitta alla sårbarheter. Man måste också komma ihåg att den information som finns på apparater, som till exempel ens tvättmaskin eller smart lampa, är kanske inte otroligt viktiga och därmed behöver man inte den bästa krypteringsalgoritmen som finns.

Referenser

[1] Lopez Research. November 2013. *"An Introduction to the Internet of Things (IoT)", Part 1. of "The IoT Series"*

[2] R. Meulen, Gartner. [Online]. URL: http://www.gartner.com/newsroom/id/3165317 [hämtat 14.2.2016]

[3] Dr. C. Miller, C. Valasek. 10 Augusti 2015. *Remote Exploitation of an Unaltered Passenger Vehicle*

[4] M. Stanislav, T. Beardsley. 29 September 2015, *HACKING IoT: A Case Study on Baby Monitor Exposures and Vulnerabilities*. Rapid7.

[5] P. Madsen. SecureIDNews, [Online]. URL: http://www.secureidnews.com/news-item/authentication-in-the-iot-challenges-and-opportunities/ .[hämtat 18.2.2016]

[6] OpenID, [Online]. URL: http://openid.net/connect/ [hämtat 24.2.2016]

[7] D. Hardt. October 2012. *"The OAuth 2.0 Authorization Framework"*. Internet Engineering Task Force. *rfc 6749*.

[8] A. Barki, A. Bouabdallah, S. Gharout, J. Traoré. 2015. *"M2M Security: Challenges and Solutions"*. IEEE.

[9] Texas Instruments. [Online]. URL: http://www.ti.com/lsds/ti/wireless_connectivity/6lowpan/overview.page .[hämtat 26.2.2016]

[10] S. Goswami, S. Misra, C. Taneja, A. Mukherjee. 2014. Securing Intra-Communication in 6LoWPAN: A PKI Integrated Scheme. IEEE ANTS 2014 1570023215.

[11] H. R. Hussen, G. A. Tizazu, M. Ting, T. Lee, Y. Choi, K. Kim. 2013. SAKES: Secure Authentication and Key Establishment Scheme for M2M Communication in the IP-Based Wireless Sensor Network (6LoWPAN). IEEE ICUFN 2013. 246-251.

[12] D. Papp, Z. Ma, L. Buttyan. 2015. *"Embedded Systems Security: Threats, Vulnerabilities, and Attack Taxonomy".* IEEE. 2015 Thirteenth Annual Conference on Privacy, Security and Trust 145-152.

[13] A. Ukil, J. Sen, S. Koilakonda. 2011. *"Embedded Security for Internet of Things"*. IEEE.

[14] B. Schiller. Fastcoexist. 07.24.2013. *"These Trash Cans Know When They're Full, And Need Picking Up"*. [Online]. URL: http://www.fastcoexist.com/1682626/these-trash-cans-know-when-theyre-full-and-need-picking-up [hämtat 20.3.2016]

[15] Hewlett Packard Enterprise. November 2015. *"Internet of things research study"*.

[16] H. Li, Z. Jia, X. Xue. 2010. *"Application and Analysis of ZigBee Security Services Specification".* IEEE. 2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing. 494-497.

[17] IEEE Computer Society. 12 juni 2007. *"IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications"*. IEEE.
[18] S. A. Kumar, T. Vealey, H. Srivasrava. 2016. *"Security in Internet of Things: Challenges, Solutions and Future Directions"*. IEEE. 2016 49th Hawaii International Conference on System Sciences 5772-5781.
[19] R. Murdock. VDC Research. 08.04.2015. [Online]. URL: http://www.vdcresearch.com/News-events/iot-blog/iot-use-cases-for-enigma-homomorphic-encryption.html .[hämtat 28.3.2016].
[20] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, C. Mortimore. OpenID. 8 November 2014. [Online]. Available: http://openid.net/specs/openid-connect-core-1_0.html [hämtat 30.3.2016]
[21] E. Geier. Enterprise Networking Planet. 09.12.2010. [Online]. URL: http://www.enterprisenetworkingplanet.com/netsecur/article.php/3916561/Implement-WPA2-Enterprise-Encryption-on-Your-WLAN.htm .[hämtat 04.04.2016]
[22] M. Rouse. TechTarget. November 2014. [Online]. URL: http://searchsecurity.techtarget.com/definition/PKI . [hämtat 05.04.2016]
[23] E. Barker. Januari 2016." *Recommendation for Key Management Part 1: General"*. NIST. NIST Special Publication 800-57 Part 1 Revision 4.
[24] J. Fitzpatrick. How-To Geek. 16.07.2013. [Online]. URL: http://www.howtogeek.com/167783/htg-explains-the-difference-between-wep-wpa-and-wpa2-wireless-encryption-and-why-it-matters/ [hämtat 05.04.2016]
[25] IT-ord. [Online]. http://it-ord.idg.se/ord/nonce/ .[hämtat 05.04.2016]
[Figur 1-3] Egna bilder