Risker och säkerhets-aspekter inom IoT

Hanna Sundkvist

Kandidatuppsats i datavetenskap

Handledare: Annamari Soini

Fakulteten för naturvetenskaper och teknik

Åbo Akademi, 2022

Referat

Idag lever människor i smarta miljöer, smarta hem och smarta städer. De har en stor mängd apparater i sin omgivning som samlar upp information omkring dem och om dem. Bland annat smarta armbandsklockor och applikationer på telefoner samlar in användardata när de är påslagna, och arbetar i bakgrunden utan att störa användarvänligheten. Det är viktigt att utomstående inte kan få tag på insamlad personlig information utan auktorisering. Man måste identifiera de brister som kan uppstå i IoT-nätverk på alla nivåer och i enlighet med detta hantera bristerna på ett kostnadseffektivt, men framgångsrikt, sätt för att kunna säkerställa hanteringen av känsliga data.

Sökord: IoT, säkerhet, smarta miljöer, nätverk

Innehållsförteckning

1. Inledning

Vartefter nya teknologier utvecklas skapas också nya smarta miljöer i människors omgivning. Sakernas internet (eng. *Internet of things*, IoT) refererar till så kallade ‘saker’ som kan kommunicera med andra ‘saker’ som ytterligare kan ta in information om sin omgivning eller en specifik faktor [1]. Sakernas internet är ämnat att integrera den fysiska omgivningen till olika kommunikationsnätverk, det vill säga en sammankoppling mellan det digitala och den verkliga världen [2,6]. Kevin Ashton nämnde sakernas internet för första gången år 1999 och termen har sedan dess fått flera definitioner och det har utvecklats olika applikationsområden [1]. Ett exempel på detta är hälsosektorn [3]. Enheter som smarttelefoner och hälsoapplikationer på smartklockor kan uppkopplas i IoT-nätverk och kommunicerar i bakgrunden med andra enheter. Det är viktigt att användardata inte faller i fel händer; följaktligen är det viktigt att förstå de risker som kan uppstå i IoT-nätverk och hur riskerna kan hanteras.

Antalet enheter världen över är för stort för att man skall kunna övervaka dem alla individuellt och således behöver enheter vara autonoma [4]. IoT-system är designade och uppbyggda enligt följande säkerhets principer: meddelandesekretess, dataintegritet, aktuella data, effektivitet, självstyrning, autentisering och tillgänglighet [4]. Exempel på säkerhetsrisker IoT-system utsätts för är överbelastningsattacker (eng. *Denial-of-service*, DOS), dataintrång och skadeprogram [2,4]. De flesta existerande säkerhetslösningar kräver omfattande beräkningar och skapar kommunikationsbelastningar; på grund av detta är speciellt billiga utomhusenheter känsliga för angrepp eller överbelastning [2].

Syftet med denna uppsats är att undersöka säkerhetsuppbyggnaden av IoT-system och de risker som kan uppstå i IoT-system, samt undersöka de säkerhetstekniker som används för att motverka risker i nuläget.

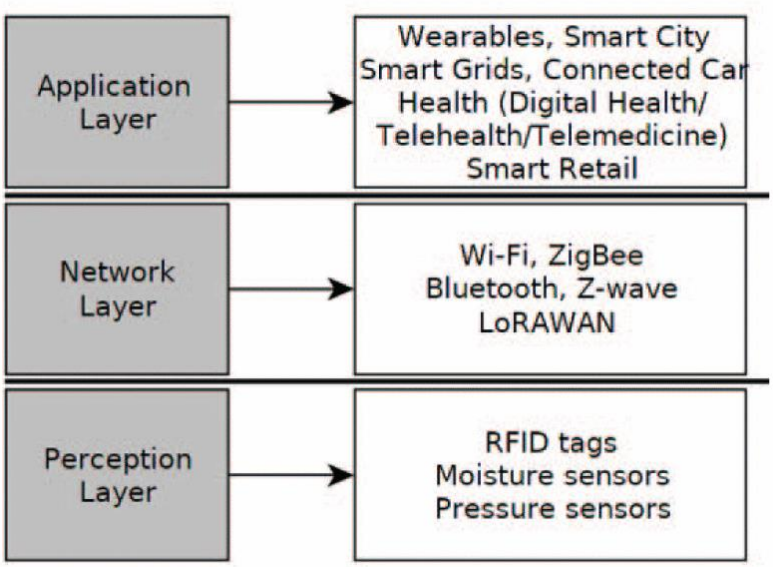
2. IoT-arkitektur och risker

2.1 Risker i IoT-arkitekturen

För att ett IoT-system ska fungera krävs kommunikation mellan enheter i ett nätverk. Radiofrekvensidentifikation (eng. *Radio Frequency Identification*, RFID) var till en början den dominerande kommunikationsteknologin, men man har senare introducerat trådlösa sensornätverk (eng. *Wireless Sensor Network*, WSN) [1]. RFID fungerar som en elektronisk identifikation som inte kräver någon egen energikälla, och använder en signal från läsaren för att identifiera sig till den RFID-läsare som avläser identifikation. WSN-använder sig av ett nätverk av sensorer som samlar in information från en miljö, distribuerar informationen och analyserar den. [3]

Säkerhetskrav för IoT-system är grundade i en önskan om datasekretess, integritet och förtroende hos användarna för systemet. Varje enhet som kopplas till nätverket och delar data skapar en potentiell säkerhetsbrist och vartefter enheter länkas till varandra och autonoma beslut fattas så ökar komplexiteten, vilket skapar en högre chans för kryphål i säkerheten och större krav på arkitekturen. [7]

Det finns inte en enhetlig modell för hur arkitekturen hos ett IoT-system skall avbildas men enligt Mohamed Litoussia, Nabil Kannoufb et al. består den mest förekommande modellen av tre lager perceptionslagret, applikationslagret och nätverkslagret [5]. Denna modell är avbildad i figur 1, där de tre lagren är avbildade som block med sina relaterade enheter. Enheterna kan enligt tre lager modellen interagera med det block, det vill säga det lager, som enheten är kopplad till. Exempelvis är perceptionslagret kopplat till många sensorer. En annan populär modell introducerar ett nytt lager ’supportlagret’ mellan applikationslagret och nätverkslagret, vilket lades till för att hantera auktorisering av användare och för hantering av säkerhetsrisker, samt för att sända vidare information till nätverkslagret. [6]



Figur 1: Tre lager IoT-arkitekturmodell [6]

2.1.1 Perceptionslagret

Perceptionslagret är det fysiska lagret som samlar informations om sin omgivning genom sensorer [5]. När man använder smarta apparater i en smart miljö samlar de in information om sin omgivning och om de som använder dem. Dessa apparater används i företag, hemmet och ute i vildmark; på grund av dessa miljöer där de mer eller mindre måste fungera autonomt och utan övervakning så kan de utsättas för bland annat manipuleras eller utsättas för förstörelse [4]. Fysiska apparater kan få sina kretsar ändrade, omprogrammeras, överbelastas med meningslösa data eller manipuleras genom att en måttlig mängd falska data matas in för att lura systemet [4].

Säkerhet måste byggas in i den fysiska enheten; den måste kunna identifiera sig själv samt andra enheter och kryptera konfidentiella data för att skydda privat information, men samtidigt även kunna interagera obehindrat med vanliga användare [7]. Neddragning av kostnader samt resurskrav är aktuellt när man designar enheter som kräver sensorer eller RFID, men billiga och ineffektiva beräkningsenheter kan leda till säkerhetsbrister [2].

Säkerhetsrisker i perceptionslagret består i huvudsak av fysiska hårdvaruattacker som är relaterade till RFID eller WSN sensorer [5]. Vanliga säkerhetsrisker i perceptionslagret är återuppspelningsattacker, tjuvlyssning och tidtagningsattacker [5].

2.1.2 Applikationslagret

Applikationslagret hanterar de applikations tjänster som erbjuds till användaren. Detta lager definierar även de olika sätt IoT-system kan tillämpas på, till exempel kan man tillämpa systemet för smarta miljöer som ett hem eller en stad [5]. För att hantera säkerheten i lagret krävs tillgång till säkerhetskonfiguration, som även är tillgänglig på distans, och säkerhets- samt mjukvaruuppdateringar [7].

Säkerhetsrisker som kan uppstå i detta lager är i huvudsak mjukvaruattacker och kan vara bland annat nätfiske, webbkodinjektion eller virus/masker [5].

2.1.3 Nätverkslagret

Nätverkslagret hanterar överföringen av data mellan enheter och servrar. Förutom detta processar det även data från perceptionslagret före överföring [5].

2.2 Överbelastningsattacker

Överbelastningsattacker fungerar genom att sabotören skickar en massiv mängd av förfrågningar, ofta meningslösa, till en service vilket leder till en stor mängd datatrafik som servicen i fråga inte kan hantera; detta leder i sin tur till fördröjningar i trafiken eller en total blockering av servicen för vanliga användare. [5] Överbelastningsattacker existerar dock i flera olika former och är några av de vanligaste samt enklaste typerna av attacker man kan utföra på IoT-system [4].

I perceptionslagret är "jamming" en vanlig typ av överbelastningsattack som skickar störningssignaler enligt ett intervall eller konstant. Målet är att sända störningssignaler som utnyttjar nätverkets bredband och IoT-enheternas minnes- och beräkningsresurser, när de upprepade gånger försöker kommunicera med varandra [2]. Antalet sabotörer behöver inte överskrida eller uppnå samma mängd som lagliga användare i nätverket, utan kan vara ett litet antal i jämförelse och ändå effektivt göra skada i ett nätverk. Applikationslagret är däremot mera mottagligt för sökvägsbaseradeöverbelastningsattacker.Sökvägsbaserade överbelastningsattacker utförs genom att sabotören skickar spontana eller tidsbundna paket i nätverket, som äter på nätverkets dataöverföring och resurser, vartefter energi och bandbredd utnyttjas till falsk nätverkstrafik. [4]

Två andra vanliga typer av överbelastningsattacker kallas för kollisionsattacker och felriktning. Kollisionsattacker är attacker där en sabotör har som avsikt att bryta etablerade kommunikationsprotokoll. I praktiken kan attacken betyda att en liten del av ett datapaket manipuleras och ändras, vilket leder till fel i kontrollsumman när man senare undersöker ifall paketet är korrekt.En felaktig kontrollsumma betyder att paketet eller paketen måste sändas på nytt ifall de påverkats av kollisionsattacken. Felriktning utförs genom att en sabotörs enhet inte dirigerar meddelanden i nätverket överhuvudtaget och det kan i sin tur leda till att delar av nätverk blir onåbara och kopplas bort. [4]

2.3 Sybilattacker

Sybilattacker är identitetsbaserade attacker. En sybilattack ämnar att sprida felaktiga eller medvetet snedvridna data genom att ta över nod i ett nätverk och sedan förvränga sin identitet; exempelvis kunde attackerna leda till negativ spridning av information och felaktiga data, som vid negativ förstärkningsinlärning för självstyrning genom maskininlärning, eller snedvridning av resultatet i en omröstning. En sabotör kan skada energikonsumtionen hos hela nätrverk när de kontinuerligt sänder felaktiga data till en IoT-enhet; många enheter använder sig av dataaggregation när de samlar in data för att minska energiförbrukningen men ifall datapaketen aggregerar tillräckliga falska data korrumperas hela datapaketet. [4]

I perceptionslagret sker sybilattacker genom fysisk manipulipation. Sabotören får sin falska identitet genom att hitta och manipulera en IoT-enhet som redan existerar i nätverket eller genom att tillverka en ny. I nätverkslagret kan sybilattacker däremot manipulera dirigeringscheman i nätverket. När det legitima noderna tar emot data från de falska noderna förvränger de nodernas dirigeringsscheman och dirigeringsbeslut som fattas i nätverket kan således misslyckas. [4]

3. Säkerhetslösningar inom IoT

3.1 Auktorisering och kryptering

För att kunna hantera risker inom IoT-system måste de som designar systemet implementera preventiva och aktiva säkerhetslösningar. När man skapar ett säkert system krävs autenticitet, integritet, åtkomstkontroll, säker avlastning och sekretess. [2,7]

Kryptering är en teknik som historiskt har använts för att förhindra att information faller i fel händer. Detta görs via ett chiffer, en algoritm, som omvandlar vanlig text till ett kryptogram vilket är oläsligt utan tillgång till en speciell nyckel som förhandlats om på förhand. Om en funktion kräver samma nyckel för kryptering och dekryptering kallas chiffret för symmetrisk kryptering och om det krävs olika nycklar för kryptering och dekryptering kallas det för asymmetrisk kryptering. Symmetrisk kryptering kräver mindre resurser än asymmetrisk kryptering, men att säkert distribuera nyckeln till alla rätta parter är ett problem vid symmetrisk kryptering [7]. Utöver symmetriska- och asymmetriska chiffer kallas ett chiffer som hanterar block av data för blockchiffer och chiffer som hanterar bitströmmar för strömchiffer. [4]

Kraven på autenticitet och integritet tillgodoses genom att utomstående inte har mixtrat med ett skickat meddelande och genom att källan kan verifieras. Detta hanteras via auktorisering; man kan använda sig av en sifferkod som på engelska kallas "*Message Authentication Code*" eller MAC. Auktorisering, exempelvis via MAC, förhindrar identitetsbaserade säkerhetsrisker som Sybil attacker genom att verifiera källan för ett meddelande [2]. MAC-koden fungerar som en ensidig hashfunktion där funktionen omvandlas till en MAC-kod genom symmetrisk kryptering och endast den som har nyckeln till funktionen kan verifiera hashkoden. En hashfunktion returnerar en bitsträng för ett block av data som man vill sända och ifall blocket har blivit manipulerat kan man urskilja detta från bitsträngen. [4] Säker avlastning gör det möjligt för IoT-enheter att utföra krävande beräkningar genom molnserver resurser [2]. Åtkomstkontroll ser till att de enheter som försöker

kommunicera i nätverket har tillåtelse att ansluta sig; samtidigt hanteras också sekretess vilket handlar om att data hålls säkert och tillgängligt för rätta parter. [7]

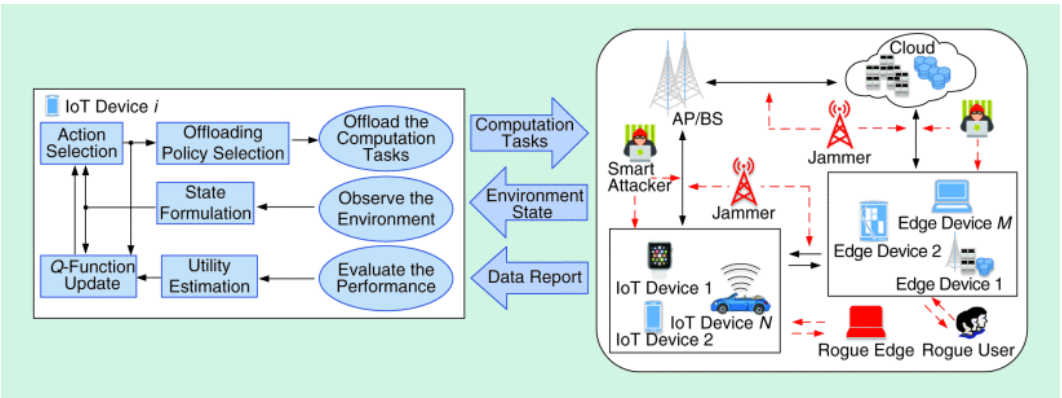
3.2 Maskininlärning

Det finns ett behov att utveckla IoT-enheters autonoma beslutfattningsprocess och det görs genom användningen av maskininlärning och artificiell intelligens. Maskininlärning använder tekniker som övervakad inlärning, oövervakad inlärning och förstärkningsinlärning för att förbättra auktorisering, åtkomstkontroll, säker avlastning och identifiering av skadeprogram. Övervakad inlärning inkluderar tekniker som neurala nätverk (NN), naive Bayes, Random forest och K-närmaste granne (K-NN). Oövervakad inlärning kräver inte data med klara etiketter utan sorterar efterhand data med liknande drag i grupper. Förstärkningsinlärning inkluderar tekniker som Dyna-Q och Q-inlärning. [2]

Eftersom en del beräkningar är så krävande och utförs med begränsade resurser är traditionella metoder inte tillräckliga för att förhindra identitetsbaserade attacker i auktoriseringsprocessen. Det fysiska lagret i IoT-arkitekturen kan använda radiokanaler och verktyg så som en indikator för ingående signalstyrka (eng. *Received Signal Strength Indicators*, RSSI) och MAC-adresser (eng. *Medium Access Control*, förkortas MAC liksom *Message Autentication Code* och referenser i denna uppsats kommer vara förkortade MAC-address) som ett skydd för att förhindra osäkra användardata. Den oövervakade tekniken IGMM kan hjälpa lokaliserade auktoriseringsprocesser. En modell som undersöks och beskrivs i Liang Xiao, Xiaoyue Wan et al. använder den oövervakade maskininlärningsmodellen IGMM, en icke-parametrisk Bayes-metod, för att undersöka RSSI och intervallen för ankomsten av radiodatapaket för att identifiera identitetsbaserade attacker. Processen består av att man kräver att den IoT-enhet på vilken testet utförs skall sända information som RSSI, MAC-addresser och paket intervall till den rätta mottagaren. Mottagaren använder sig därefter av IGMM för att jämföra mottagna data mot signaler i lokaliserade test. Ifall inga problem hittas tillhandahålls auktorisering från mottagaren till IoT-enheten. [2]

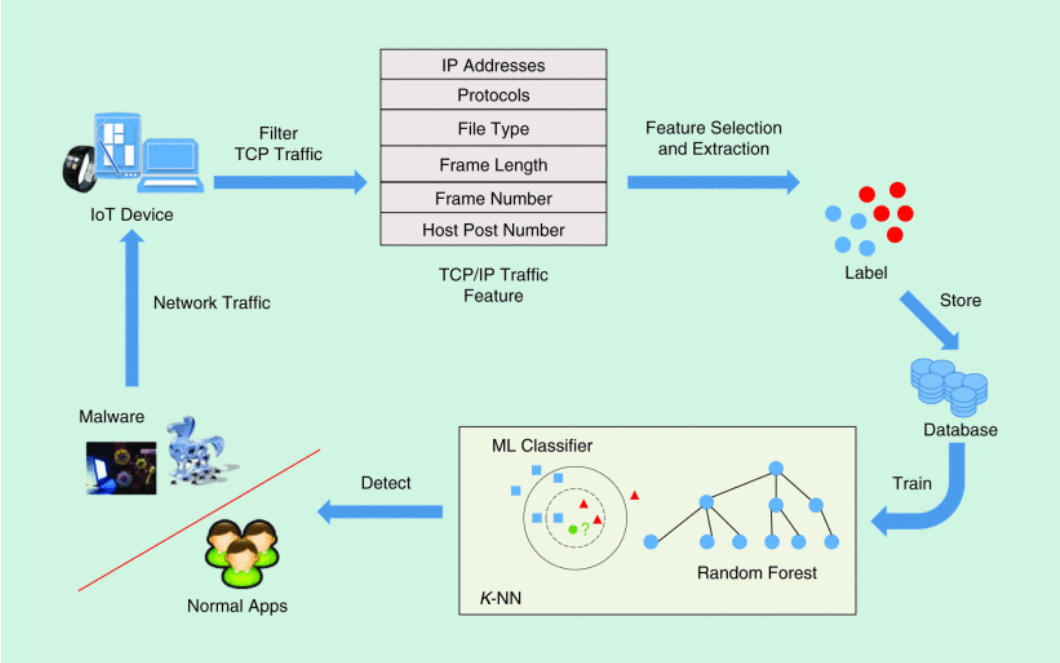
Säkra modeller för åtkomstkontroll i heterogena nätverk med många olika typer av noder och datakällor kan vara svåra att designa. Övervakade inlärningstekniker som K-NN, NN och SVM (eng. *Support Vector Machine*) används för att identifiera inkräktare i system. SVM kan exempelvis användas för att upptäcka olika typer av internettrafikattacker samt attacker på smarta nät. K-NN kan däremot användas för att spara på energiförbrukningen hos billiga utomhussensorer, vilka i normala fall kan ha bristfälliga resurser, genom att hjälpa definiera oövervakade avvikelser i WSN-sensorer. [2]

Säker avlastning i IoT-system handlar om att hantera attacker i perceptionslagret och MAC-adress attacker som till exempel tjuvlyssning, olika typer av störningar och falska enheter. Det framtida tillståndet hos en IoT-enhet är inte beroende av tidigare eller nuvarande tillstånd samt deras säkra avlastningsstrategier; detta betyder att man kan se på IoT-enhetens upprepade strategispel mot störningar och attacker som en Markov beslutsprocess (eng. *Markov Decision Process*, MDP) med finita tillstånd. Förstärkningsinlärningstekniker som Q-inlärning kan optimera detta strategispel eftersom denna teknik är praktisk att tillämpa med låg beräkningskomplexitet, detta är avbildat i figur 2. IoT-enheten formulerar sitt nuvarande tillstånd genom att observera störningar i nätet, radiobandbredd, kanalförstärkningar och hur viktig dess nuvarande uppgift är för att kunna utföra en säker avlastning. Q-inlärning drar nytta av tidigare erfarenhet och tillstånd för att göra en lämplig bedömning av att förbättra den långsiktiga optimeringen och kartläggning utåt för att inte bli instängd i endast den lokala optimala strategin. Värdena för Q-inlärning uppdateras genom en iterativ Bellman-funktion för varje avlastningsstrategi och det dåvarande nätverkstillståndet. [2]



Figur 2: Illustration av en tillämpning av Q-inlärning för att optimera säker avlastning [2]

Man kan använda övervakad inlärning för att göra en diagnos av exekveringen av olika program. Random forest och K-NN kan användas för att skapa en modell med vilken man kan upptäcka skadeprogram, vilket illustreras i figur 3. En IoT-enhet filtrerar enligt denna modell igenom särdrag hos det datapaket som sänds, exempelvis för IP-adress och datablockens nummer samt längd, vilka sedan kategoriseras och lagras i modellens databas. Modellen överlåter därefter nätverkstrafiken till den K-NN klass som innehåller flest objekt, sedan byggs ett beslutsträd enligt Random forest med hjälp av data om nätverkstrafiken från lagrade nätverket för att kunna identifiera skadeprogram. Säkerhetsserver med större skadeprogram databaser och minne kan erbjuda säkrare applikationer genom att säkerheten överförs till en server. [2]



Figur 3: En illustration av hur K-NN och Random forest används för att identifiera skadeprogram. [2]

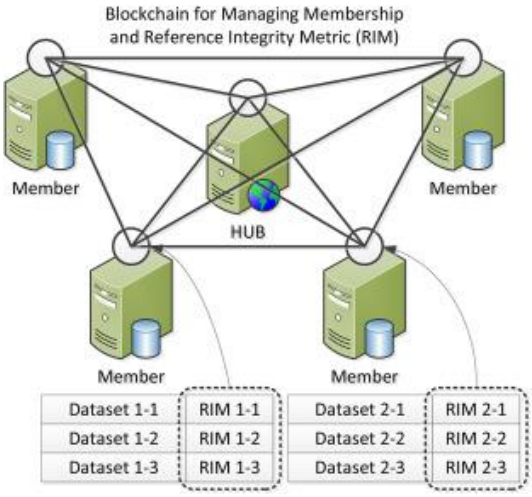
3.3 Blockkedja

För att öka säkerheten hos IoT-nätverk och enheter samt för att maximera undersökning inom IoT-säkerhet krävs bättre tillgång till data insamlad i den reella världen, eftersom det krävs både tid och möda vid insamlingen av data i värld med diverse miljöer, nätverk och enheter. Det finns fördelar med att ha möjligheten till olika datamängder att jämföra dataformat och datastrukturer inom samma typer av system vid utvecklingen av systemsäkerhet. Det ska vara möjligt att distribuera datamängder till allmänheten utan att privata data läcker ut till offentligheten. Blockkedjor är en av de metoder som har lyfts fram för att möjliggöra säker distribuering och bevara autenticiteten av data. [8]

Blockkedjor har tidigare använts för att säkerställa och notera finansiella transaktioner mellan kryptovalutor. Fördelarna med blockkedjor är att det finns ett tydligt register över transaktionerna, med vilka man kan identifiera och spåra eventuella modifikationer i transaktionerna. I en typisk blockkedja skapas när den första transaktionen sker, därefter skapas ett block närsomhelst en transaktion görs, efter det sänder blocket ut en signal till alla noder i nätverket. En av noderna behöver bekräfta blocket och skicka bekräftelsen tillbaka i nätverket. Ifall blocket kan verifieras utan problem läggs det till i en kedja av block, som alla har en referens till ett tidigare block (förutom startblocket). [8]

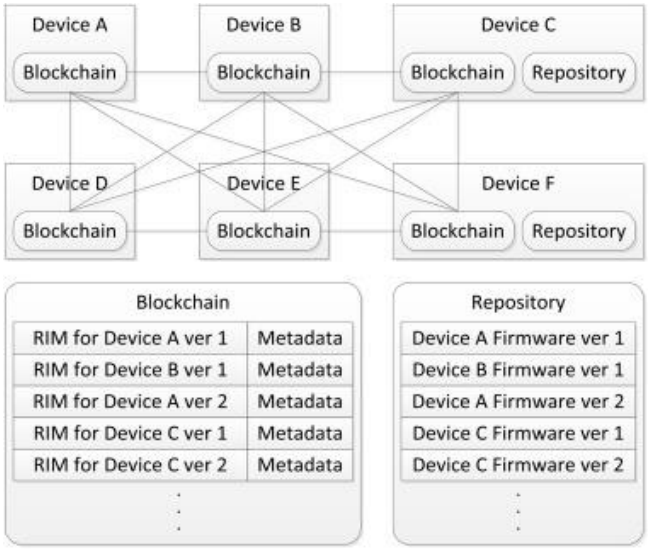
I sin artikel beskriver M. Banerjee,' J. Lee och KKR. Choo två blockkedje-exempel för användning inom IoT-säkerhet. Det första modellexemplet behandlar distribueringen av datamängder med hjälp av blockkedjor. För att verifiera datablock samt motverka felaktigheter och manipulation använder sig modellen av ett integritetsmått, eng. *Reference Integrity Metric* (RIM), som upprätthålls av blockkedjan. I modellen använder man sig av en central hubb som har hand om en lista av referenser till var nätverkets noder lagrar sina data och vart data distribueras. Blockkedjan håller reda på adresser, ägare och distribueringsinformation, på detta sätt har alla noder tillgång till informationen. Utöver detta existerar sekundära block som upprätthåller autenciteten av datamängderna genom integritetsmåttet - se figur 4. Ett problem är att transaktioner i blockkedjan är permanenta och inte kan tas bort, således krävs det att endast identitetsmåttet upprätthålls av en blockkedja. Ifall

ägaren vill sluta dela datamängden förblir endast integritetsmåttet i blockkedjan. Dock krävs det ännu att datamängder anonymiseras före de distribueras. [8]



Figur 4: En illustration över blockkedjor med integritetsmått (RIM) för distribution av datamängder. [8]

Det andra blockkedje-exemplet hanterar upptäckten av manipulerade inbyggda program och enheter, vilket illustreras i figur 5. Eftersom säkerhet aldrig kan garanteras till hundra procent krävs det ett sätt för utsatta enheter att kunna fixa sig själva. Tidigare skydd för manipulering av inbyggda program är baserade på att kunna undersöka integriteten av operativsystemet och de applikationer som systemet kör. Integritet undersöks via integritetsmåttet, vilket är beräknat på förhand för operativsystemet samt applikationer och sedan lagrad för framtida användning. När en applikation ska köras beräknas integriteten och jämförs med det lagrade värdet. Blockkedjan används även i denna modell för att bevara intregitetsmåttet, men denna gång är blockkedjan en distribuerad databas som noterar alla transaktioner mellan enheter. Säkerheten är bevarad eftersom det skulle kräva att majoriteten av alla enheter skulle utsättas för manipulation innan integriteten av systemet är äventyrad. Felaktiga data kan ersättas med en tidigare version eller liknande data som bevaras tack vare blockkedjan. Säkerheten kan därmed ökas genom att behålla tidigare versioner i lager för enheter i nätverket. [8]



Figur 5: En illustration av en blockkedja för bevaring av integritet i ett system av enheter länkade till varandra. [8]

3.3 SDN

I takten med att teknik utvecklas ökar kraven på kompabilitet och skalbarhet, trots att det existerar många olikartade typer av enheter. För att utveckla IoT-system för smarta miljöer behöver man därmed beakta kompabilitet och skalbarheten av enheter, sensorer har exempelvis mindre resurser, i en miljö, vilket ökar komplexiteten av systemutvecklingen och säkerheten hos IoT-nätverk. Mjukvarudefinierat nätverk (eng. *Software Defined Networking*, SDN) är en ny nätverkssäkerhet hantering modell som har blivit populär i ett flertal områden, bland annat företagssäkerhet, smarta miljöer och elektroniska hälsosystem [5]. Särdrag hos SDN är att arkitekturen tillåter en mer dynamisk och agil miljö. SDN separerar kontroll och dataplaner, i vanliga fall är dataplanet och controllern sammankopplade [5,9]. [9]

SDN-controllern hanterar all beslutsfattning medan dataväxlar sänder data framåt i nätverket. På grund av att beslut inte behöver fattas av nätverksenheter, så kan man undvika användningen av komplexa nätverksrouters och detta leder till inbesparingar på nätverksenheter Den förenklade komplexiteten tillåter även SDN-arkitekturen att enklare hantera problem som uppstår hos traditionella routersystem,

det vill säga säkerhetsuppdateringar, sändning av data och fel i länkar mellan enheterna. I vanliga nätverk krävs det att en enhet skall hantera uppdateringar själva, men SDN-nätverk hanterar uppdateringar för enheterna. [9]

SDN har varit en attraktiv nätverkslösning för forskning inom IoT-säkerhet. Lösningarna kan kategoriseras som nätverksbaserade-, trafikbaserade- och kryptobaseradesäkerhetslösningar. Nätverksbaserade säkerhetslösningar hanterar däremot modeller som har att göra med arkitekturen av SDN-nätverk. Trafikbaseradelösningar observerar hur data skickas fram och tillbaka i nätet för att känna igen sabotörer och illdåd. Lösningarna baserar identifieringen på insamlade data om nätverkstrafiken. Kryptobaserad säkerhet lägger i stället fokus på miljöns kryptografiska egenskaper, exempelvis svarta nätverk som krypterar alla datapaket som sänds på nätverket eller indentifikationssystem. [9]

4. Sammanfattning

Källförteckning

[1] Rajkumar Buyya and Amir Vahid Dastjerdi, Internet of Things: Principles and Paradigms. Elsevier Inc, 2016.

[2] Liang Xiao, Xiaoyue Wan, Xiaozhen Lu, Yanyong Zhang and Di Wu, IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security? IEEE Signal Processing Magazine, Vol. 35, Issue 5, 2018, pp 41 - 49.

[3] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic and Marimuthu Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, Vol. 29, Issue 7, 2013, pp 1645–1660.

[4] Vlasios Tsiatsis, Stamatis Karnouskos, Jan Höller, David Boyle and Catherine Mulligan, Internet of Things: Technologies and Applications for a New Age of Intelligence (2nd ed.). Elsevier Ltd, 2019.

[6] Nickson M. Karie, Nor Masri Sahri, Paul Haskell-Dowland, IoT Threat Detection Advances, Challenges and Future Directions. In Workshop on Emerging Technologies for Security in IoT (ETSecIoT), Sydney, 2020, pp. 22-29.

[7] Shancang Li and Li Da Xu, Securing the Internet of Things. Elsevier Inc, 2017.

[8] M. Banerjee,' J. Lee and KKR Choo, A blockchain future for internet of things security: a position paper. Digital Communications and Networks, Vol. 4, Issue 3, 2018, pp 149-160.

[9] K. Kalkan, S. Zeadally. Securing Internet of Things with Software Defined Networking. IEEE Communications Magazine, Vol. 56, Issue: 9, 2018, pp 186 – 192.