Eksamensprojekt

Sikkerhed i Z-Wave systemer

Institut for Matematisk Modellering (IMM)

Danmarks Tekniske Universitet (DTU)

Eksamensprojekt er udført af:
Rasmus Christiansen, s971458
Yavuz Seker, s992477

Vejleder på DTU
Christian D. Jensen

Skrives i samarbejde med:
Zensys A/S
Emdrupvej 26
2100 København Ø

Vejleder på Zensys A/S
Jørgen Franck

Projekt Start og Slut:
2. Februar 2004 – 2. August 2004

Forord

Denne rapport er et Polytekniskeksamensprojekt, skrevet ved Informatik og Matematisk Modelering på Danmarks Tekniske Universitet i foråret 2004 og er et samarbejde mellem *Rasmus Christiansen* og *Yavuz Seker*. Denne projektrapport tager udgangspunkt i sikkerheden af Z-Wave teknologien, hvilket er en RF baseret trådløs protokol. I projektet diskuteres og analyseres to sikkerhedsproblemer i Z-Wave netværket, nemlig initial nøgleudveksling og forhindring af replay attacks. Det forventes at læseren har basalt kendskab til trådløs kommunikation, IT-sikkerhed og kryptografi for at forstå indholdet af denne rapport.

Z-Wave teknologien er udviklet af firmaet Zensys A/S og bruges af firmaet til kommunikation mellem de enkelte enheder i et trådløst netværk. Denne teknologi bliver brugt til trådløs kontrol af lys og el-nettet i private hjem, derfor undersøges her i rapporten sikkerheden af denne trådløse teknologi. Der kigges på nøgleudvekslingen mellem de enkelte enheder, der optræder i netværket og på replay attack i netværket, og der defineres og analyseres løsningsforslag til de to begreber.

Vi har fundet frem til en del løsninger for nøgleudvekslingen, som f.eks. kan ske vha. en ledning eller ved at man bruger flerfarvede lysdioder. Et andet eksempel er, at gøre brug af Smart Card til udveksling af nøglen.

Til forebyggelse af replay attacks har vi kigget på løsninger som gør brug af Challange/Response teknik, Sekvensnumre og Time Stamps. Vi har her i rapporten beskrevet, hvordan disse løsninger generelt virker i forskellige systemer og i Z-Wave. Vi har så fundet frem til, at Time Stamps løsningen er den mest relevante løsning mod replay attacks i Z-Wave.

De ovenfor nævnte løsninger samt en del andre løsninger er beskrevet her i rapporten, og efter evaluering er den bedste løsning fundet med hensyn til pris, brugervenlighed og ønske fra Zensys A/S.

Vi har med denne rapport bidraget til, at det er muligt at bruge dette trådløse smart hus system i ens private hjem og virksomheder uden, at der er nogen risiko for at ens systems sikkerhed bliver truet. Vi har med vores idéer og løsninger kommet frem til, at det er muligt at gennemføre en

nøgleudveksling mellem de forskellige enheder i det trådløse system, som gør det svært for angribere at bryde systemet. Vi har også kommet med idéer til forbedring af undgåelse af replay attacks i systemet.

Vi vil gerne takke vores vejleder, lektor *Christian D. Jensen* , Informatik og Matematisk Modellering, DTU, vejleder Jørgen Franck, Zensys A/S, og alle Zensys A/S medarbejdere, der har været os behjælpelige under hele projektets forløb.

Abstract

This report covers two important security issues namely the key exchange and replay attacks of the wireless technology Z-Wave. Z-Wave is a RF (Radio Frequency) based wireless technology, which is developed by Zensys A/S (www.zen-sys.com) and is used to communicate between Z-Wave enabled devices in a wireless network. The Z-Wave technology is for example used for wireless light-control and other power driven applications in private houses and offices. We investigate the security of this wireless network. We examined the key exchange between the devices in a network and on the possibility of executing a replay attack against nodes in the network. We propose a couple of solutions for how key exchange can be done and how replay attacks can be avoided in the Z-Wave network.

We have found twenty-one solutions for key exchange between devices. A description of our proposals is found in this report. Three examples of such solutions are cable, smart card and multicolor diodes. In this report we will discuss each of the solutions and arrive with some suggestions for the optimal solution to this problem. To avoid replay attacks against the network have we analyzed three possible methods to address this problem. They are Challenge/Response, Sequence Numbers and Time Stamps. In this report, we will talk about how these methods are used in application to replay attacks.

We believe that our work will be particularly useful for securing the wireless network Z-Wave. If one or more of the solution proposals for key exchange and replay attacks is being used for the network, the communication between the devices will be more secure, and the replay attacks against the network can be avoid.

This thesis is a Master thesis for two Master of Science students, *Rasmus Christiansen*, Electronic Science and *Yavuz Seker*, Computer Science, from the Technical University of Denmark (DTU). This research work is done for the institute of Informatics and Mathematical Modelling (IMM) at DTU in conjunction with company Zensys A/S. We would like to thank our two supervisors, Associate Professor *Christian D. Jensen* (IMM) and software manager *Jørgen Franck (*Zensys A/S) for all their help during the project.

Indholdsfortegnelse

1.0 Indledning

Som titlen lægger op til, har vi i denne rapport valgt at fokusere på sikkerheden i Z-Wave teknologien. Vi har lagt fokus på sikkerheden af nøgleudveksling og replay attacks i Z-Wave teknologien, da det er denne protokol, der bliver anvendt til trådløs kontrol af lys og el-nettet i private hjem. Protokollen skal sikre at brugerne sikkert kan kontrollere alle de enheder[1] som optræder i deres trådløse netværk uden at blive forstyrret af andre. Da trådløs styring af lys og el-net i private hjem er ved at blive mere og mere udbredt, så skal der findes løsninger så man ikke bliver forstyrret af sin nabo, når han/hun prøver at tænde/slukke for sit lys men i stedet for tænder/slukker sit eget lys. Et andet eksempel kan være alarmsystemet. Det må ikke være muligt for naboen eller tyv, at styre ens alarm. Problemstillingen er, hvor sikkert kan et Z-Wave netværk være? Hvordan kan man undgå de ovenfor beskrevne problemer?

For at undgå at en anden person leger med ens el-net derhjemme, undersøges først og fremmest nøgleudvekslingen mellem de forskellige enheder i netværket. Nøgleudvekslingen skal sikre, at ingen andre end én selv kan styre sit netværk med andre enheder. I rapporten beskrives og analyseres nogle løsninger til nøgleudvekslingen mellem enhederne, der findes også en beskrivelse af fordele og ulemper ved disse løsninger, så den mest optimale løsning til nøgleudveksling kan bestemmes.

Endvidere undersøges replay attacks i et Z-Wave trådløst netværk. Der bliver først defineret hvad replay attacks er, derefter defineres generelle metoder til at undgå replay attacks og til sidst undersøges de nyttige metoder til at undgå replay attacks i Z-Wave netværket. Her i rapporten findes beskrivelse og analyse af replay attacks.

Gennem en evaluering bestemmes de bedste løsninger mht. nøgleudveksling og hvordan man modvirker replay attacks i Z-Wave netværket.

[1] Enhed betegner de forskellige apparater der optræder i Z-Wave trådløse netværk.

1.1 Problemstilling

Zensys er et relativt nystartet (1999) firma på omkring 35 mand, som beskæftiger sig med trådløs 'Intelligent Home Control'. Zensys har udviklet en stabil low-cost kontrol og overvågningsteknologi baseret på RF til kommunikation mellem de enkelte enheder i et netværk. Zensys' produkter kan bruges i hjemmet til f.eks.:

- Lyskontrol
- Sensorer f.eks. termometer
- Sikkerhed/Overvågning (rumfølere)
- Indendørs klimaanlæg (ventilation, køling, ...)
- Døre og porte
- Persienner og gardiner
- m.m.

Zensys har udviklet en proprietær trådløs kommunikationsprotokol Z-Wave samt en ASIC med integreret MCU[2] og RF tranceiver. Det primære marked er den private forbruger, hvorfor en lav pris er vigtig for at kunne komme ind på/få dannet et marked.

Der ønskes foretaget en analyse af visse elementer i en Wireless Security Solution til Z-Wave baseret på ZW0102 ASIC'en, hvor der er begrænsede ressourcer til rådighed.

De vigtige elementer, der skal undersøges i projektet er:

- Initial nøgleudveksling.
- Forhindring af replay attacks.

Det er vigtigt, at der undgås anvendelse af eksterne komponenter for at fastholde en lav kostpris.

De mulige metoder gennemgås med hensyn til fordele/ulemper som f.eks. krav til RAM og Flash, respons tider o. lign. For konsekvenser for applikationen i øvrigt henvises der til 'Z-Wave Security Requirement Specification'.

[2] MCU: Micro Controller Unit

Opnåelse med projektet er:

- Analyse af problemstillingen
- Design af løsninger

Evaluering af det foreslåede design enten gennem analyse af protokoller, eller gennem forsøg på ZW0102 ASIC'en eller andre platforme.

1.2 Zensys A/S

Firmaet Zensys A/S er stiftet i 1999 og er en dansk virksomhed med fokus på udvikling og markedsføring af produkter til "det intelligente hjem". Zensys har udviklet deres egen patenteret trådløse teknologi kaldet Z-Wave protokol, som er baseret på RF. Denne teknologi brev integreret i et produkt af Zensys til trådløs kontrol af lys og el-nettet i private hjem. Det første produkt er netop blevet sendt på gaden i USA. Det skruer automatisk ned for lyset, op for ventilationsanlægget og trækker gardinerne for, hvis kunden sætter en DVD-film i sin afspiller og tænder via sin lommecomputer (PDA).

Zensys satser på at trænge ind på markedet ved at levere et avanceret produkt billigere end de konkurrenter, som primært har fokuseret på eksisterende el-kabel teknologi, der er væsentligt dyrere. Denne lavprisstrategi realiseres blandt andet ved, at produktionen af de fysiske produkter er udlagt til Thailand. Produktudvikling og markedsføring foregår fra Danmark, hvor Zensys beskæftiger ca. 35 medarbejdere. Virksomhedens stiftere har en teknisk og markedsmæssig baggrund blandt andet fra den danske IT-virksomhed Olicom. Yderligere har Zensys opbygget et datterselskab i San Francisco.

Virksomheden har netop fået det blå stempel af verdens største chipproducent, Intel, der har valgt Zensys som leverandør af chip til styring af elektriske små apparater i fremtidens intelligente hjem.
Intel og Zensys skal sammen bygge bro mellem cirka 60 små elektriske komponenter i hjemmet og Microsoft Universal Plug-n-Play standard, så ethvert elektronisk apparat kan styres via pc'en eller mobiltelefonen/PDA´en.

Dermed bliver Zensys den første og indtil videre eneste leverandør af netværk til de små elektroniske apparater i det intelligente hjem, der understøtter Plug-n-Play. Da alle de store leverandører af forbrugerelektronik, såsom Samsung, Sony, Microsoft og Nokia anvender Microsoft's Plug-n-Play standard, forventer Zensys, at den danske teknologi vil blive indbygget i en lang række andre forbruger elektroniske produkter i fremtiden.

Indtil videre har Zensys Developer Kit kontrakter med omkring 100 teknologiselskaber, hvor de største officielle kunder er Intermatic (US), Danfoss og ICOM (Holland)

1.3 Projekt

Som sagt sker styringen af lys og el-nettet i private hjem og kontorer trådløst og efterhånden som det bliver mere og mere udbredt forventes det, at der skal være en god sikkerhed i det trådløse netværk, så ingen andre end en selv har kontrol over styring af sit hjem. Da der ikke findes 100% sikre netværk, så skal sikkerhedsbegreberne undersøges for at sikre netværket og her i rapporten gennemgås to vigtige sikkerhedsbegreber, nemlig initial nøgleudveksling og replay attacks.

For at undgå at en anden person leger med ens el-net derhjemme eller på kontoret undersøges først og fremmes nøgleudvekslingen mellem de forskellige enheder i netværket. Nøgleudvekslingen skal sikre, at ingen andre end en selv kan styre sit netværk med andre enheder.

I rapporten beskrives og analyseres løsninger til nøgleudvekslingen mellem enhederne det trådløse netværk. De fundende løsninger der findes til den initiale nøgleudveksling, evalueres mht. fordele og ulemper, så vi dermed kan bestemme de bedste løsninger til nøgleudvekslingen.

Endvidere undersøges replay attacks i sådan et trådløst netværk. Der bliver først beskrevet generelle metoder til at undgå replay attacks i et netværk. Derefter analyseres løsningsmetoder til at undgå/modvirke replay attacks i Z-Wave netværket. De fundende løsningsmetoder evalueres mht. fordele og ulemper, og den bedste løsning til at undgå/modvirke replay attacks bestemmes.

1.4 Status

Som sagt har vi i dette projekt valgt at kigge på to vigtige sikkerhedsbegreber inden for sikkerhed af Home automation – Intelligent huse, nemlig;

- Initial nøgleudvekslingen mellem noderne i et Z-Wave netværk
- og metoder til at modvirke replay attacks mod et Z-Wave netværk.

I rapporten er vi kommet med en hel del løsningsforslag til initial nøgleudveksling mellem noderne i et Z-Wave netværk. Vi har beskrevet løsningsforslag som;

- RF mht. software, lav sendestyrke og retningsbestemt
- Forudprogrammeret
- Ledning
- SIL/DIL kontakter
- Flere trykknapper
- Tastatur
- Drejeomskifter
- Trimmer
- Farvet lysdioder
- 7-segment
- Smart Card
- Fingeraftryk
- Stregkode
- Magnetkort
- IR
- Laser
- Ultralyd
- PAN (Personal Area Network)

Alle disse løsninger er blevet evalueret med hensyn til deres fordele og ulemper, så vi til sidst kunne bestemme den bedste løsningsmetode for initial nøgleudveksling mellem noderne i et Z-Wave netværk. Endvidere er eksperimentielle forsøg lavet for at finde ud af hvor brugervenlige løsninger er samt, hvor lang tid det tager at foretage en inkludering.

Vi har også analyseret nogle løsningsmetoder til at undgå/modvirke replay attacks mod Z-Wave netværket. Vi har beskrevet løsninger som;

- Challenge/Response
- Sekvens Numbers
- Time Stamps
- En kombination af Time Stamps løsning og Challenge/Response løsning.

Disse løsninger er blevet evalueret med hensyn til deres fordele og ulemper, så vi til sidst kunne bestemme den bedste løsningsmetode for at undgå/modvirke replay attacks.

Vi har også kigget på, hvor tit nøglerne i et Z-Wave netværk skal skiftes. Ved at kigge på Unicity distance og nogle kendte tider for at knække DES, fik vi bestemt hvor tit nøglerne i Z-Wave netværk skal skiftes.

1.5 Rapportens Struktur

I starten af rapporten er det muligt at finde et forord og abstract, der giver en beskrivelse af projektet på dansk og på engelsk. Efter forord findes de seks kapitler denne rapport består af, nemlig:

- **1.0 Indledning:**
  Giver en beskrivelse af projektets formål og mål, en dybere beskrivelse af problemstillingerne i projektet, en beskrivelse af firmaet Zensys A/S, beskrivelse af projektet, status over projektet og rapportens struktur.
- **2.0 State of the art:**
  Her bliver beskrevet punkter som; Home automation, netværk, trådløst netværk, trådløse teknologier som IRDA, 802.11x, Bluetooth, RFID, Sensor netværk og Z-Wave, som er udgangspunkt i hele projektet. Dette afsnit ender med en opsummering. Kapitlet indeholder endvidere et afsnit om sikkerhed, hvor der bl.a. gives en beskrivelse af autentifikation, kryptering samt angrebsmetoder. Det er også muligt at finde en opsummeringer for afsnittet om sikkerhed.
- **3.0 Design:**
  Her beskrives først karakteristika for secure Z-Wave. Derefter beskrives de forskellige løsningsforslag til, hvordan den initiale nøgleudveksling kan finde sted i netværket og løsningsforslag til, hvordan man kan undgå replay attacks mod netværket. Der findes, i dette kapitel, en opsummering af initial nøgleudveksling og en opsummering af replay attacks.
- **4.0 Evaluering:**
  I dette kapitel findes en evaluering af de forskellige løsningsforslag der er defineret til både nøgleudveksling og replay attacks. Gennem denne evaluering bestemmes de/den bedste løsning(er) til nøgleudveksling og til undgåelse af replay attacks.
- **5.0 Konklusion:**
  En samlet konklusion til projektet findes i dette kapital.
- **6.0 Bilag:**
  Til sidst, i kapitel seks findes bilagene.

2.0 State of the art

I dette afsnit defineres og gennemgås nogle begreber inden for netværk og sikkerhed, som hjælper med at forstå problemstillingen og løsningerne til problemstillingerne. Først beskrives kort, hvad Home Automation, er, som giver et overblik over hvad opgaven generelt handler om.. Derefter beskrives netværk og trådløst netværk, som giver et overblik over, hvilken form for netværk der arbejdes med. Nogle trådløse kommunikationsteknologier beskrives. Det er sig IRDA, Bluetooth, RFID, IEEE 802.11x samt Zensys' Z-Wave teknologi. Sidste afsnit handler om sikkerhed. Her beskrives nogle begreber indenfor kryptografi og angrebsmetoder. Alt i alt hjælper dette afsnit med forskellige grundlæggende begreber til at forstå projektets indhold.

2.1 Home Automation

Begrebet Home Automation spænder over et stort område, som eksempler kan nævnes den simple fjernkontrol, automatisk styring af lys, sikkerhedssystemer med bevægelsessensorer samt avancerede kontrollere med stemmegenkendelse. Basalt set betegner man Home Automation som ting, der har med fjernstyring eller automatisering af udstyr i og omkring hjemmet. Meningen er at gøre livet i og omkring hjemmet nemmere og mere sikkert. Nedenfor beskrives kort de mest populære applikationer:

Den mest populære kategori inden for Home Automation er lysstyring. Her kan brugeren styre lyset ved hjælp af en fjernbetjening, via programmeret kontakter eller via en computer/Internettet. Inden for dette område er LK en af de førende i Europa med deres IHC system[3].

En anden meget populær kategori er sikkerhed og kontrol systemer. Dette dækker over bevægelsessensorer, vindue- og dørfølere samt røg- og gasdetektorer. Disse kan f.eks. give melding til en mobiltelefon, hvis en af dem aktiveres. Et andet eksempel er garagedøren, der går op, når en bil kommer ind ad indkørelsen.[4]

Næste kategori er audio og video. Af eksempler kan nævnes at lyden fra f.eks. en radiokanal følger personen/-erne rundt i huset. Ved at placere højtalere og rumfølere i alle lokaler tændes højtalerne i det/de rum man befinder sig i. En anden ting er trådløs overførelse af audio og video fra f.eks. computer eller DVD-afspiller til hjemmebiografen. Et scenarium er at projektionslærredet ruller ned, forstærker og projektor tændes og gardinerne trækker for, så snart der sættes en DVD i afspilleren. Dette er blevet meget populært i USA og er så småt på vej i Europa.

Den sidste kategori der gennemgås her er klimastyring. Konceptet er her, at man kan styre klimaet samt aflæse diverse sensorer i og omkring hjemmet. Det er f.eks. muligt at tænde og slukke for varmen eller sætte sprinklerne på græsplænen i gang i sit sommerhus via Internettet[5].

[3] Kilde: http://www.lk.dk/public/s_gen1.asp?what=tekst&sideid=192
[4] For aktuelle US produkter se f.eks.: http://www.smarthome.com/secvehicle.html
[5] Kilde: http://www.computerworld.dk/default.asp?Mode=2&ArticleID=23946

2.2 Netværk

Et netværk består af to eller flere enheder (f.eks. computere), som er koblet sammen enten via kabler, trådløst kommunikation eller modemer, for at udveksle data.

Et netværk består af tre dele:

- Transmissions medie (ledning, kabler, ...)
- Hardware enheder (switches, router, …)
- Software komponenter (protokol stakke, drivere, …)

Overordnet set findes to slags netværk:

- LAN (Local Area Network) fungerer i et geografisk begrænset område, f.eks. i bygninger eller etager. Netværket bruges både til private personer samt virksomheder.
- WAN (Wide Area Network) er et netværk som er sammensat af flere LAN. Det bruges primært af store virksomheder eller offentlige institutioner, hvor der er behov for national eller international dataoverførelse. Internettet er et eksempel for WAN.

For at enhederne kan kommunikere med hinanden, skal de følge en protokol i netværket. En protokol er i et netværkssprog et sæt regler for, hvordan enheder kan kommunikere med hinanden. Forskellige netværksprotokoller har hver sin kommunikationsmetode.

Open System Interconnection[6] (OSI) er en standard, som bestemmer rammen for implementering af netværksprotokoller og defineres af ISO[7]. OSI-standarder beskriver f.eks. hvad der bliver placeret på et netværkskabel, hvornår og hvordan det bliver placeret der. De definerer derimod ikke den enkelte transmissionsmetode eller protokol.

[6] Kilde: http://libra.unitbv.ro/internet/OSI%20model.htm
[7] ISO: International Organization for Standardization (http://www.iso.org)

I OSI-rammen opdeles netværkskommunikationen i syv lag[8]:

- **Lag 1, Fysiske lag:** Dette er den måde hvorpå information bliver overført. Gennem fiberkabel, radio, laser m.m. På dette lag beskrives specifikationer for det fysiske medium, som signalerne bevæger sig igennem.
- **Lag 2, Datalink lag:** Her tages højde for svagheder ved det fysiske medium. Der tilføjes informationer der bruges til at sikre mod forvanskning af data. Hvis der kan være flere end to computere på samme fysiske netværk, tilføjes der også information om afsender og modtager.
- **Lag 3, Netværks lag:** Dette lag bestemmer dataveje i systemer med flere fysiske netværk. Her kontrolleres de sendte meddelelser mellem enhederne.
- **Lag 4, Transport lag:** Sørger for at data som er blevet skilt ad og sendt via flere forskellige veje gennem nettene, igen bliver samlet i den rigtige rækkefølge. Her sikres det også, at data ikke går tabt undervejs. En dataforbindelse, som går gennem mange computere, har kun et transport lag i enderne af forbindelsen.
- **Lag 5, Sessions lag:** Holder styr på hvis tur det er til at sende, og om forbindelsen er åben eller ej.
- **Lag 6, Præsentations lag:** Oversætter dataformater, hvis de to kommunikerende computere ikke bruger samme tegnsæt, byterækkefølge eller på anden måde opbygger informationerne forskelligt.
- **Lag 7, Applikations lag:** Dette lag indeholder de informationer/data, som applikationen hos den oprindelige afsender og modtager skal bruge.

I afsnit 2.2.1.6 ”Z-Wave” vises en lignende oversigt over, hvordan Z-Wave protokollen er opbygget af lag med en oversigt, som viser sammenhængen mellem de to modeller.

Der findes mange forskellige slags netværk og disse kan overordnet set karakteriseres som værende trådløst eller ikke-trådløst. Da Z-Wave netværket er rent trådløst, vil vi nu se på nogle af de mest benyttede trådløse netværksteknologier der findes. Dette gøres for, at skabe overblik over fordele og ulemper samt, hvor deres primære anvendelsesområde ligger.

[8] Kilde: http://libra.unitbv.ro/internet/OSI%20model.htm

2.2.1 Trådløse netværk

Et trådløst netværk er et datanetværk, som bruger radio- eller lysbølger for at flere enheder kan kommunikere med hinanden, dvs. der er ingen fysisk kontakt mellem enhederne. Der er fire fordele ved trådløs kommunikation:

- ***Mobilitet:*** Brugere kan koble sig til netværket uden problemer, så længe de befinder sig i dækningsområde.
- ***Fleksibilitet:*** Brugere kan omstrukturer arbejdspladsen hvor som helst og kan blive opkoblet til netværket.
- ***Økonomi:*** Ved omstrukturering slipper man for kabler, dvs. økonomisk fordel.
- ***Skalerbarhed:*** Nye enheder kan adderes i netværket uden påvirkningen af tidligere enheder eller brugere.

Trådløse netværk er underdelt i tre grupper pga. deres dækningsrækkevidde. Der er Wireless Wide Area Network (WWAN), Wireless Local Area Network (WLAN) og Wireless Personal Area Networks (WPAN). WWAN dækker teknologier som 2G mobil, Global System for Mobile Communication (GSM). WLAN dækker over teknologier som 802.11, og WPAN dækker over Bluetooth samt IR.

Der findes to typer af trådløse netværk[9]:

- **Infrastrukturnetværk**, bruges når man vil have adgang til ressourcer, som befinder sig på et almindeligt kabel netværk.
    - *BSS – Basic Service Set,* dækker over et antal trådløse enheder, som fungerer sammen i et trådløst netværk
    - *ESS – Extended Service Set,* mest normale, fixed point. Flere BSS'er kan knyttes sammen til et ESS
- **Ad-Hoc**, bruges når man vil have et lille trådløst netværk, der ikke skal have adgang til et kabel netværk. Dette betegnes som nemt at implementere.
    - *IBSS – Independent Basic Service Set*, IBSS topologi involvere to eller flere trådløse stations kommunikation, peer-to-peer.

[9] Kilde: http://www.netgear.com/docs/refdocs/Wireless/wirelessBasics.htm

I ad-Hoc netværk kræves der en højere sikkerhed end i kabel- eller infrastrukturnetværk, fordi i kabel netværk sker data udvekslingen gennem kabler, og i infrastrukturnetværk benytter man generelt en central database, som indeholder sikkerhedsinformationer såsom nøgler og password. I ad-hoc netværk deles alle informationer, også sikkerhedsinformationer, over netværket.

I trådløse netværk er det muligt at benytte forskellige transmissionsteknologier. Disse teknologier beskrives nedenunder i tabellen:

	Infrared		**Spread Spectrum**		**Radio**
	Diffused Infrared	**Directed Beam Infrared**	**Frequency Hopping**	**Direct Sequence**	**Narrowband Microwave**
Data Rate (Mbps)	1 to 4	1 to 10	1 to 3	2 to 20	10 to 20
Mobility	Stationary/ mobil	Stationary with LOS	Mobile	Stationary /mobil	Stationary/mobil
Range (m)	15 to 60	25	30 to 100	30 to 250	30 to 40
Detectability	Negligible		Little		Some
Wavelength/frequency	λ: 800 to 900 nm		902 to 928 MHz 2.4 to 2.4835 GHz 5.725 to 5.85 GHz		902 to 928 MHz 5.2 to 5.775 GHz 18.825 to19.205GHz
Modulation technique	ASK		FSK	QPSK	FS/QPSK
Radiated power	-		< 1 W		25 mW
Access method	CSMA	TokenRing , CSMA	CSMA		Reservation ALOHA, CSMA
License required	No		No		Yes, unless ISM

Table 1: Sammenligning mellem forskellige transmissionsteknologier[10]

Der er en del problemer med trådløse netværk og trådløs kommunikation[11]:

- Det første problem er strømforbruget, da enheder i et trådløst netværk som regel er batteridrevene. Dette problem kan løses enten ved at formindske sendestyrken eller ved at minimere mængden af kommunikationen i netværket.
- Andet problem er, at pakkerne ofte tabes gennem kommunikationen. Dette sker pga. at enhederne er mobile, der er støj i omgivelserne eller at der sker kollision af datapakker.

[10] Kilde: Wireless Communications and networks by William Stallings, Prentice Hall 2001
[11] Kilde: Wireless Communications and networks by William Stallings, Prentice Hall 2001

- Det tredje problem er, at der oftes kun er mulighed for unidirectionelle links, dvs. at kommunikation er muligt fra en enhed til den anden enhed, men ikke omvendt. Dette sker pga. retningsbestemt kommunikation som f.eks. IR. Dette kan også ske i RF, hvis enhederne ikke sender med samme sendestyrke. Unidirectionelle links problem kan løses ved ikke at benytte unidirectionelle links, eller ved at vælge de enkelte enheder således at disse links ikke opstår.
- Det fjerde problem er, at forbindelser brydes i netværket pga. stor mobilitet.
- Det sidste problem opstår, når to enheder ikke kan kommunikere med hinanden, men begge prøver at sende en besked til samme modtager. Dette problem kaldes Hidden Terminal.

Da Z-Wave også er et trådløst netværk gives der i dette kapitel et overblik over hvad trådløs netværk er, hvilke typer trådløse netværk der findes, fordele og ulemper ved trådløse netværk og hvad for nogle transmissionsteknologier der findes for trådløse netværk. Alt i alt hjælper dette afsnit med at forstå hvordan Z-Wave netværket virker og hvad det består af.

I de næste par afsnit beskrives nogle trådløse kommunikationsteknologier, som er følgende:

- IRDA
- Bluetooth
- RFID
- 802.11
- Sensor Netværk
- Zensys' Z-Wave

2.2.2 IRDA

Infrarød teknologi, også kaldt IRDA[12], er en trådløs teknologi, der bruges til dataoverførsel fra en elektronisk enhed til en anden via lysstråler i det infrarøde spektrum. IRDA benytter lysbølger af en lavere frekvens end dem det menneskelige øje kan opfatte. For at en kommunikation kan finde sted, må enhederne være i line-of-sight, dvs. enhederne skal "se" hinanden. Rækkevidden af den infrarøde port er begrænset, idet man sjældent placerer enhederne mere end et par meter fra hinanden og har en bithastighed på op til 4 Mbps.

Infrarød teknologi er f.eks. belejligt i fjernbetjeninger. Mobiltelefoner benytter sig af teknologien, idet næsten alle bærbare computere og nyere mobiltelefoner har indbygget en infrarød port, men efter både Bluetooth og WLAN er blevet mere og mere udbredt, bliver IRDA mindre brugt.

Der er fordele ved at bruge infrarød teknologi:

- Infrarøde enheder koster ikke ret meget at indbygge i et produkt
- Infrarøde enheder er meget pålidelige

Der er dog også ulemper:

- Enhederne skal være i line-of-sight, dvs. frit udsyn mellem enhederne
- Infrarød teknologi er næsten altid envejskommunikation, dvs. en enhed kan ikke modtage og sende samtidig.

At infrarøde enheder skal være i line-of-sigth, kan også være en fordel nemlig, at man undgår interferens/støj fra andre enheder, da de to kommunikerende enheder skal stå overfor hinanden, dvs. overførelsen bliver sikre. Envejskommunikation kan også være en fordel da den sikrer, at det der sendes, kun accepteres af den ønskede modtager, selvom der er andre infrarøde modtagere til stede.

Mht. Z-Wave er infrarød teknologi en helt anden trådløs teknologi. I Z-Wave bruges RF til kommunikation, og derfor er "line-of-sight" ikke en nødvendighed som i IRDA, hvilket har indflydelse på sikkerheden. IRDA kommunikerer med 4 Mbps og kan brugs til streaming, hvor Z-Wave kun bruger 9,6 kbps og udelukkende bruges til kontroldata.

[12] IRDA: InfraRed Data Association, Kilde: http://www.hw.cz/english/docs/irda/irda.html

2.2.3 802.11x (WEP/WPA)

Den mest anvendte specifikation for trådløse netværk såkaldt WLAN, er udviklet af IEEE's (Institute of Electrical and Electronics Engineers)[13] 802.11 gruppe[14]. Standarden blev frigjort i juli 1997 og er et trådløst LAN (Local Area Network). Det har følgende egenskaber[15]:

- To medier:
  - Radiofrekvens (RF)
  - Infrarød (IR)

- To radiofrekvens (RF) teknologier bruges:
  - DSSS - Direct Sequence Spread Spectrum (2 og 11Mbps). DSSS virker ved at data krypteres med en matematisk formel og spredes udover en række kanaler, så kan modtageren dekrypterer med en "omvendt" matematisk formel. Har den fordel at den har højere datarater end FHSS ved færre kanaler, men dog på bekostning af højt strømforbrug
  - FHSS - Frequency Hopping Spread Spectrum (1 og 2 Mbps). FHSS virker ved, at der bruges en række frekvenser, som der hoppes "tilfældigt" i mellem. Fordelen ved FHSS er at det er nemt at implementere, svær at aflytte og har et lavt strømforbrug. Ulempen er dog den lave datarate i forhold til DSSS

IEEE gruppen arbejder stadigvæk med at forbedre de nuværende trådløse netværksstandarter. Nedenunder beskrives kort de forskellige grupper[16] i IEEE 802.11.

Gruppe a:

Udvikling af standard for både fysisk og logisk lag som tillader trådløs transmission op til 54 Mbps i 5 GHz båndet. (Offentliggjort i 1999)

[13] Website: http://www.ieee.org/portal/index.jsp
[14] Website: http://www.ieee.org/portal/index.jsp?pageID=corp_level1&path=about/802std&file=index.xml&xsl=generic.xsl#802_11gen
[15] Kilde: 802.11 Wireless Networks, The Definitive Guide, Matthew S. Gast, ISBN: 0-596-00183-5
[16] Kilde: http://grouper.ieee.org/groups/802/11/

Gruppe b:

Udvikling af både fysisk og logisk lag som tillader trådløs transmission op til 11 Mbps i 2.4GHz båndet (ISM). (Offentliggjort i 1999)

Gruppe b-cor1:

Definition af visse Management Information Bases i forbindelse med 802.11b. (I gangværende)

Gruppe c:

Tilføje sub-layer support til specifikke MAC procedurer, som vil tillade MAC bridging efter 802.1D standarden. (Færdiggjort)

Gruppe d:

Tilføjelse af bl.a. channelization, hopping patterns, nye MIB attributter og andre egenskaber som eventuelt vil være krævet hvis 802.11 skal anvendes i andre regulatordiske domæner til de fysiske lag i 802.11 . (I gangværende)

Gruppe e:

Forbedring af MAC-laget i 802.11 så der kan indføres Quality of Service, Class of service samt forbedrede sikkerhedsfunktioner. Formålet er at både 802.11a og 802.11b skal blive i stand til at øge båndbredden og forbedre protokoleffektiviteten så standarden f.eks. kan benyttes til Voice og Video applikationer. (I gangværende)

Gruppe f:

At udvikle en Inter-Access Point Protokol (IAPP) med egenskaber, der tillader flere internetudbydere at tilbyde services på det samme fysiske Access-punkt. (I gangværende)

Gruppe g:

Udvikling af fysisk lag til 802.11b standarden som tillader hurtigere dataoverførsel. Målet er at opnå en maksimal hastighed på 20 Mbps og det nye fysiske lag skal indeholde alle obligatoriske egenskaber fra 802.11b standarden og vil følgelig være bagud kompatibel med denne. (I gangværende)

Gruppe h:

Forbedring af det fysiske lag i 802.11a standarden, sådan at det tilpasser de europæiske krav til brugen af 5 GHz båndet (Hiperlan). Samtidig skal der implementeres mekanismer til styring af transmissionseffekt og spektrum udnyttelse som krævet i Europa. (I gangværende)

Gruppe i:

Indbyggelse af sikkerhedsfunktioner i 802.11 MAC-laget. Der indgår ikke sikkerhed i den nuværende 802.11 MAC standard. En række producenter er gået sammen om at udvikle en fælles Wi-Fi standard, som dog har vist sig meget ringe. (I gangværende)

Sikkerheden i IEEE 802.11 sikres ved brug af WEP[17] (Wired Equivalent Privacy), som er bygget på en algoritme (RC4), der bliver brugt til at beskytte den trådløse kommunikation mod aflytning og beskytter ligeledes mod uautoriseret adgang til netværket. WEP bygger på en hemmelig nøgle, som er delt mellem den mobile klient (f.eks. bærbar) og et access point. Den hemmelige nøgle bruges til at kryptere datapakkerne, før de bliver sendt over det trådløse netværk.

Af WEP nøgle's egenskaber, kan det nævnes at nøglen svarer til et password og bruges som et password. Det bruges både til autentificering og kryptering, som foregår ved 40 bits eller 104 bits RC4 kryptering. Alle brugere deler samme nøgle, som vælges af administratoren, der som regel indtastes manuelt og gemmes lokalt på hver enkelt PC.

Fordelen ved WEP er, at der ikke er nogen sammenhæng mellem nøgle og klartekst, og kryptering/dekryptering er hurtig, op til 10 gange hurtigere end DES. RC4 er nem at implementere og har et lavt processor forbrug.

Der er dog nogle sikkerhedsrisiko ved WEP, og det er bla. at WEP nøgler deles mellem alle enhederne og at hemmeligholdelse og autentificering sker med samme nøgle. Desuden skal nøglen være gemt på PC eller WLAN-kort (krav for at enheden kan komme på netværket).

I WEP er der fundet indtil flere svagheder. Det er muligt at dekrypterer trafik efter statistisk analyse (aflytning) af trafikken efter ét døgns opsamling. Derved er indsættelse af trafik fra ikke autoriserede stationer baseret på kendt klartekst (spoofing), og det er muligt at foretage aflytning af trafik på netværket ved aktivt at "narre" access punkterne til at dekryptere trafikken (aflytning).

Ved samarbejde mellem Wi-Fi og IEEE har man udviklet og udgivet WPA[18] (Wi-Fi Protected Access) i 2003, pga. de forskellige svagheder i WEP. WPA er altså afløseren for WEP og er noget

[17] Kilde: Kapitel 5, 802.11 Wireless Networks, The Definitive Guide, Matthew S. Gast, ISBN: 0-596-00183-5
[18] Kilde: http://www.wifialliance.com/OpenSection/protected_access.asp

mere sikker. WPA bruger 128 bit nøgler i stedet for 40 eller 104 bit nøgler i WEP. WPA har dynamiske nøgler pr. bruger, session og pakke. Nøglerne WPA distribueres automatisk.

Væsentlig forbedring af sikkerheden er sket ved, at der stadig anvendes WEP til kryptering sammen med der bruges TKIP - Temporal Key Integrity Protocol, som benytter RC4. Der understøttes, at der foretages nøglerotation for hver 10.000 pakke. Endvidere understøttes både shared secret og authentication server og der er support for 802.1x og EAP[19] (til store miljøer).

WPA er blevet forbedret og den nye version WPA2[20] (Wi-Fi Protected Access version 2) bringer WPA tættere på den endelige IEEE 802.11i standard.

Egenskaberne i WPA2 er følgende:

- AES (Advanced Encryption System)
- CCMP = CTR + CBC + MAC (Counter Mode CBC-MAC Protocol)
    - CTR = Counter Mode Encryption
    - CBC/MAC = Cipher Block Chaining/Message Authentication Code
- TKIP (Temporal Key Integrity Protocol)
    - Forventes ratificeret fra Wi-Fi Alliance i midten af 2004
- Frivillig i Wi-Fi produkter fra Q3/2004
- Krav i Wi-Fi produkter fra primo 2005:
    - WPA2 udstyr kan formentlig opgraderes til IEEE 802.11i
- WPA version 2 vil understøtte 802.11i
- Standard baseret
    - Eksisterende WEP udstyr kan ikke software-opgraderes til WPA2 /IEEE 802.11i

Der er nogle ulemper ved 802.11x, når vi sammenligner det med Z-Wave. Først og fremmest er 802.11x for meget strøm og ressource krævende, hvilket ikke er særlig godt i Z-Wave's tilfælde. For det andet er sikkerheden i 802.11x ikke særlig god, da der findes ulemper ved det brugte krypteringprotokol, WEP.

[19] EAP – Extensible Authentication Protocol, en autentifikationsprotokol som bruges i trådløse netværk samt VPN.
[20] Kilde: http://www.wifialliance.com/OpenSection/protected_access.asp

2.2.4 Bluetooth

Bluetooth er en standard for trådløs kommunikation via radiobølger, som anvendes i distribuerede netværk. Det er specielt designet for mobiludstyr med et lavt energiforbrug og anvender FHSS[21] (Frequency-Hopping Spread Spectrum), da det brugte frekvensbånd ikke kun bruges af Bluetooth. Vha. FHSS undgår man i stor stil interferens fra andre enheder. Bluetooth sender kun på den samme kanal, hvor hver kanal er delt ind i såkaldte ”time slots” med en varighed på 625µs. Dette betyder, at der skiftes kanal 1.600 gange i sekundet, og sandsynligheden for interferens er derfor lille. Desuden er den datamængde, der evt. ville gå tabt hvis interferens skulle ske, meget lille. Fejlkorrektion og retransmission er indbygget. Mønstret der hoppes efter, er pseudotilfældigt og udledes af en 48 bit unik adresse.

Bluetooth har følgende egenskaber[22]:

- Benytter ISM[23] frekvensbåndet fra 2.402 til 2.480 GHz
- Har p.t. en rækkevidde på 10 m, men der arbejdes på en ny version af Bluetooth som kommer til at have en rækkevidde på 100 m.
- Datahastighed op til 1 Mbps
    - 432 kbps fuld dupleks
    - 721/57 kbps symmetrisk, halv dupleks
    - 3 audio forbindelser á 64 kbps
- 79 separate kanaler
    - 1 MHz kanal til pseudotilfældigt at sende og modtage

Bluetooth’s primære formål er, at skabe en fælles standard for trådløs kommunikation og trådløs overførelse af data. Hvis Bluetooth enheder er indenfor rækkevidde, kan de opdage hinanden og hvilke services de udbyder. Enhederne i Bluetooth behøver ikke ”se” de enheder, de vil kommunikere med, så længe de er inden for den ønskede rækkevidde. Når flere Bluetooth enheder kommer inden for rækkevidde af hinanden, starter en ”forhandling”, som bestemmer om der skal udveksles data, eller om den ene skal tage kontrol over den anden.

[21] Kapitel 7, Wireless Communications and Networks by William Stallings, ISBN 0-13-040864-6
[22] Kilde: Kapitel 15, Wireless Communications and Networks by William Stallings, ISBN 0-13-040864-6
[23] ISM: Industrial, Scientific and Medical band (http://en.wikipedia.org/wiki/ISM_band)

Når enhederne er klar, vil de danne et netværk. Bluetooth enheder danner et Personal Area Network (PAN) også kaldet piconet[24]. Når forbindelsen er etableret, skifter medlemmerne af netværket samtidigt "tilfældigt" mellem frekvenserne, så de kan holde kontakt og undgå interferens med andre piconet, der eventuelt skulle være i nærheden. Dette gør forbindelsen sikker. Der kan maksimalt være 255 enheder i et piconet og af disse kan højest 8 enheder være aktive på samme tid. En af enhederne i piconettet styrer kommunikationen. Denne kaldes for master og de øvrige kaldes for slaver. Masteren begynder al kommunikation. Der sker ingen direkte kommunikation mellem slaverne.

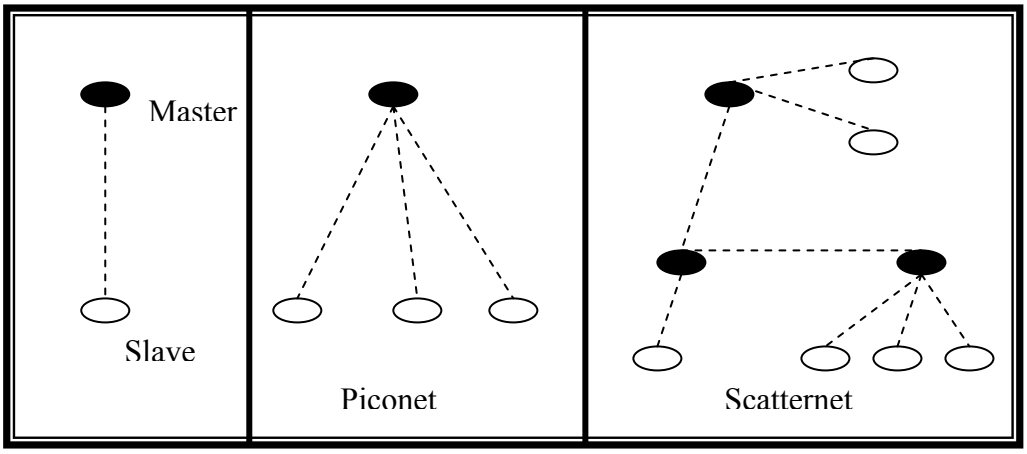


Figure 1: Piconet & Scatternet

I Bluetooth benyttes enten punkt-til-punkt eller punkt-til-multipunkt kommunikation. En master enhed kan lave enten en punkt-til-punkt eller en punkt-til-multipunkt forbindelse, mens en slave kun deltager i en punkt-til-punkt kommunikation med én master enhed. En punkt-til-multipunkt kommunikation foregår mellem master og slaver i et piconet.

Der er mulighed for, at enhederne kan deltage i flere piconet og så kaldes det for et scatternet[25]. Scatternet kan forøge risikoen for datafejl og retransmission, da de piconet der indgår i scatternettet, ikke koordinerer deres frekvenshop.

Pakkerne i Bluetooth kan blive op til 5 time slots brede og data i en pakke kan være op til 2.745 bits i længden. Der er i øjeblikket 2 forskellige typer af dataoverførsler mellem enhederne, nemlig:

[24] Kilde: http://www.webopedia.com/TERM/p/piconet.html
[25] Kilde: http://www.webopedia.com/TERM/S/scatternet.html

- SCO (Synchronous Connection Oriented)
- ACL (Asynchronous Connection-Less)

I et piconet kan der være op til 3 SCO forbindelser af 64 kbps hver. For at undgå timing- og kollisionsproblemer, bruger SCO forbindelserne reserverede slots dikteret af masteren. Masteren kan supportere op til 3 SCO forbindelser med 1, 2 eller 3 slaver. En master og en slave kan have en enkelt ACL forbindelse. ACL er enten punkt-til-punkt (master til én slave) eller broadcast til alle slaverne. ACL slaver kan kun sende, når masteren beder om det.

Som sagt, bruges der en 48 bit unik adresse til hver enhed. Der benyttes derudover to hemmelige nøgler, en 128 bit Private user key (autentifikation) og en 8-128 bit Private user key (kryptering). Endvidere bruges der et tilfældigt genereret 128 bit tal. Alt dette kombineres, og derfor kan Bluetooth standarden karakteriseres som en sikker kommunikationsform.

Det er en god idé at anvende Bluetooth, når man har med mobile enheder at gøre, da den tilbyder flere muligheder til mobile enheder. Den tilbyder kredsløb- og pakkekoblede forbindelser. Den anvender mindre pakker, da den ikke er tiltænkt LAN kommunikation og transport af store datamængder. Desuden er strømforbruget lavt. Ulemperne ved Bluetooth er, at der findes begrænset antal produkter, brugermæssig kan det være en udfordring at sætte netværket op, dog er der ved at ske fremskridt på begge områder.

Bluetooth og Z-Wave minder om hinanden, men Z-Wave har nogle bedre egenskaber end Bluetooth. Noderne i Z-Wave er ikke begrænset som noderne i Bluetooth, da slaverne i Bluetooth ikke kan kommunikere med hinanden. Rækkevidden i Z-Wave er også højere end rækkevidden i Bluetooth og sikkerhedsmæssigt er de to teknologier på samme niveau, da sikkerhed pt. er ikke eksisterende.

2.2.5 RFID

RFID står for Radio Frequency Identification[26] og består af tre ting:

- En antenne eller spole
- En tranceiver
- Et RF tag programmeret med unik data

Der findes to forskellige enheder under RFID, den ene er en kontroller og den anden er et tag. Grundidéen med RFID er trådløs identificering. Tag´et kan sammenlignes med en stregkode hvad angår den fysiske størrelse, men har flere fordele frem for stregkoden.

Af fordele kan nævnes:

- Trådløs aflæsning
- Kommunikation skal ikke foregå i line-of-sight, dvs. de skal ikke være synlige
- Mere robust da aflæsning ikke foregår optisk
- Kan have indbygget processor/intelligens
- Kan omprogrammeres

Passive tags har ingen strømkilde og ligger derfor og ”venter” på at blive genoplivet. Når en kontroller vil kommunikere med et tag, genereres et magnetisk felt, som antennen på tag’et omdanner til en elektrisk strøm og tag’et vågner. Afstanden som kontrolleren skal være fra tag’et for at vække det til live afhænger af styrken på feltet. Det varierer fra alt mellem 1 tomme til 100 fod (et par centimeter til 30 m). Passive tags har typisk ingen processor, men svarer tilbage til kontrolleren med et unikt nummer på mellem 32 og 128 bits.

Aktive tags har egen strømkilde bestående af et batteri og har en maksimal levetid på 10 år, hvorimod passive tags i teorien har en uendelig levetid. De aktive tags har typisk en processor samt hukommelse på op til 1 MB som kan aflæses/skrives til. Typisk vil tag´et give kommandoer til en maskine, som så udfører nogle målinger. Resultaterne af disse målinger skrives tilbage til tag’et og det bliver derved en ”logbog” for den enhed, som tag’et er tilknyttet. Skrive-/læseafstanden for aktive tags er typisk større end for passive grundet batteriet.

[26] Kilde: http://www.aimglobal.org/technologies/rfid/what_is_rfid.asp

Indenfor RFID finder der forskellige systemer, som opererer ved forskellige frekvenser. Lav-frekvens (30 – 500 kHz) systemer har kortere læse-/skriveafstand og kommunikerer langsommere end de systemer, som opererer ved 850 – 950 MHz eller ved 2,4 GHz. Disse tags kan aflæses hurtigere, men er til gengæld dyre. RFID benytter sig af FHSS da det er en af de mest effektive måder at undgå/modvirke interferens.

RFID har været i brug i omkring 10 år indenfor "tog-sporing" samt ved opkrævning af told ved landegrænser.[27] RFID er blevet mere og mere populært og er nu også ved at finde vej til supermarkederne. RFID er et godt supplement til stregkoder, men det vil nok aldrig udkonkurrerer stregekoder, da disse er yderst effektive/billige til mærkning.

RFID og Z-Wave har nogle fælles egenskaber som; begge bruger RF, begge er baseret på lav båndbredde og begge har lavt strømforbrug. Ser vi på sikkerheden, så er Z-Wave bedre sikret end RFID, udover RFID har kort afstand så findes der ikke nogen sikkerhed i RFID, da det primært bruges til simpel identifikation ligesom stregkoder.

[27] Kilde: http://www.aimglobal.org/technologies/rfid/resources/RFIDCharacteristics.pdf

2.2.6 Sensor Netværk

Sensor netværk er netværk af sensornoder, som har til opgave at indsamle informationer om omgivelserne. Sensor netværk bruges i mange forskellige industrier, som f.eks. transport (til at holde øje med trafikken på motorveje), sikkerhed (sættes f.eks. i storcentre, garager og indkøbssteder til at sikre sikkerheden), militæret (samle information om fjenden, deres bevægelser og handlinger) osv. Et eksempel på et Sensor netværk er et netværk af fastplacerede enheder, der har til opgave at overvåge trafik ved at sende oplysninger om f.eks. kødannelse tilbage til en station. Disse enkelte enheders rolle er, at indsamle data fra sit eget område og sende disse data tilbage til en station.

Et alternativ til stationære sensor netværk er mobile sensor netværk. Enhederne i et mobilt sensor netværk skal have mulighed for at kommunikere trådløst. Dette kan opnås ved, at alle enhederne er i direkte forbindelse med deres kommunikerende station ved hjælp af en Wireless Local Area Network (WLAN) teknologi. Hvis det ikke er muligt for alle enhederne, at være i direkte forbindelse med opsamlingsstationen, er det nødvendigt at videresende data gennem en eller flere andre enheder, dvs. det er muligt at enhederne kan kommunikere med hinanden. Da enhederne er mobile, er det oplagt at benytte teknikker som ad hoc netværk til at sende data fra de enkelte enheder til stationen. Enhederne i Sensor netværk har en lille radiosender med lav effekt og afstanden for enhederne ligger mellem 50-100m.

En general karakteristik for sensor netværk er følgende:

- Netværkstopologien skiftes hyppigt
- Noder bruger broadcast i deres kommunikation, hvor de fleste netværk bruger punkt-til-punkt
- Noder er begrænset mht. strøm og kapacitet
- Noder er tilbøjelige til at fejle
- Noder har ikke en global identifikation (ID)

Der er tre kategorier for kommunikationsmodellen for et Sensor netværk:

1. Node til station - kommunikation, sker f.eks. ved sensor aflæsning
2. Station til en node - kommunikation, sker f.eks. ved speciale krav
3. Station til alle noder - kommunikation, sker f.eks. efter krav fra stationen eller ved omprogrammering af hele netværket

Sensor netværk har typisk opgaver som:

- Finde en ønsket parameter værdi for et givet sted, f.eks. i miljø kan man tænke sig gerne at ville vide temperaturen for et bestemt sted
- Opdage/finde informationer om en ønsket ting og give informationen videre til stationen, f.eks. i transport hvor man gerne vil have viden om en specifik bils bevægelse
- Bestemmelse af et opdaget objekt, f.eks. ved betalingsanlæg ved broer vil man gerne vide om den opdagede ting er en motorcykle, bil, lastbil eller bus
- Holde øje med et objekt, f.eks. i militæret, hvor man gerne vil holde øje med fjenden

Sensor netværk og Z-Wave minder meget om hinanden, da de har mange fælles egenskaber. De bruger begge RF til kommunikation og dataoverførsel, der er lav båndbrede, noderne kan route data til andre og enhederne i begge netværk er begrænset mht. strøm, kapacitet og regnekraft.

2.2.7 Z-Wave

Z-wave er en trådløs teknologi, der er blevet udviklet af Zensys A/S som bruges til kommunikation mellem forskellige enheder i et trådløst netværk. Z-Wave teknologien skal gøre det nemt at styre en række af hjemmets installationer så som lyskilder, elektriske apparater, termostater, HVAC[28]-applikationer af forskellige art samt sikkerhed i hjemmet, som f.eks. adgangskontrol. Z-Wave protokollen bruger RF kommunikation med en lav datarate i applikationer, hvor turn-on tiden skal være hurtig, ca. 250 millisekunder. Alle enheder i et Z-Wave baseret system fungerer som en selvstændig repeater, og derved forøges rækkevidden fra oprindelige 30 meter til 120 meter. Z-Wave teknologien er baseret på en batteridreven arkitektur, hvor de enkelte enheder primært befinder sig i en sleepmode indtil aktion kræves.

Arkitekturen/Grundidéen i et Z-Wave netværk minder på nogle områder det der bruges i Bluetooth og i sensor netværk. I et Z-Wave netværk har nogle noder (kontroller noder) flere funktionaliteter end andre (slaver), og disse (kontrollerne) kan styre netværket på forskellige måder. I afsnit 2.2.1.5.2 ”Enheder i Z-Wave netværk” beskrives de forskellige nodetyper. De fleste noder i netværket har ikke samme begrænsning som et Bluetooth netværk, hvor flere slave noder ikke kommunikerer med hinanden. I et Z-Wave netværk kan noder route data videre til andre noder som i et sensor netværk.

Da Z-Wave netværk er lavet til Home Automation, kan det ske at flere netværk skal ”overlappe” hinanden, derfor er der brug for adskilles. Adskillelsen er nødvendig for, at et netværk ikke begynder at route pakker for andre netværk, og ligesom i andre netværk skal alle noder identificeres unikt, så fejl kan detekteres og arbejdes uden om. Dette er vigtigt, da noder kan blive utilgængelige, f.eks. pga. støj, så er det vigtigt, at datatrafikken kan ledes en anden vej. For at opfylde disse krav, gøres der brug af Home og Node ID.

Et Home ID er det som bruges til, at adskille de forskellige netværk fra hinanden. Det er meningen, at hver hjem har hver sit Home ID. Et Home ID er på 4 bytes og er preprogrammeret ned i alle kontroller enheder. Kontrollere kan deles om det samme Home ID, så flere kontrollere kan styre det samme netværk.

[28] Står for: Heating, Ventilating and Air-Conditioning

Alle noder/enheder i et Z-Wave netværk identificeres med et unikt Node ID (1 byte). Dette bruges til, at skabe overblik over hvilke noder, der kan se hvem. Ved at have en oversigt over hvordan netværket er opbygget, kan datatrafik ledes andre veje for at nå frem til destinationen, hvis en eller flere noder skulle fejle.

Noder (slaver/kontrollere) som ikke er med i et netværk har Home ID: 0 og Node ID:0, derved kan alle se, at de ikke er med i et netværk. Noder tilsluttes netværket via en kontroller, som tildeler dem Home og Node ID. Den primære kontroller i netværket har altid Node ID:239 (Hex: EF) og alle andre noder får tildelt Node ID mellem 0 og 232. De sidste Node ID's er reserveret til senere brug.
Vi vil nu kort se på, hvordan Z-Wave protokollen er opbygget i lag[29]:

- **Lag 1, MAC:** Dette lag styrer RF mediet. Laget opbygger en datastrøm som sendes igennem luften. Den er uafhængig af RF mediet, frekvensen samt modulationsformen der anvendes. Laget har en kollisionsundgåelsesmekanisme, som sørger for at der ikke bliver sendt, hvis en anden node sender.
- **Lag 2, Transport lag:** Her styres selve overførelsen af data mellem to noder. Det er inklusiv retransmission, kontrol af checksum samt kvittering for succesfuldt modtagelse af data.
- **Lag 3, Route lag:** Sørger for at data bliver sendt fra en node til en anden. Laget er ansvarlig for, at data bliver routed frem til den rigtige node. Hvor lag 2 står for forbindelsen mellem de individuelle noder, står dette lag for forbindelsen mellem start- og slutnode.
- **Lag 4, Applikations lag:** Dette lag indeholder de informationer/data, som applikationen hos den oprindelige afsender og modtager skal bruge.

Sammenlignes der med OSI modellen, kan Z-Wave protokollen betegnes som en 'light' model:

Z-Wave	Sammenhæng	OSI
1	→	1,2
2	→	3
3	→	4
4	→	5,6,7

Table 2: Sammenhæng mellem Z-Wave og OSI protokol

[29] Kilde: 903100103, Z-Wave Protocol Overview, May 5, 2003

2.2.7.1 Z-Wave netværk

Et Z-Wave netværk er et mesh[30] netværk, som er et trådløst lokal area netværk (WLAN) der bruger en eller to forbindelsestopologier, nemlig enten fuld mesh eller partiel mesh. Topologi er en beskrivelse af ordningen af et netværk, som indeholder noder og kommunikationslinier. Med et trådløst mesh netværk er det muligt at klare mange-til-mange forbindelser, og det er også både muligt at opdatere samt at optimere disse forbindelser.

En fuld mesh topologi går ud på, at alle noder er forbundet direkte til hinanden. En partiel mesh topologi er når nogle noder er forbundet til de andre noder, mens andre af noderne kun er forbundet med de andre noder, som de udveksler mest data med.

Et mesh netværk er pålidelig. Hvis en af noderne svigter i et mesh netværk ved ikke at virke længere, så fungerer kommunikationen i netværket stadigvæk enten direkte eller gennem en eller flere noder. På figuren nedenunder kan man se to mesh netværk, nemlig et fuld mesh og et partiel mesh.

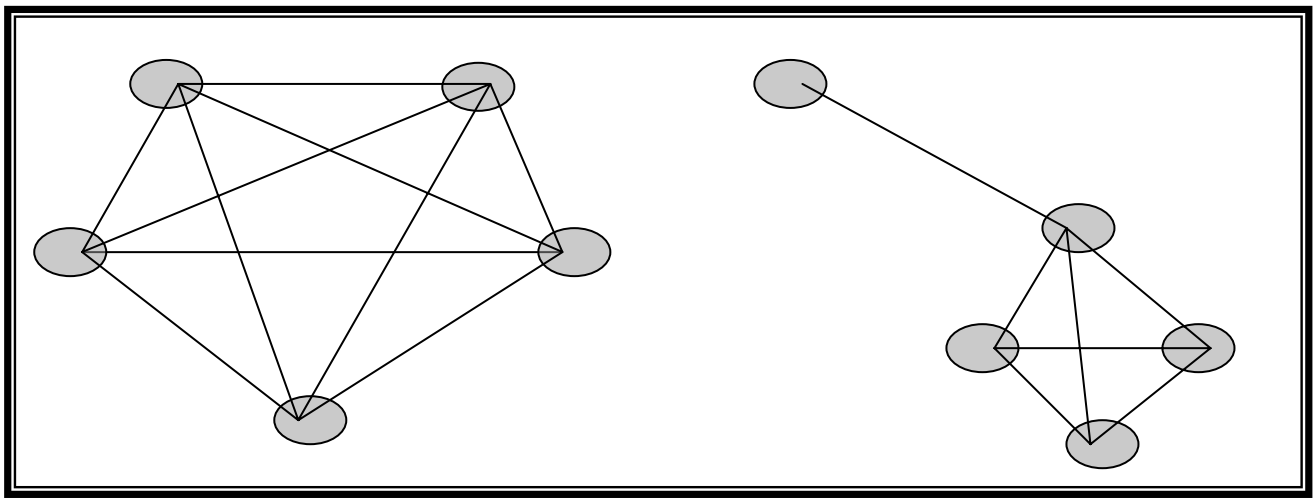


Figure 2: Mesh netværk, et fuld mesh og et partiel mesh.

En god introduktion af Z-Wave netværket og hvordan det virker fås ved at kigge på linket http://www.zen-sys.com/media/57.htm og se Flash demoen.

[30] Mesh netværk: http://www.webopedia.com/TERM/m/mesh.html

2.2.7.2 Pakketyper i Z-Wave netværk

I et Z-Wave netværk er kommunikationen struktureret ved, at gøre brug af nogle på forhånd defineret pakketyper. Hvilken pakketype der bruges til, at sende informationer frem og tilbage afhænger af, hvem pakken sendes fra/til samt formålet med pakken. De fire pakketype der findes, er beskrevet nedenfor:

- Singlecast

  Denne pakketype bruges til at sende informationer til én bestemt modtager.
  Det kunne f.eks. være at give én bestemt lampe besked på at tænde, eller en føler vil give besked til en kontroller om sin status.

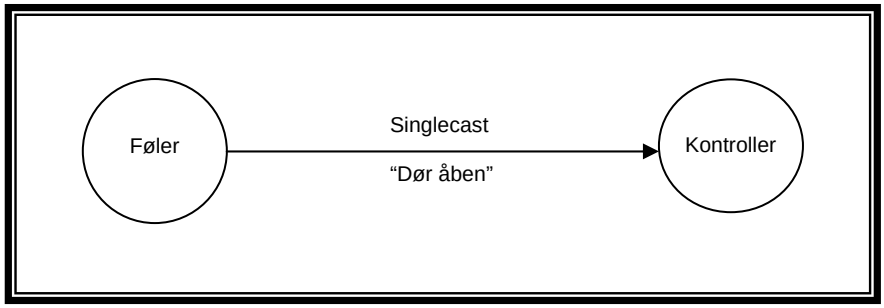


Figure 3: Singlecast pakke - En node sender en pakke til en specifik modtager

Kommunikation der foregår via denne pakketype betegnes som pålidelig kommunikation, da modtageren af pakken altid skal kvittere for modtagelsen. Hvis ikke der kommer noget svar inden ca. 200 ms, sendes pakken en gang til og der lyttes atter efter svar. Kommer der heller ikke svar denne gang prøves en sidste gang.

Alt efter hvilken applikation der prøver at sende, kan der ske forskellige ting ved manglende svar. Er det en slave, vil der ikke ske noget yderligere, men er det en kontroller kan denne vælge at prøve at bruge en af de andre noder til at rute igennem.

Størrelsen af denne pakketype varierer af hvor meget payload der er med. Maksimalt er pakken på 64 bytes i alt, men typisk ligger den på 20 bytes. Det kan vælges, at man ikke ønsker at modtage svar fra modtageren, men det frarådes.

- Transfer acknowledge (Transfer.ACK)

  Som beskrevet ovenfor så svarer modtageren af singlecast pakker altid tilbage, så afsenderen ved at pakken er modtaget. Dette gør den med denne pakketype.

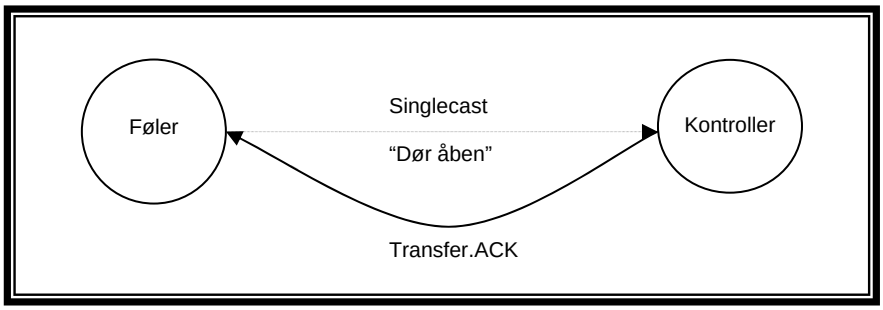


Figure 4: Transfer ack - Modtager kvittere for succesfuldt modtagelse af data

Denne type af pakke kan ikke have nogen payload og har derfor en fast størrelse på i alt 10 bytes.

- Multicast

  Denne pakketype bruges, når man ønsker at sende de samme informationer til flere noder. Pakketypen bruges når man ved hvem modtagerne er, og der kan sendes til maksimalt 232 noder på én gang. Det kan f.eks. bruges til at gruppere sine lamper, så man kan tænde/slukke for alle sine lamper i stuen på én gang.

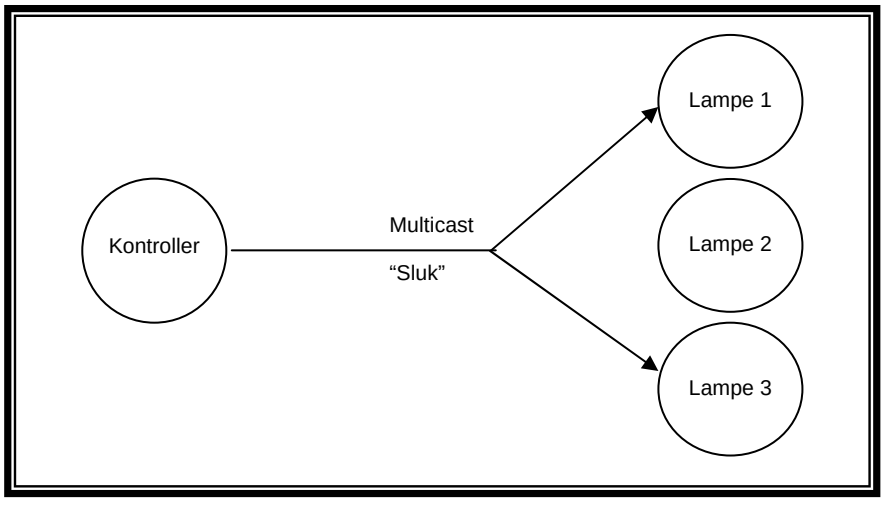


Figure 5: Multicast pakke - Node sender én pakke til flere specifikke noder

Der bliver kun sendt én pakke ud men med et udvidet modtager felt, hvor alle node ID´erne på modtagerne står. Derved spares der på den samlede kommunikation. Denne kommunikationsform er ikke pålidelig ligesom singlecast, da modtagerne ikke kvitterer tilbage at de har modtaget pakken. Hvis man vil have pålidelig kommunikation, må den opfølges af singlecast pakker til de enkelte noder. Alt efter payload har pakken en maksimal størrelse på 64 bytes.

- Broadcast

  Ligesom med multicast, så bruges denne pakketype til at sende den samme information til flere noder, men til forskel fra multicast, så er der ikke nogen specifikke modtagere. Dvs. pakketypen bruges når man ikke ved hvem modtageren/-ne er. Dette bruges f.eks. når en ny node skal inkluderes i netværket. Her ved den nye node ikke hvilken kontroller (modtager), der bruges til at inkludere med.

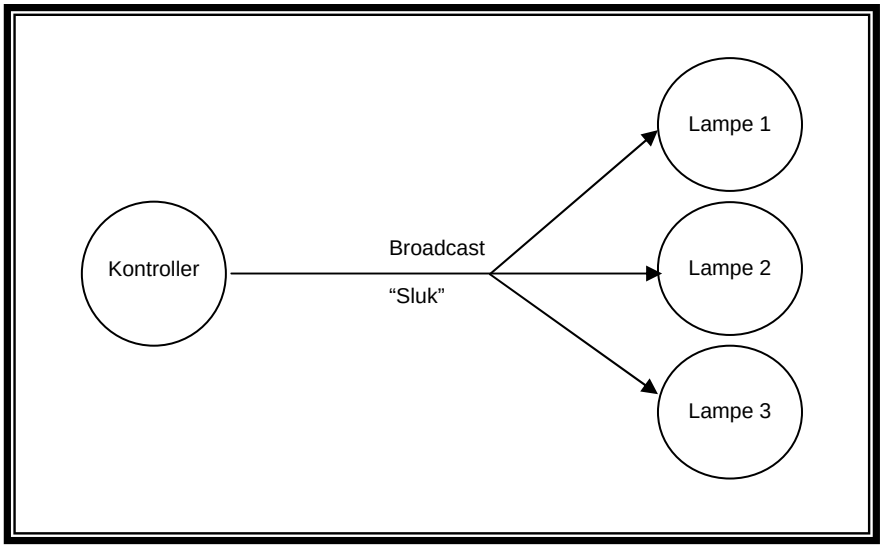


Figure 6: Broadcast pakke - Node sender én pakke til alle i nærheden (uspecifikke noder)

Denne kommunikationsform er upålidelig ligesom multicast, men kan opfølges af singlecast hvis nødvendigt. Alt efter payload har pakken en maksimal størrelse på 64 bytes.

2.2.7.3 Enheder i Z-Wave netværk

I et Z-Wave netværk findes mange forskellige enheder (noder), som har hver sin funktionalitet. Nedenfor er enhederne opstillet med forklaring på, hvad de er i stand til at gøre/kan bruges til:

- Slave (Slave)

  Denne enhed er den simpleste af alle enhederne. En slave skal altid befinde sig fysisk samme sted i netværket, fra den bliver inkluderet til resten af dens ”levetid”. Skal enheden flyttes, er det nødvendigt at ekskludere den for igen at inkludere enheden til netværket. Dette skal gøres for at nye naboer får kendskab til dens ”nærvær” og de ”gamle” til dens ”fravær”. Enheden er altid i en lyttende tilstand, så den er klar til at modtage kommandoer og sende svar tilbage. En slave kan ikke på eget initiativ sende kommandoer til andre. Kommandoer som ikke er til den selv, sender den videre i netværket (repeater).

- Rutende Slave (Routing Slave)

  Den rutende slave har samme funktionaliteter som den ovenfor beskrevet slave, samt nogle udvidelser. De ekstra funktionaliteter enheden råder over er, at den kan sende pakker/kommandoer op til 5 andre enheder, dvs. den kan f.eks. sættes op til når der sker et event, så giver den besked til andre enheder om at tænde sig. Den anden ekstra funktionalitet er, at den rutende slave har en RTC for tidsstyret opvågning, dvs. den kan f.eks. sættes op til, at vågne op på et bestemt tidspunkt, foretage en måling og så sende resultatet til en anden enhed. Der findes to typer af rutende slaver. Den ene er batteridrevet og er for det meste af tiden i sleepmode, men vågner op i bestemte intervaller for at spørge efter opdateringer. Denne repeater ikke. Den anden type er hele tiden vågen og repeater alle pakker som en normal slave.

- Portabel Kontroller (Portable Controller)

  Til forskel fra de ovenfor beskrevne enheder, så kan den portable kontroller flyttes rundt i netværket, som man har lyst til. Den bruges primært til at inkludere og ekskludere noder til netværket. Kontrolleren repeater ikke, og da den er batteridrevet, vil den typisk ikke være en lyttende node for at spare på strømmen. Den portable kontroller kan enten være primær eller sekundær kontroller i netværket.

- Statisk Kontroller (Static Controller)

  Den statiske kontroller skal altid befinde sig samme sted i netværket ligesom slave noderne. Den er netdrevet og lytter hele tiden efter pakker. Ligesom den portable kontroller, kan den sende og modtage pakker til/fra alle andre noder. Det anbefales, at den er sekundær kontroller, og kan konfigureres til at være en SUC (Static Update Controller).

- Statisk Opdaterings Kontroller (Static Update Controller)

  Denne enhed er lidt speciel, da den har en meget specifik opgave. Den modtager netværksopdateringer fra den primære kontroller og kan sende dem videre til sekundære kontrollere samt rutende slaver.

- Installerings Værktøj (Installer Tool)

  Enheden har samme funktionaliteter som en portabel kontroller. Den kan lave en kopi af netværksopsætningen, og applikationen der kører på denne enhed, kan se routing tabellen. Denne enhed er typisk den primære kontroller i netværket, når der skal inkluderes ny noder.

2.2.7.4 Inkluderingsprocessen i Z-Wave

I et Z-Wave netværk kan der forefindes forskellige slags enheder, som beskrevet ovenfor. Kommunikationen der foregår mellem enhederne i netværket benytter forskellige typer af pakker, som ligeledes er beskrevet tidligere. For at enheder kan fungere i et Z-Wave netværk, skal de først igennem en inkluderingsprocedure. Dette gøres for, at give dem en identitet samt at finde ud af hvorhenne fysisk set enhederne befinder sig i netværket.

Når en ny node (f.eks. en termostat) skal inkluderes, vil man typisk stå med enheden i den ene hånd og en kontroller (f.eks. Portable Controller) i den anden hånd. Vi går ud fra at kontrolleren i forvejen er inkluderet i netværket, hvorfor den allerede har et Home samt et Node ID[31]. I dette eksempel har kontrolleren Home ID: 9E5 og Node ID: 239. Alle enheder der ikke er inkluderet i et netværk har Home ID: 0 og Node ID: 0.

For at inkludere en ny node sættes kontrolleren i en oplæringstilstand (learning mode), så den er klar over, at en ny node skal inkluderes i netværket. Dette bevirker, at den lytter efter broadcast pakker. Når den modtager en broadcast pakke undersøges det om noden har et Home samt Node ID, dvs. om de er forskellige fra 0.

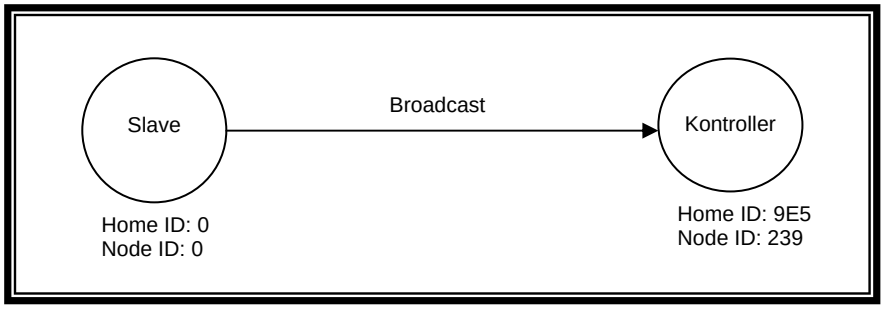


Figure 7: Node sender Broadcast pakke til kontroller, da den gerne vil inkluderes i netværket

Er dette tilfældet er noden allerede en del af et netværk og bliver derfor ikke inkluderet. Er noden ikke inkluderet, sender kontrolleren en ”Assign ID” pakke til noden.

[31] Se afsnit 2.2.7. ”Z-Wave” for yderligere gennemgang af Z-Wave netværk

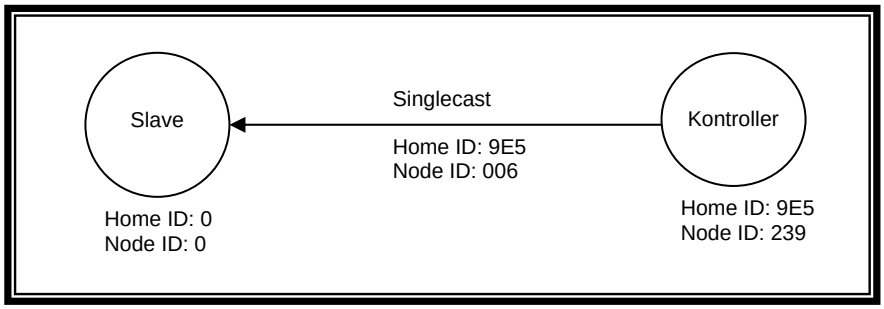


Figure 8: Kontroller sender AssignID til node

I denne pakke står der hvilket Home ID samt Node ID den skal bruge fremover. Bliver pakken succesfuldt modtaget (CRC[32] = ok), kvitterer noden tilbage til kontrolleren ved at sende en Transfer.ACK pakke.

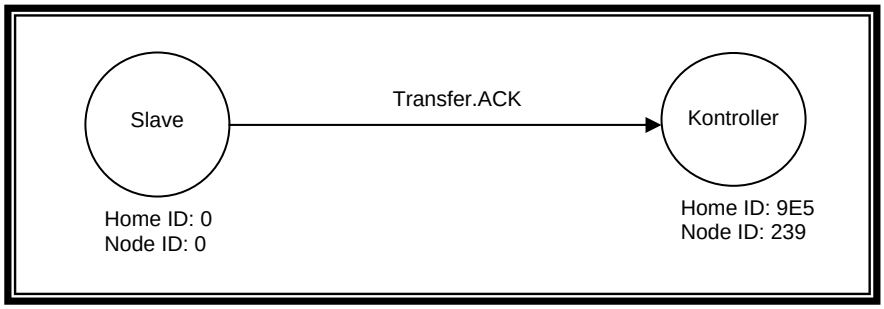


Figure 9: Node kvitterer kontroller for modtaget pakke

For at verificere at noden blev succesfuldt inkluderet, slutter kontrolleren af med at pinge[33] den nye node / sende en NOP (No OPeration) pakke til den nye node i netværket.

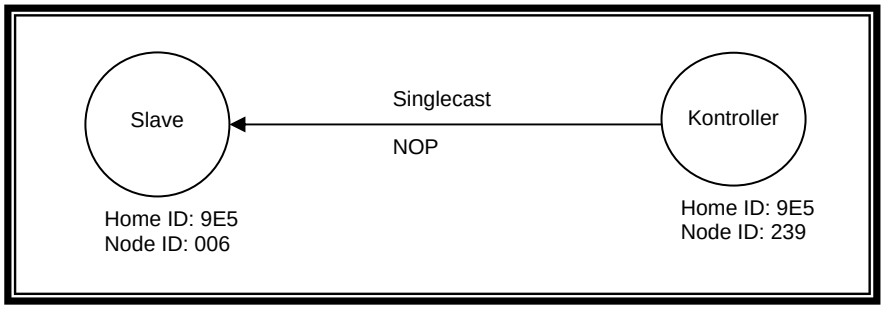


Figure 10: Kontroller sender NOP til node for at verificere Home og Node ID

Noden sender igen en Transfer.ACK pakke tilbage til kontrolleren, men denne gang med sit nye Home ID og Node ID. Kontrolleren ved nu, at noden blev succesfuldt inkluderet i netværket.

[32] CRC: Cyclic Redundancy Check, En metode til at detektere fejl under datatransmission (http://www2.rad.com/networks/1994/err_con/crc.htm)
[33] Begrebet "at pinge" stammer fra radar/sonar verden, hvor en impuls sendes af sted og man lytter efter ekkoet for at bestemme placeringen af/afstanden til et objekt. Skaberens beretning kan findes her: http://ftp.arl.mil/~mike/ping.html

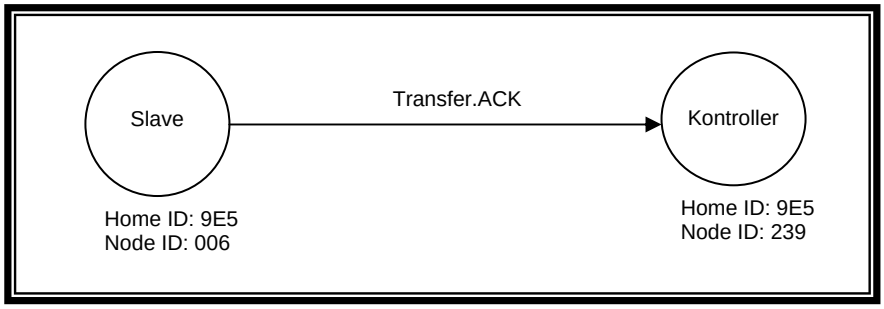


Figure 11: Node kvitterer kontroller for modtaget pakke

Hvis kontrolleren ikke får et svar tilbage fra den nye node efter ca. 200 ms, sendes endnu en NOP pakke af sted til noden. Dette gentager sig tre gange i alt, og hvis ikke kontrolleren får svar tilbage fra noden, regner kontrolleren ikke med, at noden blev succesfuldt inkluderet i netværket, hvorfor den slettes fra routing tabellen[34]. Routing tabellens egenskaber beskrives ikke yderligere, da det ligger uden for rammerne for denne rapport.

Den nye node er nu inkluderet og den sættes til at finde ud af, hvilke andre noder i netværket den kan se. Dette gøres ved, at kontrolleren sender en liste over hvilke andre noder, der er inkluderet i netværket (Node ID´er), hvorefter noden prøver at pinge disse noder. Når den har fundet ud af, hvilke andre noder den kan se, sender den listen tilbage til kontrolleren. Mere præcist sker det som beskrevet her nedenfor.

Kontrolleren sender en pakke til noden indeholdende en bit maske som fortæller hvilke andre noder den skal prøve at få fat i.

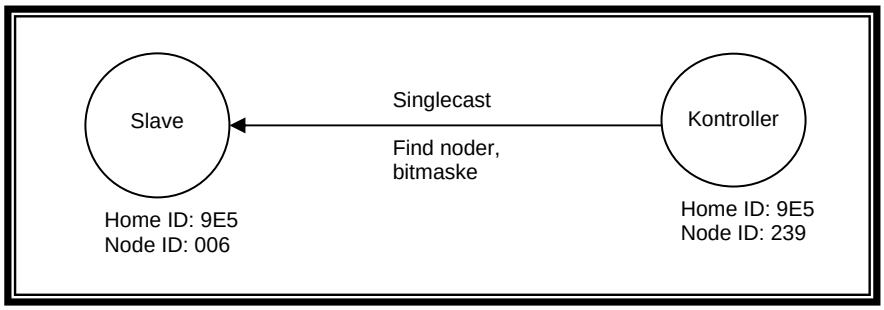


Figure 12: Kontroller sender liste over noder som skal forsøges kontaktet

Da der i alt kan være 232 enheder i et netværk, har vi maksimalt brug for at sende en 29 bytes (29 bytes * 8 bits/byte = 232) payload pakke. Bitmasken er opbygget på den måde, at hvis noden skal undersøge hvorvidt den kan se en anden node, så er der sat et 1-tal ved den position i bit masken svarende til node ID´et. Dvs. en node der lige er blevet inkluderet og som skal undersøge om den kan se Node ID 11, 12, 15 og 16 vil se således ud:

[34] Routing tabellen er en tabel, som bruges til at route pakker efter. Den viser hvilke noder der kan se hvem.

Node ID	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Bit værdi	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0

Table 3: Opbygning af bitmaske som viser hvilke noder der skal kontaktes

Noden sender derefter en Transfer.ACK tilbage til kontrolleren, så denne ved at pakken er modtaget.

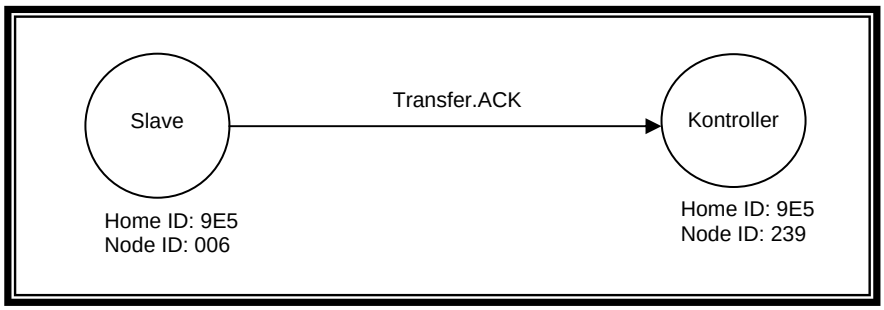


Figure 13: Node kvitterer kontroller for modtaget pakke

Noden vil nu prøve at pinge de noder den har fået besked på at kontakte via bit masken den lige har modtaget. Dette gør den ved, at sende en NOP_POWER pakker afsted til de forskellige noder og så vente på svar.

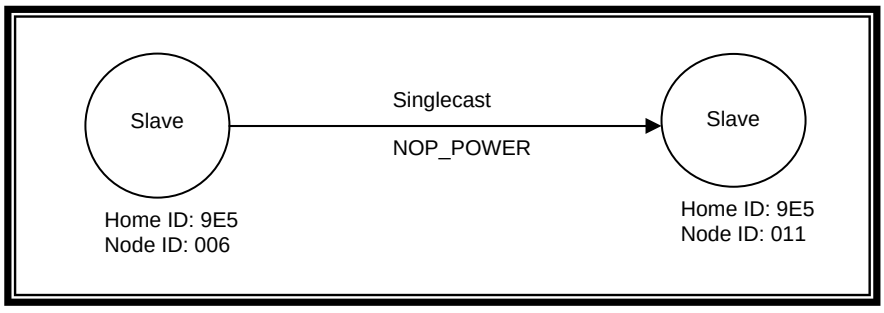


Figure 14: Node 'pinger' andre noder med lavere sendestyrke

Når den anden node modtager pakken, sender den en Transfer.ACK tilbage for at fortælle at de to noder godt kan nå hinanden.

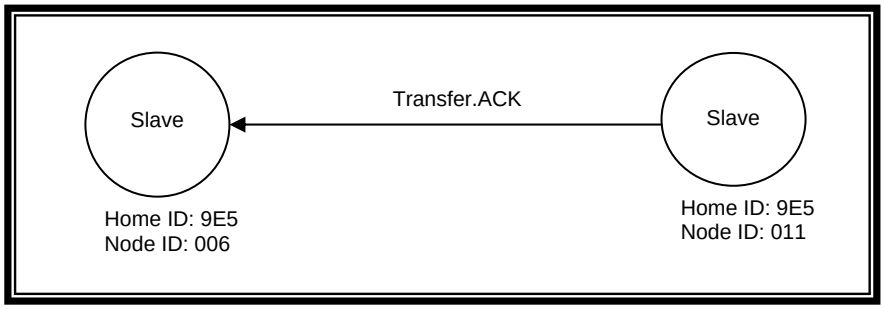


Figure 15: Node kvitterer node for modtaget pakke

Således fortsætter noden, indtil den har nået hele listen igennem. Pakken NOP_POWER er den samme pakke som NOP, men den sendes med an lavere sendestyrke (1 dBm[35]). Dette gøres fordi man gerne vil have "sikre" links. Ved at skrue ned for sendestyrken, sikrer man at de noder der ellers ville ligge lige på grænsen, og som derved ikke er helt sikre links, bliver "udelukket". Ved at gøre det på denne måde, får man færre links, men det er til gengæld "gode"/stabile links.

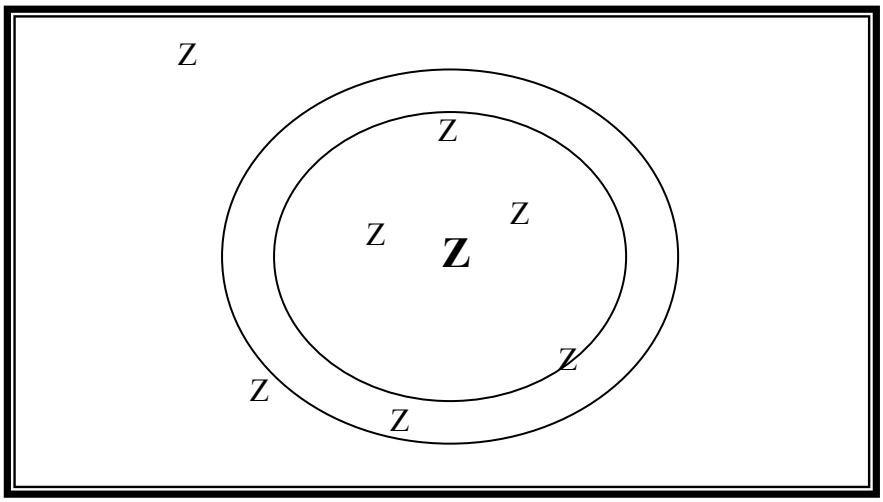


Figure 16: Ved nedsat sendestyrke fås færre men sikre links/noder

Noden er nu færdig med, at finde ud af hvilke andre noder, den kan se og sender derfor besked tilbage til kontrolleren, at den er færdig med sin opgave.

[35] dBm står for milideciBel og er en relativ måleenhed som benyttes inden for bla. den akustiske, elektriske og optiske verden. Yderligere forklaring findes på: http://www.kingfisher.com.au/appnotes/A01.htm

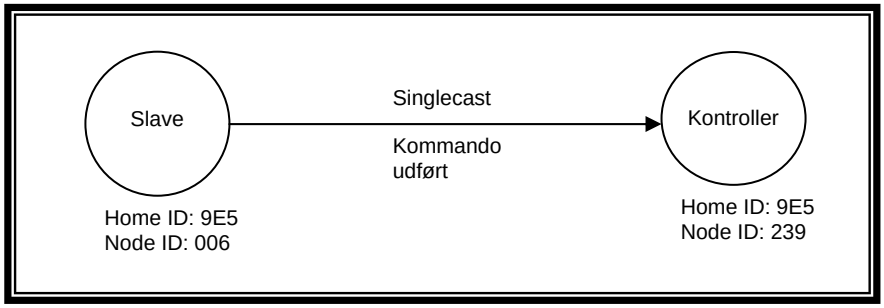


Figure 17: Node kvitterer kontroller for at kommandoen er udført

Kontrolleren kvittere tilbage til noden, at pakken er modtaget med en Transfer.ACK efterfulgt af en singlecast pakke, hvor den beder noden om, at fortælle hvilke noder den kunne se.

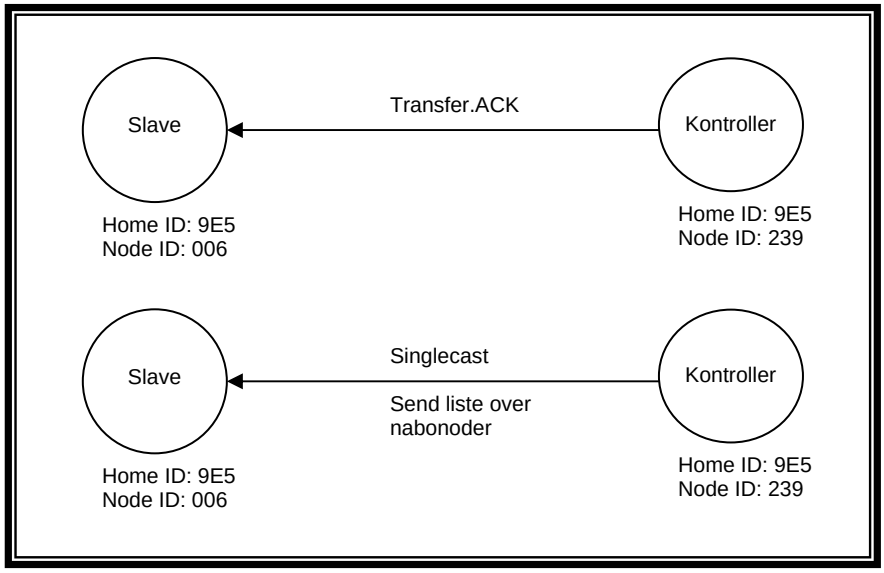


Figure 18: Kontroller kvitterer node for modtaget pakke og efterspørger liste over nabonoder til noden

Noden svar tilbage til kontrolleren med listen over noder, den kan se som en bitmaske i samme format som den modtog. Dvs. bit værdien på den position svarende til de noder den kan se, er sat til 1 mens de andre er 0.

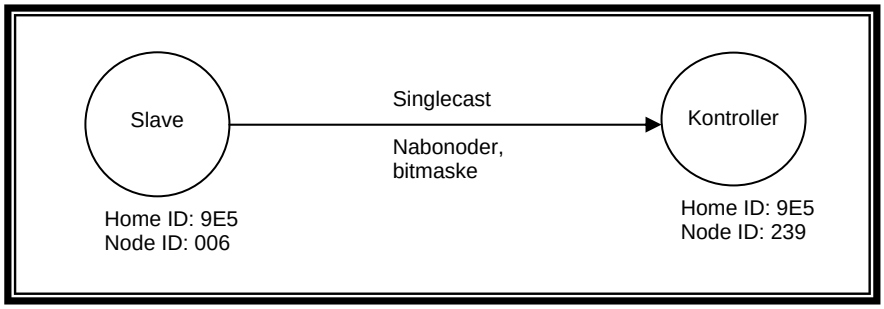


Figure 19: Node sender liste over nabonoder til kontroller

Kontrolleren svarer tilbage med en Transfer.ACK og inkluderingsprocessen er nu slut. Noden er blevet succesfuldt inkluderet i netværket.

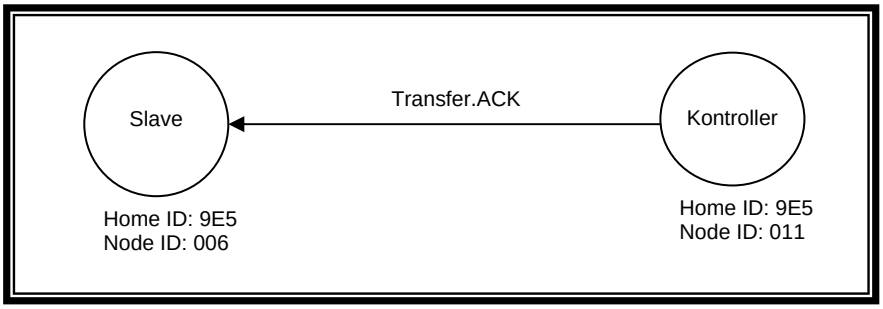


Figure 20: Kontroller kvitterer node for modtaget pakke

2.2.8 Opsummering af netværk

I starten af dette afsnit blev begrebet netværk beskrevet for, at give en indblik i hvad et netværk er og hvordan det er opbygget. Derefter beskrives trådløse netværk, hvor fordele og ulemper belyses.

Inden for trådløse netværk beskrives forskellige transmissionsteknologier, som man benytter i dag. Der er beskrevet trådløse teknologier som IRDA, 802.11x, Bluetooth, RFID og Sensor netværk. Disse trådløse teknologier er beskrevet for, at få et overblik over hvilke fordele og ulemper de besidder, og hvor deres primære anvendelsesområder er.

Efterfølgende beskrives Z-Wave netværket. Der beskrives, hvordan det er opbygget, hvordan kommunikationen fungerer (pakketyper), hvilke enheder der kan forekomme, samt hvordan selve inkluderingsproceduren fungerer. Disse afsnit giver en fuldstændig forståelse af, hvad for et netværk der er tale om her i opgaven, da det giver et indblik i hvilke ting man skal tage hensyn til, når man skal opbygge et trådløst netværk.

I slutningen af hvert afsnit, der beskriver de forskellige netværksteknologier, laves en sammenligning med Z-Wave. Vi finder f.eks. frem til, at Z-Wave er en ”light” model af OSI modellen. Z-Wave har kun 4 lag, OSI modellen har 7 lag[36].

[36] Se table 2 side ???

2.3 Sikkerhed

Et systems sikkerhed kan være svært at gennemskue, og ofte vil et system blive betragtet som sikkert, indtil andet er bevist. Men hvordan kan man forhindre fremmede personer i, at tilgå de data man gerne vil holde privat? Man kan f.eks. forhindre dem fysisk i at komme i nærheden af systemet, men dette er umuligt for de fleste systemer, som er distribueret rundt omkring på Internettet. Når man snakker om sikkerhed og det at sikre et system mod forskellige trusler, angreb og forstyrrelser, så er der nogle overordnede sikkerhedsmål som skal opfyldes. Disse sikkerhedsmål er:

- ***Fortrolighed*** (***Confidentiality***) giver sikkerhed for, at uvedkommende ikke får adgang til fortroligt data. Derfor er det vigtigt både, at kunne identificere brugeren af et system (***autentifikation***) og at gøre op, hvem der har adgang til visse data, men ikke til andre (***autorisation***). Sikkerhed mod tab af fortrolighed tilvejebringes gennem forskellige teknikker. Man kan enten afskære den fysiske adgang, f.eks. gennem adgangskontrol mv. til systemet, lægge logiske begrænsninger for, at komme ind i systemet ved brug af passwords mv. eller modificere dataene via kryptering, så det kun er forståeligt for brugere der kender proceduren, hvorigennem denne modifikation har fundet sted.

- **Integritet** (***Integrity***) giver sikkerhed for, at data ikke bliver ændret undervejs, så den meddelelse der bliver modtaget, er den ægte meddelelse som blev afsendt. I den papirbaserede verden er der ingen problemer med integritet, da det er svært at ændre den oprindelige meddelelse uden, at det bliver opdaget, men i den elektroniske verden kan man nemt ændre bits, da de ser ens ud. Ægthed og uafviselighed er to aspekter af integritet. Ægthed sikre at meddelelsen virkelig kommer fra den kilde, som den angiver, mens uafviselighed er den egenskab, at en person som sender meddelelsen ikke kan benægte afsendelsen.

- **Tilgængelighed** (***Availability***) giver sikkerhed for, at systemer og data skal være tilgængelige for autoriserede personer og fungerer på trods af mulige forstyrrelser. Forstyrrelser kan fx være angreb, uheld, strømafbrydelser og naturkatastrofer. Derfor er det vigtigt, at beredskabet er veludviklet.

Som sagt skal de ovenstående sikkerhedsmål opfyldes for, at et system kan være sikkert. Dette kan bl.a. gøres ved hjælp af krypteringalgoritmer og metoder. En måde f.eks. at undgå aflytning er ved, at formatere (kryptere) informationen, som skal transporteres, på en måde så kun modtageren kan forstå informationen. Det indebærer selvfølgelig, at både afsenderen og modtageren er klar over hvorledes informationen er formateret, dvs. de skal have kendskab til krypteringalgoritmen og nøglen(-erne).

Er der f.eks. tale om et klient-server system, så vil det være smart, at dele systemet op så fortroligt data befinder sig i et sikkert miljø. I sådan et klient-server system vil det ikke være smart, at lade en klient kommunikere direkte med serveren, da det kan få alvorlige konsekvenser for integriteten af data, da andre på Internettet kan få fat i informationer eller ændre i data på systemet, hvis forbindelsen mellem klienten og serveren ikke er sikker nok. Hvis klienten skal kommunikere med serveren for at lave ændring i data på serveren eller lignende, så skal serveren først og fremmest være sikker på, at klienten er den han/hun udgiver sig for at være, dvs. klienten skal identificerer sig. Derefter besluttes det om klienten har rettigheder til at ændre i systemet.

Identificering mellem klienten og serveren foregår ved, at klienten sender noget data til serveren, der identificerer klienten unikt. Da data sendes over en offentlig forbindelse, så er der risiko for, at uvedkommende opsnapper dataene. For at uvedkommende ikke kan bruge den opsnappede data til noget, så er det nødvendigt, at dataene ikke bliver brugt flere gange for derved undgås, at uvedkommende ikke kan bruge de opsnappede og krypteret data til at logge sig på systemet.

Når man snakker om et systems sikkerhed, så er det først og fremmest de ovenstående beskrevne sikkerhedsmål, som der skal tages stilling til og for, at disse mål bliver opfyldt, så kigges der nærmere på de ”mekaniske” begreber, nemlig kryptering. I de efterfølgende afsnit beskrives hvordan kryptering fungere, forskellige krypteringmetoder samt hvilke angrebsmetoder, man kan blive udsat for.

2.3.1 Kryptobegreber

Kryptering handler om at holde data hemmelig for uvedkommende, og er blevet et vigtigt emne over hele verden af computerbrugere, da mange ting foregår elektronisk og man vil ikke have, at andre få adgang til ens oplysninger. Der findes flere forskellige metoder, at kryptere med og i denne rapport gennemgås generelle kryptobegreber såsom Autentifikation, MAC, samt DES og RSA som er to af de mest brugte metoder i dag.

Ordet kryptologi stammer fra græsk og er sammensat af to ord, nemlig ordet krypto, som betyder at skjule og ordet logi, som betyder læren om. Dvs. kryptologi betyder læren om at skjule. Derimod betyder kryptografi hemmelig skrift, og et kryptosystem[37] består af fire dele:

- En mængde ***P*** af klartekster
- En mængde ***C*** af ciffertekster
- En mængde ***K*** af nøgler
- Et funktionsspar
    - En krypteringfunktion til at foretage en *kryptering,* der gør en givet klartekst *m* ulæselig til cifferteksten *c* (Se figuren nedenunder)
    - En dekrypteringsfunktion til at foretage en *dekryptering,* der oversætter cifferteksten *c* til klarteksten *m*. (Se figuren nedenunder)

Som regel er formlen for krypterings- og dekrypteringsfunktionen kendt af alle, mens der er nogle ukendte informationer, såsom en nøgle *K*, der bruges ved kryptering og dekryptering. Man kan skrive krypteringfunktionen som $E_k(m)$, dvs. klarteksten *m* krypteres med nøglen *K* og dekrypteringsfunktion kan skrives som $D_k(c)$, hvilket betegner cifferteksten *c* dekrypteret med nøglen *K*, altså samme nøgle.

[37] Peter Landrock og Knud Nissen. *Kryptologi – fra viden til videnskab*. ABACUS, 1997.

Ud fra de to nedenstående figurer kan man se at krypteringen $E_k(m)$ og dekrypteringen $D_k(c)$ kan evt. være hinandens inverse funktioner ($E_k(m)=c$ og $E_k^{-1}(c)=D_k(c)=m$) eller sammenfaldende.

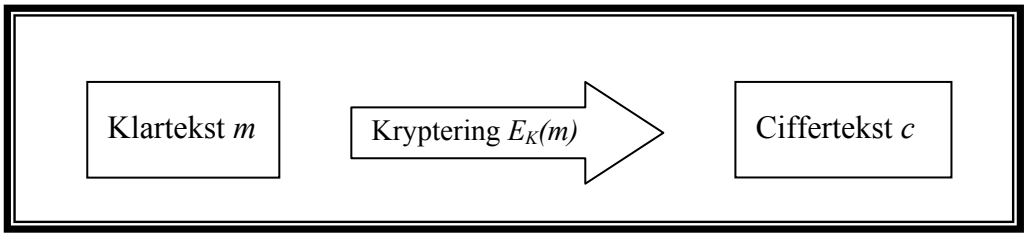


Figure 21: Kryptering $E_K(m)=c$

Man kan vha. en krypteringfunktion *E*, kryptere en klartekst *m* til en ciffertekst *c*

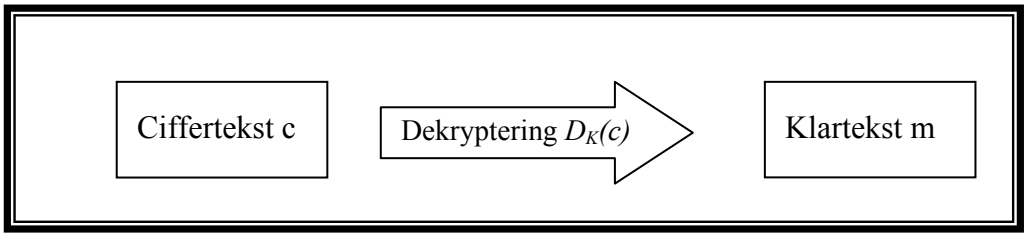


Figure 22: Dekryptering $D_K(c)=m$

Man kan vha. en dekrypteringsfunktion *D*, dekryptere en ciffertekst *c* til en klartekst *m*

Hvis man i et kryptosystem bruger samme nøgle til både kryptering og dekryptering, så er der tale om et ***private key kryptosystem*** (symmetrisk kryptosystem), men bruger derimod man to forskellige nøgler, én til kryptering og én til dekryptering, så har vi at gøre med et ***public key kryptosystem*** (asymmetrisk kryptosystem).

Sikkerheden af et kryptosystem kan deles op i tre dele[38]:

- Beregningsmæssig sikkerhed (Computaional Security) betyder, at de bedst kendte algoritmer for, at knække et system kræver et stort antal operationer og vil dermed kræve lang tid eller en stor beregningskapacitet for, at finde nøglen eller klarteksten. Det betyder altså, at det i praksis er umuligt at knække et system.
- Bevisbar sikkerhed betyder, at brydning af et system kan vises at være ækvivalent med, at løse et vanskeligt problem (faktorisering, diskret logaritme).
- Ubetinget sikkerhed (Unconditional Security) betyder, at der ikke er nogen begrænsninger på mængden af operationer, som en angriber foretager sig for at bryde et system. Kryptosystemet er dermed sikkert mod en angriber med uendelige beregningskraft.

[38]Dougles R. Stinson, Cryptography, theory and practice. CRC Press 2002

Der kan opstå forskellige angreb på kryptosystemer, disse angreb nævnes nedenunder:

- Ciffertekst, angriberen er kun i besiddelse af et antal chiffertekster
- Kendte klartekst, dvs. angriberen er i besiddelse af et antal klartekster og de tilhørende cifferteksters
- Selvvalgt ciffertekst, angriberen kan vælge en bestemt mængde cifferteksters og erhverve de tilhørende klartekster
- Selvvalgt klartekst, angriberen kan vælge en bestemt mængde klartekster og erhverve de tilhørende cifferteksters

2.3.1.1 Generelle begreber

Generelt betyder Autentifikation `at stå inden for` eller `at sige god for` og Meddelelses Autentifikation bruges til, at forhindre ændring af en meddelelse som 'den rigtige afsender' sender. Dvs. meddelelses autentifikation er den elektroniske ækvivalent til en underskrift på et stykke papir. Man sender en Message Authentication Code (MAC) med meddelelsen, ved brug af en algoritme, som skal sikre hvem meddelelsen kommer fra. Modtagere af meddelelser kan være sikker på, at den modtagne meddelelse kommer fra en rigtig afsender og ikke en angriber.

I det følgende afsnit gives der en beskrivelse af Autentifikation samt af MAC.

2.3.1.1.1 Autentifikation

Autentifikation bestemmer sikkerheden for identifikationskorrekthed. Autentifikation kan foregå med forskellige autentifikationsinformationer, som f.eks. PIN-kode, certifikat, kryptologiske nøgler og kodeord. I kryptografiens verden, hvor Alice og Bob gerne vil kommunikere med hinanden, skal de to brugere (Alice og Bob) være sikker på, at det er dem de kommunikere med, dvs. systemet skal understøtte autentifikation mellem dem. Det er ikke nok, at systemet understøtter autentifikation, det skal også sikre integritet sådan, at de to parters meddelelser ikke bliver ændret under transmissionen. Der skal både være autentifikation og meddelelsesintegritet, hvis kommunikationen skal være sikker. Der er tre metoder mennesker kan autentificere sig med, disse tre metoder nævnes nedenunder[39];

- Noget man har; Smart Kort, ID kort, Nøgle osv.
- Noget man ved; PIN kode, Password osv.
- Noget man er; Finger aftryk, stemme, DNA osv.

De ovenstående metoder bruges, når man skal autentificere sig på forskellige måder. F.eks. når man gerne vil hæve penge fra en bankautomat, så skal man først bruge et magnetkort (noget man har) til at identificere sig med. Kortet indeholder information om personen. Efter kortet er accepteret, skal sikkerheden for identitetsbevisets korrekthed bestemmes, og dette kan gøres vha. autentifikations-protokoller. Dvs. personen skal i denne situation vha. sin PIN-kode (noget man ved) bestemme korrektheden af sit identitetsbevis. Et andet eksempel er, når man skal logge sig på en mail-server, så man identificerer sig først med et brugernavn, og derefter beviser man sin identitet med et kodeord.

Dvs. de ovenstående metoder bliver brugt på forskellige autentifikationsprotokoller[40], og nogle af disse protokoller gennemgås herunder.

[39] Kilde: Applied Cryptography, Bruce SCHNEIER
[40] http://www.iscit.surfnet.nl/team/Erik/masterth/authenti.htm og http://www.cs.wm.edu/~hnw/courses/cs420/slides/ch13.pdf

- **Autentifikation ved anvendelse af offentlige nøgler**[41]

Her kigges der nærmere på en autentifikationsprotokol baseret på en offentlig nøgle og hvor begge parter er i besiddelse af hinandens hemmelige nøgle. Denne protokol er vist på figuren nedenunder.

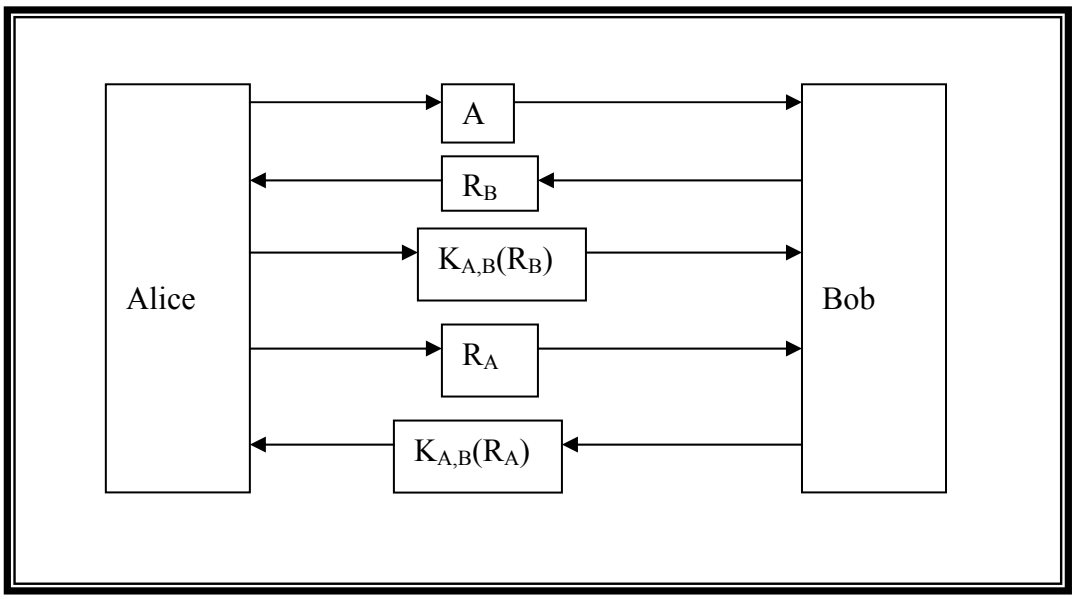


Figure 23: Autentifikation ved anvendelse af offentlige nøgler

Der sker følgende:

1. Først sender Alice sit ID (A) til Bob
2. Bob svarer med en udfordring R_B, han vil nemlig være sikker på, at det er Alice han kommunikere med
3. Alice krypterer Bobs' udfordring R_B og sender den tilbage til Bob. Bob dekrypterer den og konstaterer, at det er Alice
4. Alice sender så en udfordring R_A til Bob for, at sikre sig at det virkelig er Bob, hun er i kommunikation med
5. Bob krypterer Alices' udfordring R_A og sender den tilbage til Alice. Alice dekrypterer og konstaterer, at det er Bob

[41] Kilde: Kapitel 3 i Appied Cryptography, Bruce Schneier

- **Autentifikation ved anvendelse af nøgle distribuerings center (KDC)[42]**

Et af de problemer der opstår ved anvendelse af en delt nøgle til autentifikation er problemer/begrænsninger på skalerbarheden. Hvis et distribueret system indeholder N hosts og hver host skal dele hemmelige nøgler med hver af de andre N – 1 hosts, så skal systemet som helhed være i besiddelse af N(N-1)/2 nøgler, og hver host skal kunne håndtere N-1 nøgler. Når N er stort, vil dette føre til problemer. Et alternativ er at anvende en centraliseret metode ved hjælp af et nøgle distribuerings center (KDC). Dette KDC deler en hemmelig nøgle med hver af gæsterne (hosts), men har ikke parvis behov for at dele en hemmelig nøgle med samtlige hosts. Med andre ord, anvendelsen af KDC kræver at der skal håndteres N nøgler i stedet for N(N-1)/2, hvilket er en klar forbedring.

Hvis Alice ønsker at oprette en sikker kanal sammen med Bob, så kan hun gøre det med hjælp af det troværdige KDC (nøglecenter). Hele idéen er, at KDC giver en nøgle til både Alice og Bob som de kan anvende til kommunikation. Kerberos- og Needham-Schroeder protokollen[43] er nogle eksempler på denne autentifikationsprotokol.

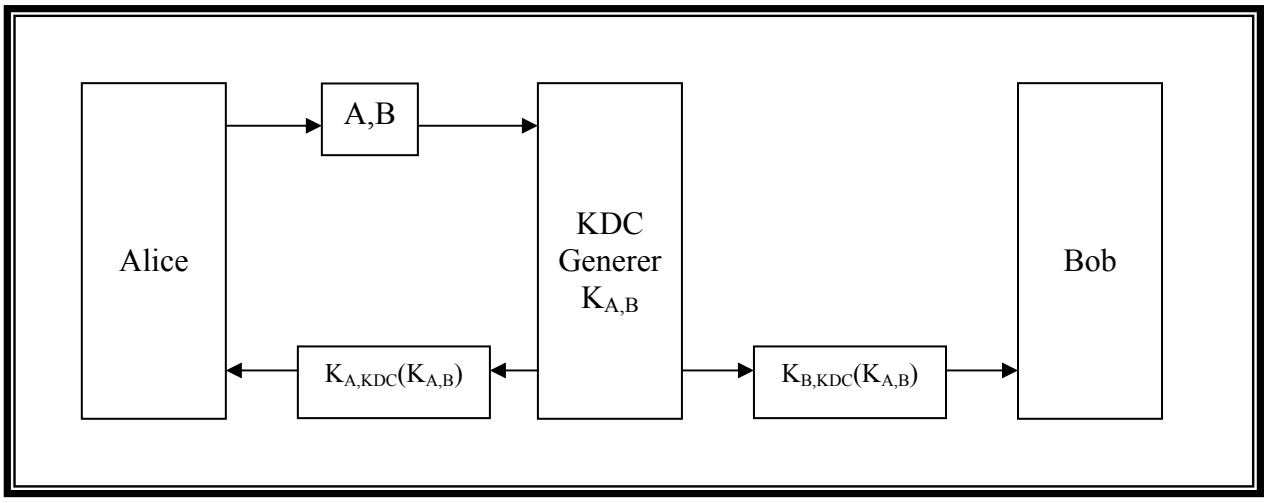


Figure 24: Autentifikation ved anvendelse af KDC

[42] Kilde: Kapitel 3 i Appied Cryptography, Bruce Schneier
[43] Se Kapitel 3 i Appied Cryptography, Bruce Schneier

Der sker følgende ved brug af nøgle distribuerings center:

1. Alice starter med at sende sit eget ID samt ID´et på den, som hun ønsker at lave en session med, til KDC
2. Nøglecenteret KDC sender en sessionsnøgle $K_{A,B}$ krypteret med Alices og KDC's fælles nøgle til Alice
3. Det samme sender KDC til Bob, men her krypteres der med Bobs og KDC's fællesnøgle

Den store ulempe med denne metode er, at Alice kan komme til at starte en sikker kanal med Bob, inden Bob har modtaget den delte nøgle fra KDC. Desuden er KDC forpligtiget til, at give Bob denne nøgle til denne session. Disse problemer kan undgås, hvis KDC blot overdrager $K_{B,KDC}(K_{A,B})$ til Alice for derefter, at lade hende varetage forbindelsen til Bob. Dette fører til protokollen som vist nedenunder. Meddelelsen $K_{B,KDC}(K_{A,B})$ betegnes også som en billet. Det er Alices job at overdrage denne billet til Bob. Bemærk, at Bob stadig er den eneste der kan få fornuft ud af denne billet, da han er den eneste sammen med KDC der ved, hvordan man dekrypterer informationen af denne.

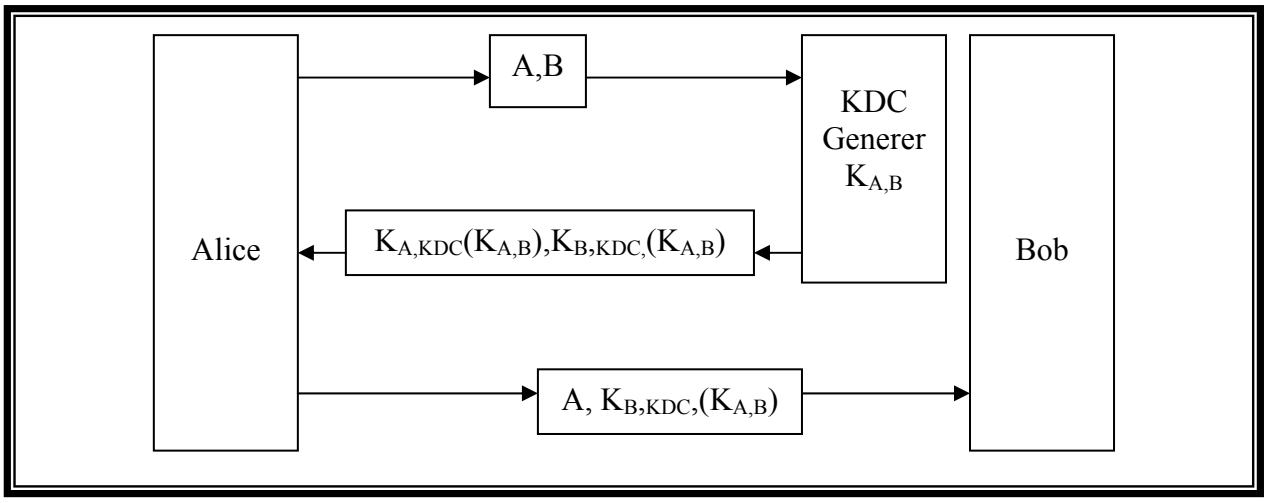


Figure 25: Autentifikation ved anvendelse af KDC

Der sker følgende:

1. Alice sender ID på sig selv og den hun vil tale med til KDC
2. KDC sender fælles sessionsnøglen $K_{A,B}$, $K_{B,KDC}(K_{A,B})$ den fælles sessionsnøgle krypteret med KDC's og Bobs hemmelige nøgle, $K_{A,KDC}(K_{A,B})$ den fælles sessionsnøgle krypteret med KDC's og Alices hemmelige nøgle
3. Til sidst sender Alice sin ID sammen med $K_{B,KDC}(K_{A,B})$ sessionsnøglen til Bob

- **Autentifikation ved anvendelse af hemmelig nøgle og kryptografi**

Nu kigges der på autentifikation med en hemmelig nøgle, der ikke kræver en KDC. Igen overvej den situation, at Alice ønsker at starte en sikker kanal til Bob. En typisk autentifikationsprotokol der er baseret på en hemmelig nøgle kryptografi vises nedenunder:

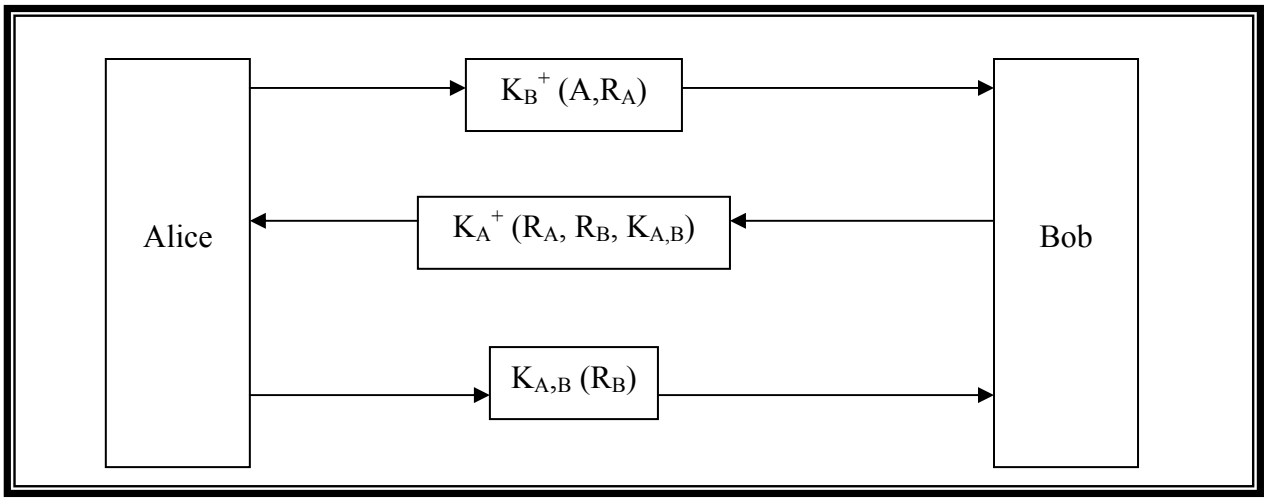


Figure 26: Autentifikation ved anvendelse af hemmelig nøgle og kryptografi

Der sker følgende:

1. Først sender Alice hendes ID og udfordring R_A, krypteret med Bob's hemmelige nøgle, så det er kun Bob der kan dekryptere denne meddelelse
2. Bob svarer med, at sende svar på udfordringen R_A, giver Alice en udfordring (R_B) og en sessionsnøgle $K_{A,B}$. Det hele krypteres med Alice's hemmelig nøgle, så kun Alice kan dekryptere det
3. Alice svarer på udfordringen fra B ved, at kryptere svaret med sessionsnøglen $K_{A,B}$. Dermed bliver Bob bevidst over, at det er Alice han kommunikere med

2.3.1.1.2 MAC (Message Authentication Code)

Der er to metoder til, at sikre besked integriteten, man kan enten benytte public-key digital signatur (kan godt betragtes som en MAC, men er generelt dyrt at bruge og har nogle problemer) eller man kan benytte MAC.

MAC har følgende egenskaber:

1. Det er svært at lave en MAC, hvis man ikke kender nøglen
2. Hvis man kender meddelelsen og den rigtige MAC til meddelelsen, så er det svært at lave en ny meddelelse med samme MAC eller en anden MAC
3. Det er svært at finde meddelelsen, hvis man kender den rigtige MAC

Meddelelse Autentifikation Kode er en standard i kryptografi, som bruges til at autentificere en meddelelse. MAC står for Message Authentication Code (Meddelelse Autentifikation Kode) eller Data Autentifikations kode og bruges til, at sikre at data ikke bliver ændret under transmissionen. Der sker det, at man bringer en MAC sammen med meddelelsen ved hjælp af en algoritme, der afhænger både af meddelelsen og en nøgle. Nøglen kendes kun af sender og modtager, og dermed gør det svært for angribere at bryder meddelelsen. Både meddelelsen og MAC'en kan have vilkårlig længde, men i de fleste tilfælde har MAC en fast længde, som kræver brugen af en hash-funktion til at komprimere meddelelsen til en fastsat længde.

F.eks. kan en MAC fås ved, at anvende den mest kendte symmetrisk krypteringsmetode DES[44]. Dette gøres ved at bruge den i *Cipher Block Chaining mode*[45] (*CBC*), hvor hver 64-bit tekstblok efter kryptering bliver kombineret med den foregående krypterede blok. Kombinationen foregår ved en XOR operation, den første blok bliver kombineret med en *initialiseringsvektor* - en blok, som skal være kendt af begge parter. En MAC fremstilles nu ved, at man kører DES med CBC igennem alle meddelelsens blokke - og beholder den sidste blok. Denne blok kan opfattes som en slags avanceret checksum - en unik 64-bit værdi, som karakteriserer meddelelsen.

[44] Se afsnit 2.3.1.2 for mere information om symmetrisk kryptering og DES.
[45] Se Handbook of Applied Cryptography afsnit 7 for nærmere information om CBC

Den værdi der ved en proces, med hjælp af en hash-funktion omdanner en meddelelse til en streng af en fast længde, kaldes for en hashværdi. En hash-funktion skrives H(x), hvor x er meddelelsen. Anvendt i kryptering har hashfunktionen en række egenskaber udover, at den kan omdanne en meddelelse til en fast værdi:

- Hash-funktionen kan anvendes på en meddelelse af vilkårlig længde
- Hashfunktionen er en envejsfunktion, dvs. det er usandsynligt at finde den meddelelse som svarer til en bestemt hashværdi
- Hashfunktionen er kollisionsfri, dvs. det er usandsynligt at to forskellige meddelelser giver samme hashværdi

Tilsammen giver det hashfunktionen en høj sikkerhed, der gør den brugbar. Det skal ikke være muligt ved, at prøve sig frem med forskelligt input at komme frem til et given output i en hashfunktion, altså er det umuligt at gennemføre et brute force angreb. En anden ting er, at algoritmen skal sikre, at der er stor forskel på input og output. Ved kun at ændre én bit i input, så skal stort set halvdelen af output være ændret. En af de mest brugte hashfunktioner er Message Digest 5 (MD5) og har en længde på 128-bit hash, dvs. MD5 producerer et tal på 128 bits ud fra en tekstblok med en vilkårlig længde.

MD5[46] er udviklet af Ron Rivers i 1992. En anden meget brugt hash funktion er SHA-1 (Secure Hash Algorithm) som har en længde på 160-bit hash, dvs. SHA-1 producerer et tal på 160 bits ud fra en tekstblok med en vilkårlig længde.

[46] For mere information om MD5 hash algoritme se afsnit 18 i Applied Cryptography (Secund Edition) by Bruce Schneier

2.3.1.2 Symmetrisk kryptering med gennemgang af DES

En ***symmetrisk kryptering***, der også kaldes for Private key kryptering, er det simpleste og mest brugte kryptosystem. Kryptosystemet bruger kun én nøgle, som bruges både til kryptering og dekryptering. Et eksempel: Alice vil sende en meddelelse til Bob, så hun bruger nøglen til at kryptere meddelelsen og sender den til Bob. Bob modtager den krypterede meddelelse, og dekrypterer den med samme nøgle. Her skal man passe på, at de private nøgler holdes hemmelige, da sikkerheden i systemet hviler på disse nøgler.

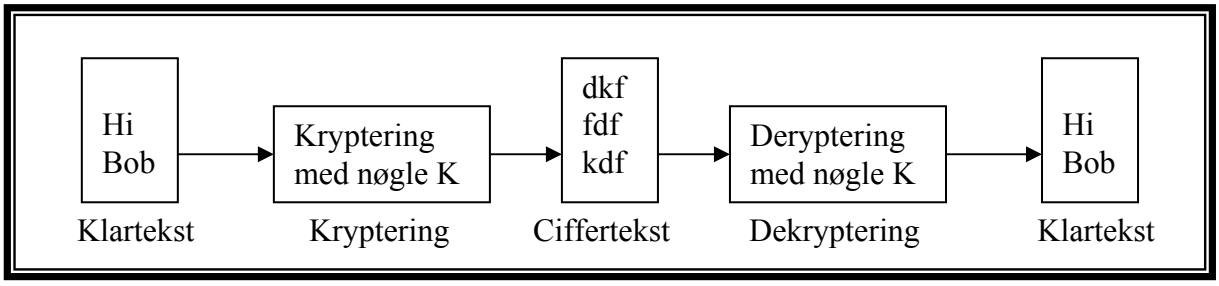


Figure 27: Symmetrisk kryptering

Der er et par problemer med symmetriske kryptering, disse problemer er følgende:

- Nøglerne i systemet skal deles gennem en sikker kanal ellers kan enhver få fat i nøglen og bryde systemet
- Hvis der er n-antal personer, der ønsker adgang til at kommunikere med hinanden, bliver antallet af nøgler meget stort, nemlig $\binom{n}{2}$, og dermed bliver det svært at holde nøglerne hemmelige
- Da man bruger samme nøgle til kryptering og dekryptering, så er det ikke muligt at bruge systemet til at signere meddelelser. Hvis der er to parter som besidder nøglen, kan begge have krypteret den

Der findes en del eksempler på ***private key kryptosystemer,*** men det mest simpleste og udbredte er ***DES (Data Encryption Standard). DES*** blev udviklet af IBM og siden adopteret af den amerikanske regering i 1977. Derefter blev den nærmest standard for al kryptering overalt i verden. Selv Dankortet er baseret på ***DES,*** som vi nu vil beskrive.

DES er en forkortelse for ***Data Encryption Standard***, og er en symmetrisk kryptering, der bruger kun en enkelt nøgle til både kryptering og dekryptering. ***DES*** er udviklet af IBM og National Security Agency og er en amerikansk standard kryptering. ***DES*** standarden er gjort offentlig og kan bruges og bliver brugt af mange. ***DES*** bruger en serie af bit permutations-, substitutions- og rekombinationsoperationer på blokke med 64 bit data og 56 bit nøgle.

DES bygger på et antal permutationer og en metode kaldet en ***Feistel-stige*** (hvor teksten sendes ind foroven og kombineres efter den indledende permutaion med undernøgler efter tur (se figuren på næste side) og bliver ***XOR***[47]. De 64 bit blokke går igennem en indledende permutation, ***IP*** (se nedenstående figur) og derefter opdeles de i to dele, hvor L_o er de første 32 bit og R_o er de sidste 32 bit. R_o krypteres vha. en funktion ***f*** med undernøgle ***K1*** og bliver ***XOR*** med L_o. Resultatet af dette bliver så til $R_1 = L_o$ (***XOR***) $F_{K1}(R_o)$ og den gamle R_o bliver $L_1 = R_o$. Denne permutation gentages hele 16 gange med forskellige undernøgler, og resultatet byttes om ved den sidste gentagelse. Dette har nemlig den fordel at dekryptering kan foregå på samme måde som kryptering, blot med undernøglerne i modsat rækkefølge. Til sidst permuteres dataene igennem med den tilsvarende inverse funktion til ***IP*** nemlig IP^{-1}. (Se figuren nedenunder for at forstå hvad der præcis sker).

I IP ombyttes dataene fra ***m*** bitvist til andre placeringer, dette sker efter et fastlagt skema[48]. Første linie i skemaet ser sådan her ud; ***58 50 42 34 26 18 10 2*** og det betyder at den første bit hentes fra 58`te position i ***m***, den anden bit hentes fra 50`te position osv. Permutationen sørger for at alle lige bits bliver til L_o og alle ulige bits til R_o.

På samme måde som ***IP*** er der fast lagt et skema som ***f-funktionen*** arbejder ud fra. Det første skridt er E[49]***,*** som laver de 32 bit data fra ***R*** om til 48 bit data. Første linie i ***E*** skemaet ser således ud; ***32 1 2 3 4 5*** og virker ved at første bit der kommer ud af ***E,*** er den 32`te bit i ***R***, næste er 1`te bit osv. Nu har man de 48 bit data som bliver ***XOR*** med undernøglen K_n. Nu bliver dataene inddelt i 8 blokke på 6 bit og ledes ind i hver sin ***S-boks***. En *S*-boks laver de 6 bit om til 4 bit data vha. S[50]-boks skemaet, som er fast defineret. ***S-boks*** skemaet bruges ud fra at den 1. og 6. bit i en blok beskriver et tal mellem 0 og 3, som angiver linjenummeret i skemaet. De inderste 4 bit angiver

[47] Kapitel 7, Handbook of Applied Cryptography, by A. Menezes, P. van Oorshot and S. Vanstore, CRC Press 1996
[48] Se bilag B.
[49] Se bilag B.
[50] Se bilag B .

kolonnenummer, og resultatet bliver et tal fra skemaet som er mellem 0 og 15. Nu har man altså 32 bit blok data og denne blok gennemgår endnu en fast permutation ***P***[51]. Denne permutation sørger for, at alle input bit bliver flyttet til en output position. Første linie i dette ***P*** permutationsskema er; ***16 7 20 21***. Dvs. at bit 16 flyttes til bit 1, bit 21 flyttes til bit 4, osv.

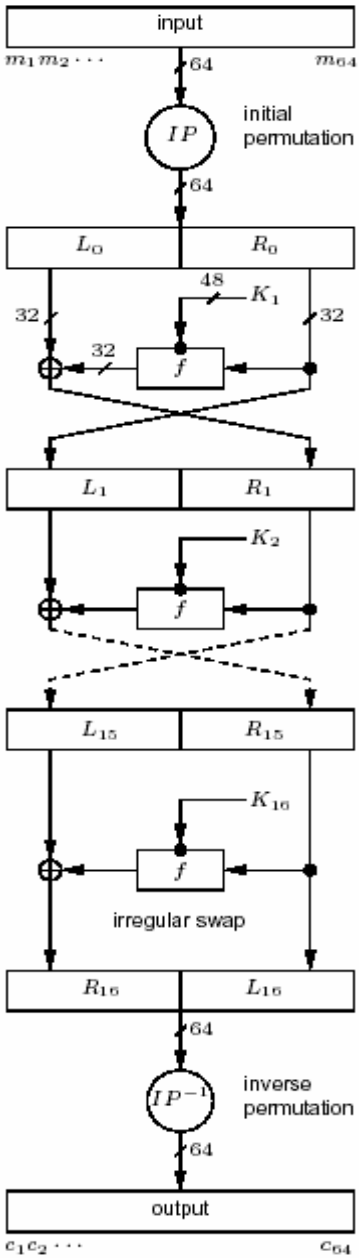


Figure 28[52]: Viser hvordan DES virker

Så mangles der at beskrive hvordan de brugte undernøgler i ***DES*** genereres. Først bruger man en indledende permutation, ***PC-1*** skemaet[53] til at reducere den 64 bit ***DES*** nøgle ned til 56 bit. ***PC-1***

[51] Se bilag B.
[52] Figuren stammer fra kapitel 7 I Handbook of Applied Cryptography.

vælger placeringen af de 56 nøglebits i to 28 bits halvdele C_oD_o. Når man nu har både C_o og D_o så køres de to 28 bit halvdele igennem en rotationsfunktion, som sørger for at rotere alle bit en (ved undernøglerne 1,2,9,16) eller to (ved resterende undernøgler) pladser til venstre. Til sidst bruges ***PC-2*** skemaet[54] til at reducere de to 28 nøglebits ned til to 24 nøglebits. Undernøglerne bestemmes udfra; $K_i = PC\text{-}2(C_i, D_i)$. For at forstå nøgle generering se figuren nedenunder:

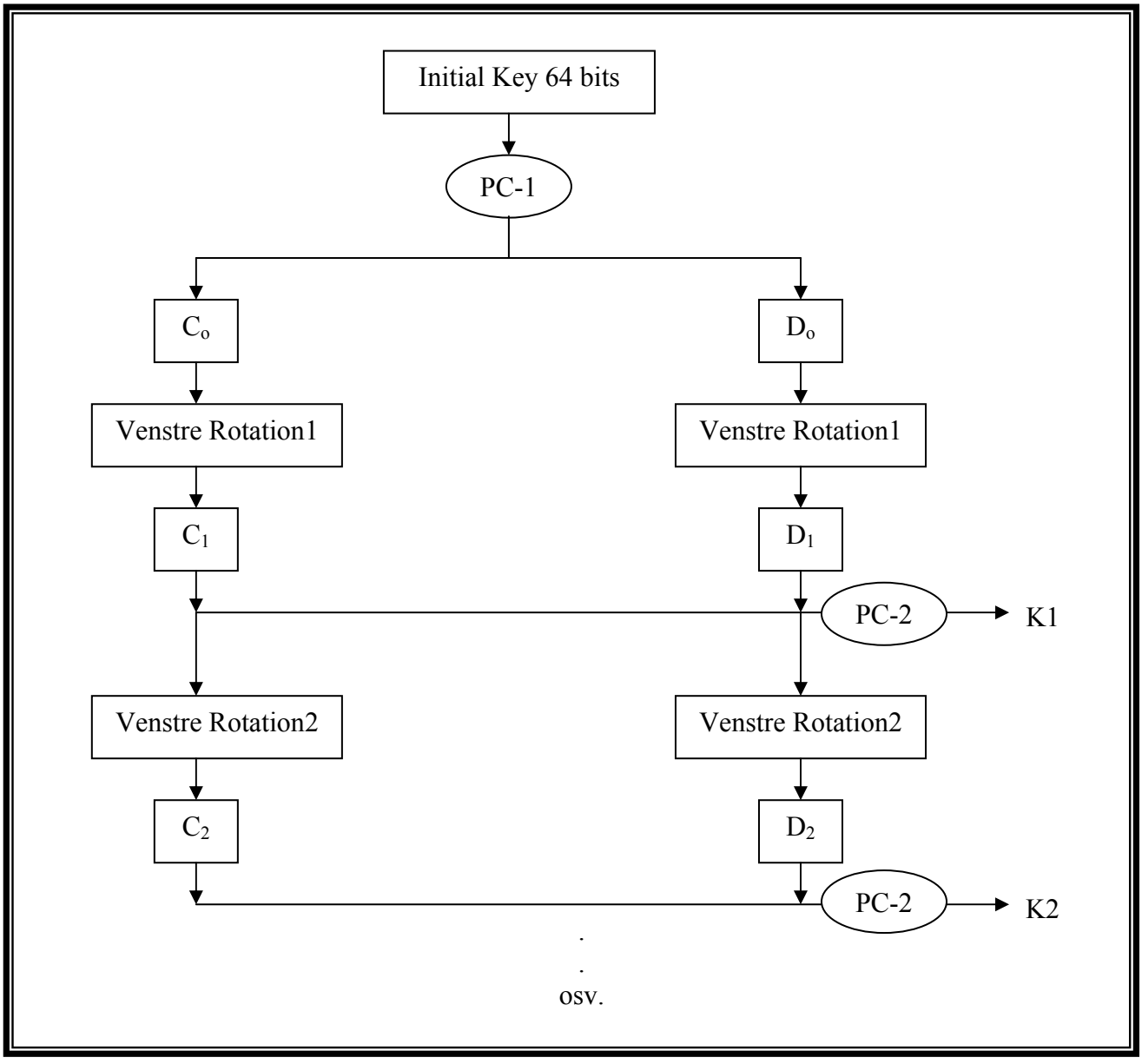


Figure 29: Viser hvordan undernøglerne i DES genereres.

[53] Se bilag B.
[54] Se bilag B.

Ved dekryptere med ***DES*** er det mulig,t at bruge samme undernøgler K_i som blev anvendt ved kryptering med den ene betingelse, at man skal huske og bruge undernøglerne i modsat rækkefølge ved dekryptering end man gjorde ved krypteringen.

Måden at bruge et blok ciffersystem kaldes brugsmåden (Mode of use). ***For DES*** er der defineret 4 brugsmåder. ***DES*** kan enten bruges i block-mode eller stream-mode. I block-mode opdeles cifferteksten i blokke og krypteres enten med ***Electronic Code Book (ECB)*** eller ***Chipher Block Chaining (CBC).*** I stream-mode opereres på bitsekvenser ved brug af ***Chipher Feedback (CFB)*** eller ***Output Feedback (OFB)***[55]**.**

Det er over 20 år siden ***DES*** er udgivet, men bruges stadigvæk. Der er heller ikke fundet nogle svagheder i algoritmen, men nøglelængden i DES er ved at være for kort. Man kan nemlig nu fortiden bygge en hurtig computer, som er i stand til at bryde ***DES*** på ca. et døgn ved at prøve alle tænkelige nøgler. Ikke alle kan opbygge sådan en computer pga. prisen på ca. 2 mio. dollars men man kan godt tænke sig, at regeringer og store virksomheder må have bygget sådan en maskine.

Det er muligt at øge sikkerheden af ***DES*** ved, at foretage flere krypteringer med ***DES***. Der findes to meget brugte metoder, de kaldes ***Double DES*** (Med ***double DES*** kryptere man én besked to gang med ***DES*** algoritmen, så antal mulige nøgler bliver $(2^{56})^2$) eller ***Triple DES*** (Med ***triple DES*** kryptere man en besked tre gang med ***DES*** algoritmen, så antal mulige nøgler bliver $(2^{56})^3$). ***Double DES*** ser ud til at være sikker, men skindet bedrager, da der er blevet fundet et angreb mod ***double DES***, nemlig et "plaintext angreb". Så ønsker man ekstra sikkerhed, er man nød til at bruge ***triple DES***. ***Triple DES*** er forskellige fra ***double DES*** på to måder, den bruger tre gang ***DES*** og udover det bruger den også ***DES*** dekryptering som en del af det samlede system.

En kryptering forgår ved; $\mathbf{c = E_{K3}(D_{K2}(E_{K1}(m)))}$
En dekryptering forgår ved; $\mathbf{m = D_{K1}(E_{K2}(D_{K3}(c)))}$

Hvor c = chiffertekst, m = klartekst, K = (K1,K2,K3) er nøglerne, E er kryptering og D er dekryptering.

[55] For mere information om de forskellige modes, se kap 7 i Handbook of Applied Cryptography

Da antallet af nøgler bliver på $(2^{56})^3$ pga. DES bliver afviklet tre gange, giver dette en tilstrækkelig sikkerhed, men til gengæld bliver det temmelig langsomt. Realiteterne har vist, at der ikke er nogen forskel på, om der bliver brugt to eller tre forskellige nøgler i trible DES, hvor en af de to nøgler bruges to gange, nemlig i første og sidste gennemløb.

2.3.1.3 Asymmetrisk kryptering med gennemgang af RSA

Man har udviklet et ***asymmetriske kryptosystem,*** som også kaldes for ***public key kryptosystem*** pga. de forskellige problemstillinger der findes for de ***symmetriske kryptosystemer***. I et ***asymmetrisk kryptosystem*** benytter man sig af to nøgler. Den ene nøgle er offentlig og bruges til kryptering, og denne deles ud til alle dem man skal kommunikere med. Den anden nøgle bruges til dekryptering og betegnes den private nøgle. Denne skal gemmes, så ingen andre kan få fat på den. En bruger i ***asymmetrisk kryptering*** skal altså have et nøglepar, én offentlig og én privat nøgle. For dette nøglepar skal der gælde, at data krypteret med den ene nøgle kun kan dekrypteres med den anden nøgle i nøgleparet. Asymmetrisk kryptering er altså en ressourcekrævende krypteringsalgoritme.

Et eksempel på et ***asymmetrisk kryptosystem*** er følgende; Alice vil sende Bob en hemmelig besked så hun finder Bob's offentlige nøgle og krypterer sin besked med den. Nu er det kun Bob, som kan dekryptere beskeden med sin hemmelige nøgle. Hvis K_h betegner den hemmelig nøgle og K_o betegner den offentlige nøgle, så virker kryptosystemet ved:

Kryptering: $E_{K_o}(m) = c$	*offentlig nøgle bruges til kryptering*
Dekryptering: $D_{K_h}(E_{K_o}(m) = c) = m$	*hemmelig nøgle bruges til dekryptering*

Der findes mange asymmetriske kryptosystemer, og den mest kendte er ***RSA***[56] (Rivers, Shamier og Adleman), hvilket blev udviklet af disse tre personer i 1977. RSA bruges både til kryptering og digital signatur. Man kan med ***RSA*** bruge nøgler af forskellige længder, og der gælder at jo større nøgler, des sikre bliver systemet. Styrken i ***RSA*** ligger i, at det er meget svært at faktorisere meget store tal, som er produkt af to store primtal. ***RSA*** er baseret på *numerisk teoretiske enheder af modulær aritmetik og primtal*, som benytter en funktionen der hedder ***Euler Totient funktionen***[57]. Man bliver nød til, at generere et ***RSA*** nøglesæt, før man går igang med kryptering eller dekryptering.

[56] Website: http://www.rsasecurity.com/
[57] Se Bilæg C

Generering af et *RSA-nøglesæt*[58] sker på følgende måde:

- Vælg to store primtal ***p*** og ***q***
- Beregn ***n = pq*** og ***φ(n) = (p-1)(q-1)***
- Find **d** sådan at **d** og ***φ*** ikke har fælles primfaktor, dvs ***gcd(d, φ(n))=1***
- Find **e**, så ***de % φ(n) = 1***
- ***d*** er den private nøgle
- ***n*** og ***e*** den offentlige nøgle

Efter nøglesættet er genereret, er det nemt at foretage en kryptering, det sker ved:

$$c = m^e \pmod n$$

Dekryptering sker ved:

$$m = c^d \pmod n$$

Kontrol af at ovenstående passer ser sådan ud[59]:

$$D_{k_h}(E_{k_o}(m)) = (m^e)^d \,(\mathrm{mod}\, n) = m^{ed} \,(\mathrm{mod}\, n) = m \qquad \text{for alle } 0<m<n.$$

Det der sker i ***RSA*** er, at man først skal bestemme nøglerne ***n, e*** *og* ***d.*** Dette gør man ved, at bruge to meget store primtal ***p*** og ***q*** (af flere hundrede cifre). Disse to store primtal skal holdes hemmelige, ellers kan systemet nemt brydes, da man vha. dem kan beregne ***φ(n)***. Så beregner man Eulers Totient funktion ***φ(n)= φ(pq)*** vha. de to primtal. Derefter bestemmer man ***d*** ud fra, at ***d*** skal være et relativ primtal til ***φ(n)***, **gcd(d, *φ(n)*)=1**[60] sådan at ***1<e< φ(n)***. Efter ***d*** *er blevet bestemt*, beregner man ***e*** vha. ***ed mod φ(n)=1***. Dermed har man de nøgler, der skal bruges og man kan gå i gang med kryptering og dekryptering. Her er et lille eksempel til hvordan RSA virker:

- Man starter med at vælge ***p = 2357*** og ***q = 2551***
- Så beregner man ***n = p*q = 2357*2551 = 6012707*** og ***φ(n) = (p-1)*(q-1)=6007800***
- Så vælger man ***e = 3674911*** sådan, at ***gcd(e, φ(n))=1***
- Så finder man ***d = 422191*** ved, at bruge ***ed = 1 (mod φ(n))***
- ***(n,e)*** nøglepar er den offentlige nøgle og ***d*** er den private nøgle

[58] Kapitel 8 i handbook of Applied Cryptography
[59] Beviset til denne ligning kan findes I kapitel 8 I handbook of Applied Cryptography
[60] gcd() er største fælles divisor, se bilag.

- Man kryptere og dekryptere en meddelelse ***m = 5234673*** ved følgende:
  - $c = m^e \bmod n = 5234673^{3674911} \bmod 6012707 = 3650502$
  - $m = c^d \bmod n = 3650502^{422191} \bmod 6012707 = 5234673$

Hvis man f.eks. bruger ***n = 6012707***, så kan man kode en besked der er mellem ***0*** og ***n-1***, nemlig ***6012706***. For at forstå, hvordan RSA helt præcist virker, så skal man kigge på lidt talteori, dette kan man se i bilag C.

RSA er ca. 1.000 gange langsommere end DES, men er derimod mere sikker end DES[61]. Sikkerheden af ***RSA*** afhænger af ***Eulers funktion φ(n)***, hvis man kender denne, så kan man nemt bryder ***RSA***. Den eneste måde, at finde ***φ(n)*** på er, at faktorisere ***n***, hvilket er meget svært eller umuligt med den nuværende matematiske viden. Der findes ingen effektiv måde, at faktorisere tal på i dag og selv om man bruger store og hurtige computer, så kan det alligevel tage en del år (hundrede/millioner år) for, at faktorisere et tal, alt afhængig af tallets størrelse. Man kan også sætte sig til, at gætte værdier for (p-1)*(q-1), men dette er ligeså svært som faktorisering.

[61] Kapitel 19 Applied Cryptography by Bruce Schneir, second edition..

2.3.2 Angrebsmetoder

Et angreb er, at en uønsket part prøver, at være med eller at ødelægge ens kommunikation, netværk eller software. Der findes flere forskellige typer angreb, der kan ramme ens maskine, kommunikation eller netværk, og der findes forskellige angrebsmetoder en angriber kan benytte sig af. Det største sikkerhedsmæssige problem i trådløse netværk (eller et overordnet netværk) er, at forbindelsen breder sig langt ud over de områder, man har kontrol over, da radiobølgerne, som de fleste trådløse netværk bruger, nemt går gennem vægge. Det betyder, at en angriber, som er interesseret i at få adgang til netværket, kan stå tæt ved sådan et trådløst netværk og opsamle følsomme informationer fra netværket.

For at et angreb finder sted, så skal der både være nogle sårbarheder og trusler i netværket, som angriberen formår at udnytte. En sårbarhed er en svaghed, og en trussel er en begivenhed eller omstændighed, der tillader at en sårbarhed kan udnyttes. Der findes to sårbarheder, interne og eksterne sårbarheder. Interne sårbarheder kan kun udnyttes af interne personer i systemet, mens eksterne sårbarheder også kan udnyttes af folk udenfor systemet. En angriber er ikke kun afhængig af, om sårbarhederne er interne eller eksterne, men er også afhængig af de ressourcer personen selv besidder, såsom viden, økonomi og udstyr.

En angriber der ikke ved, at en given sårbarhed findes, vil således oftest have svært ved at udnytte den. Omvendt kan en angriber med en meget stor viden måske endda finde frem til en hidtil ukendt sårbarhed og udnytte denne.

Der er to angrebsmetoder en angriber kan formå at benytte. Disse to angrebsmetoder er:

- ***Passiv angreb***: En passiv angriber opfanger de data, der bliver udvekslet mellem de to kommunikerende parter. Passive angreb er vanskellige at opdage, fordi sådanne angreb ikke modificerer på de oprindelige data, der bliver sendt mellem to kommunikerende parter. Passive angreb kan opdeles i endnu to typer af angreb, nemlig;

- *Opsnappelse af kommunikationsdata (Interception)*

  Her er to kommunikations parter A og B, som udveksler data, mens en uønsket part C får adgang til data, sendt fra part A til part B. Part B modtager også de afsendte data. Denne type angreb er illustreret på figuren nedenfor:

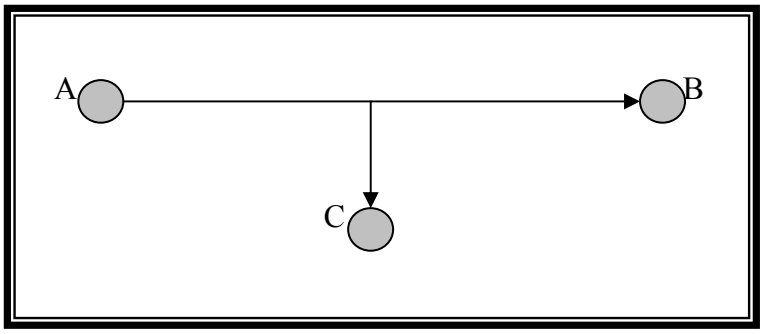


Figure 30: Interception, A og B er de to kommunikationsparter, mens C er den uønskede part

- *Trafikanalyse*

  En uønsket part C kan vha. en trafikanalyse indhente vigtige oplysninger om en kommunikation mellem de to kommunikationsparter A og B. Uønsket part C kan også vha. trafikanalysen få hyppigheden og størrelsen af de data, som bliver udvekslet mellem A og B. Et trafikanalyseangreb er ikke alvorligere end et Interception angreb, idet der i dette tilfælde aldrig ville kunne hentes mere information om en meddelelse mellem to parter end, hvad der ville have været tilfældet ved et interception-angreb.

- ***Aktive angreb***: En aktiv angriber kan gøre det samme som en passiv angriber, men en aktiv angriber kan ydermere forsøge at påvirke sender og modtager på forskellige måder. F.eks. ved at narre afsenderen til at sende en bestemt krypteret meddelelse eller data til modtager, eller kan ændre opsnappede sendte data eller meddelelse. De forskellige aktive angreb beskrives nedenunder:

- Angribere som angriber ens netværk kan f.eks. ødelægge de sendte data ved at forstyrre signalet, så de afsendte data ikke kommer frem, dvs. kommunikationen afbrydes (Interruption, hvor tilgængelighed af data bliver truet).

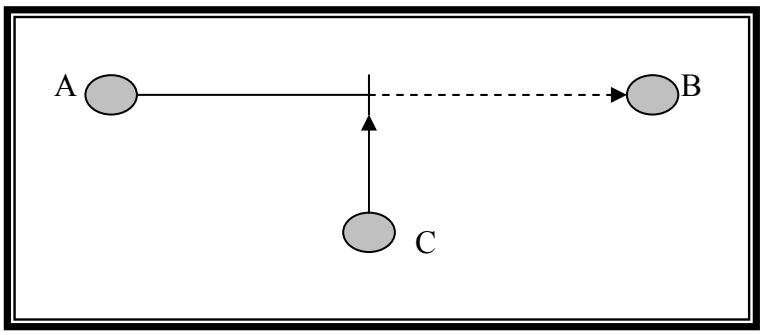


Figure 31: Interruption, A og B er de to kommunikationsparter, mens C er den uønskede part

- En angriber kan modificere (dvs. ændre) de opfangede data og sende dem videre (Modifikation, hvor integritet af data trues). Sådanne ændringer kan have store konsekvenser for sender og modtager.

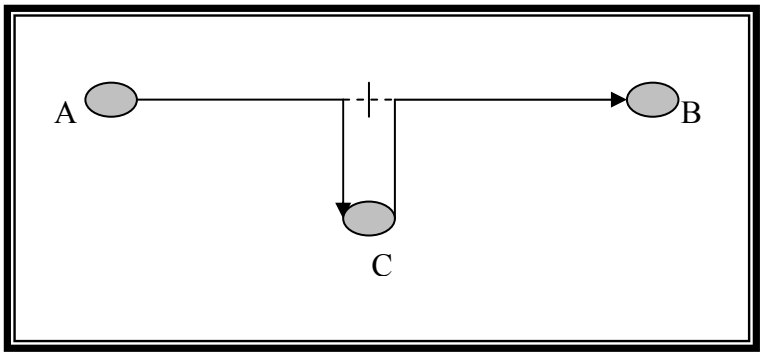


Figure 32: Modifikation, A og B er de to kommunikationsparter, mens C er den uønskede part

- Angriber kan sende helt nye data i netværket, uden at sender og modtager opdager det, dvs. angriberen kan spille ”senderen” (fabrikation).

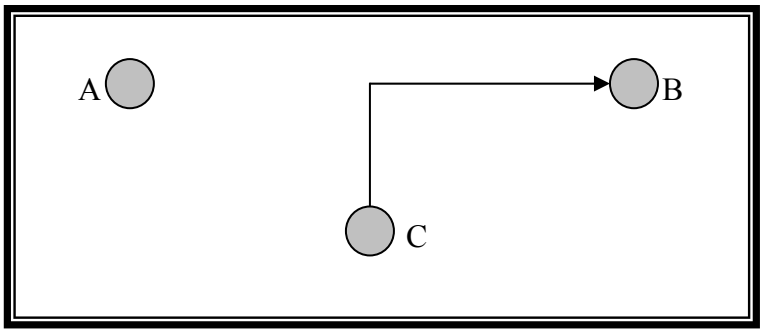


Figure 33: Fabrikation, A og B er de to kommunikationsparter, mens C er den uønskede part

Hvis det lykkedes en angriber, at opfange og aflæse fortrolige meddelelser/data (Interception), kan vedkommende udnytte dette alt afhængig af, hvilke parter der var involveret i kommunikationen og meddelelsens indhold. Det kan f.eks. tænkes udnyttet økonomisk eller politisk. Ligeledes er det også kun fantasien, der sætter grænsen for, hvad et Interruptions-, Modifikations- eller fabrikations-angreb kan tænkes at have af konsekvenser.

Men hvad kan der gøres for, at der ikke opstå succesfulde angreb på sit netværk? Man skal starte med, at finde og lukke de sårbarheder ens netværk har. Derefter kan man benytte en eller flere kryptosystemer til, at kryptere de afsendte data, så det bliver svært for angribere at se, hvad der sendes. I de næste par afsnit vil der blive gennemgået nogle af de vigtigste angrebsmetoder man kan komme ud for.

Der gennemgås følgende angrebstyper:

- Replay attacks
- Brute Force
- Man in the middle
- Denial of Service
- Matematiske angreb på selve algoritmen

2.3.2.1 Replay attacks

Replay attacks er angreb, hvor en angriber får fat i de afsendte data. Han kan enten bruge samme data og sender dem videre på et senere tidspunkt, eller han kan bruge nogle dele af opsnappede data og så sende dem videre til modtageren. Dvs. angriberen kan spille afsenderen overfor modtager uden, at nogle at de oprindelige parter opdager noget. Tager vi et eks., så kan der f.eks. være tale om en banktransaktion, hvor en bruger vil overføre penge fra en konto til en anden. Får angriberen fat i disse transaktionsdata, så kan angriberen sende samme data igen på et senere tidspunkt. Dette betyder, at forskellige mekanismer der er blevet brugt til, at beskytte fortroligheden og integriteten af data, stadig er gyldige og derfor skal der udvikles yderligere mekanisme til, at sikre at beskeden er ”frisk”, dvs. ikke har været sendt før.

Der er en del muligheder for, at undgå replay attacks. Man kan f.eks. bruge en Challenge/Response teknik, bruge et Sekvensnummer eller bruge et Time Stamp. De vigtigste metoder til, at undgå replay attacks bliver beskrevet i kapitel 3.4, hvor replay attacks og de forskellige metoder til at forhindre replay attacks bliver undersøgt nærmere. For mere information om replay attacks se kapitel 3.4.

2.3.2.2 Brute force

Brute force angreb er et ”kendt klartekst angreb” og går i al sin enkelthed ud på, at afprøve samtlige mulige kombinationer. Hvis man f.eks. har kendskab til hvordan en datapakke ser ud før og efter den er blevet krypteret, vil man krypterer datapakken med samtlige mulige nøgler fra en ende af. Hvis vi i vores eksempel for overskuelighedens skyld siger, at nøglelængden er på 8 bit, så vil man starte fra en ende af med nøglen 00000000, derefter 00000001, så 00000010, 00000011 osv. indtil 11111111 eller man har fundet den rigtige nøgle. Man siger som tommelfingerregel, at det kun er nødvendigt at gennemsøge halvdelen af nøglerummet, da der derefter er stor chance (50%) for at den næste nøgle man prøver med, er den rigtige.

Brute force angreb virker altid så længe man prøver alle mulighederne, men problemet er, at det tager for langt tid. Det er alment kendt, at mange mennesker vælger kodeord som på en eller anden måde kan relateres til personen, og derfor er disse kodeord lettere at bryde. Der findes offentlig tilgængelige ordbøger som f.eks. indeholder de mest anvendte engelske ord osv.

Det bedste man kan gøre for, at forhindre brute force angreb i at lykkes er, at vælger en nøglelængde, som er stor nok, da antallet af mulige nøgler (nøglerummet) stiger eksponentielt med forøgelsen af nøglelængden.

Hvis der bliver benyttet brute force angreb mod en DES algoritme, så afprøves alle mulige nøgler (2^{56}). I stedet for at bruge DES til kryptering, vil det være mere sikkert at bruge 3DES, da sikkerheden af 3DES er bedre end DES, pga. den udvidet nøglelængde (112/168 bits) og –rum ($2^{112}/2^{168}$). Det er muligt vha. brute force angreb, at bryde koden til DES pga. nøglelængden kun er på 56 bit. Citat af Lars R. Knudsen fra Matematisk Institut DTU: ” Det anbefales, at DES ikke mere bruges til hemmeligholdelse af data. I 1999 fandt "man" en DES nøgle på bare 22 timer ved at gennemsøge alle mulighederne én efter én”.

En anden mulighed er, at benytte en asymmetrisk krypteringsalgoritme fra f.eks. RSA, da selv om man kender den offentlige nøgle, så er det umuligt beregningsmæssigt at benytte brute force angreb til, at finde den private nøgle. Ressourcemæssige begrænsninger kan dog forhindre, at man kan bruge asymmetriske krypteringssystemer.

2.3.2.3 Man-in-the-middle

Her er der tale om et angreb, hvor en angriber sidder i et netværk, hvor sender A og modtager B udveksler meddelelser, og sniffer de sendte meddelelser. Angriberen overtager forbindelsen mellem de to kommunikerende parter A og B i netværket uden, at de opdager noget, og de bliver derfor ved med at tro, at de kommunikerer med hinanden (Interception). Angriberen kan vælge, at sende de sniffede meddelelser videre enten ændret eller uændret.

Der sker det, at en angriber X kommer ind i mellem A og B, de to kommunikerende parter. Protokollen starter med, at A sender sin offentlige nøgle til B, men dette bliver sniffet af angriber X, som sender sin egen offentlige nøgle til B. B svarer med, at sende sin offentlige nøgle som også bliver sniffet af X. X sender sin egen offentlige nøgle til A. Når A sender en meddelelse til B, dekryptere hun meddelelsen med den offentlige nøgle, som hun har fået og tror at den tilhører B, men som tilhører angriber X. X dekrypterer denne meddelelse med sin hemmelige nøgle, da den er blevet krypteret med hans egen offentlige nøgle. Så krypterer han igen meddelelsen med B´s nøgle og sender den videre til B (Her kan han vælge at ændre meddelelsen, hvis han vil). Når B svarer tilbage med en krypteret meddelelse, så kan X opsnappe denne meddelelse og dekryptere den, da den er krypteret med hans egen offentlige nøgle. Denne meddelelse eller en ny meddelelse kan sendes til A ved, at kryptere meddelelsen med A´s offentlige nøgle.

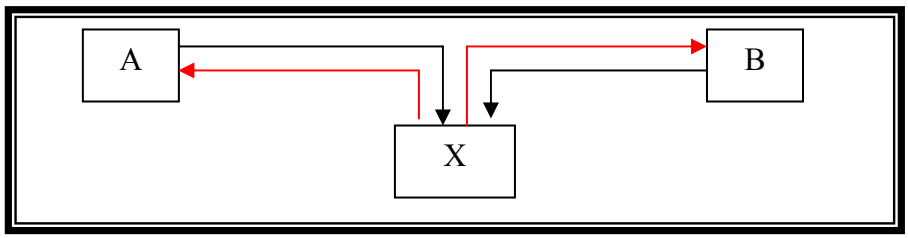


Figure 34: Man-in-the-middle angreb

Denne angrebsmetode kan undgås, hvis man benytter autentifikation. Derved kan de to kommunikerende parter A og B være sikker på, at de kommunikere med hinanden og ikke en anden, da det bliver gjort svært for en angriber, at være med i kommunikationen. Efter parterne har autentificeret sig med hinanden, så er det en god idé, at de benytter en eller flere krypteringsteknikker, så det bliver svært/umuligt for angriberen, at dekryptere de sendte meddelelser og dermed svært, at forstå indholdet af disse.

2.3.2.4 DoS (Denial of Service)

Ude-af-drift (DoS) angreb er et cracker[62] angreb, som har til formål at få ens maskine til at holde op med gøre det, den er sat til, dvs. det er et angreb mod tilgængelighed (*Availability*). Angriberen prøver at forhindre maskinen i at yde den service, den skal ved at overbelaste nettet eller maskinens software. Et virus angreb kan betragtes som en DoS angreb.

I et typisk DoS angreb sender en angriber fejlagtige data eller lignende til ens maskine/system, og overbelaster maskinen da den forsøger på at finde et fornuftigt svar. Sådanne slags overbelastninger får maskinen/systemet til at bryde sammen. Dette er knap så alvorligt som et reelt indbrud på systemet, men konsekvenserne er ofte af økonomiske og tidsmæssige tab for den angrebne part. Angriberen opnår som regel ikke en økonomisk gevinst ved sådanne angreb.

Et DoS angreb betagnes som en lidt anden type af angreb end de øvrige. Ved de øvrige angrebstyper forsøger man at udføre angreb i al ubemærkethed, hvilket er stik modsat af hvad der gør sig gældende i et DoS angreb. Oftest er det blot selve forbrydelsen, der i denne type angreb motiverer angribere, men andre gange kan angrebet spores tilbage til en bestemt politisk holdning hos angriberne, som bruger angrebet som et middel til at få sit budskab frem til offentligheden.

[62] Cracker, en person som bryder sikkerheden på et system. Læs mere på: http://catb.org/~esr/jargon/html/C/cracker.html

2.3.2.5 Angreb på algoritmen/implementeringen

I de foregående afsnit er nogle af de mest kendte og benyttede angrebsformer beskrevet. En ”ny” angrebsform, som er ved vinde større og større indpas, er angreb på selve algoritmen eller implementeringen og derfor gives der her en introduktion til disse.

Angreb på algoritmen går ud på at udnytte svagheder man har fundet i algoritmen, som man bruger til selve krypteringen. Resultatet af sådanne angreb kan have forskellige virkninger. Et eksempel[63] er ”svage” nøgler i DES. Her findes nøglepar hvor der gælder, at hvis man kryptere en klartekst med den ene nøgle, så kan man dekryptere cifferteksten med den anden nøgle og få den oprindelige klartekst. Dette gør sig gældende pga. den måde som DES genererer undernøgler på.

Flere og flere af de angreb som lykkes, sker ikke pga. svagheder i selve algoritmen, men pga. den måde algoritmen er implementeret i systemet. Et eksempel er generering af nøgler vha. tilfældige tal. Det er set, at de tilfældige tal ikke er så tilfældige igen som man havde håbet, hvilket bevirker en begrænsning af nøglerummet. Dette gør sig også gældende, hvis en eller flere bits i nøglen er låst fast til én fast værdi, eller hvis man på en anden måde kan bestemme værdien via andre parametre. En anden måde er, at præge de inputs som bruges til tilfældighedsgeneratoren.

[63] Kilde: Bruce Schneier, Applied Cryptography, kapitel 12.3 “Security of DES”

2.3.3 Opsummering af sikkerhed

Der er i foregående afsnit behandlet og beskrevet de vigtige emner i sikkerhed, kryptering og angreb. I sikkerhedsafsnittet blev forskellige sikkerhedsmål beskrevet, som skal være opfyldt for at sikre sig mod trusler og angreb. Der blev set på hvilke hensyn der skal tages til datafortrolighed for, at sikre sig mod at uvedkommende får adgang til dem. Dette kan sikres ved, at bruge autentifikation og autorisation[64]. Endvidere blev dataintegritet belyst, som sikre at data ikke kan ændres undervejs uden at det kan ses. Datatilgængelighed sikre, at systemer og data fungerer hele tiden og er tilgængelig for autoriserede personer. Disse sikkerhedsmål gælder for Z-Wave, så kommunikationen mellem noder kan foregå sikkert.

Man kan med kryptering opfylde ovenstående sikkerhedsmål, så derfor er forskellige krypteringbegreber blevet gennemgået. Begreberne er: autentifikation (MAC), symmetrisk kryptering (DES) og asymmetrisk kryptering (RSA).

Autentifikation bestemmer sikkerheden for identifikationskorrekthed, dvs. man identificerer sig, når man vil have adgang til data eller et system, det afgøres om man skal have adgang eller ej. Ved at bruge MAC kan man sikre dataintegriteten, så data ikke kan ændres undervejs uden, at det opdages. Man kunne tilføje MAC til meddelelserne, men det giver ingen yderligere beskyttelse overfor, at tredjepart ikke kan læse meddelelserne som sendes frem og tilbage kun, at de ikke ubemærket kan ændres undervejs. Symmetrisk og asymmetrisk kryptering er metoder til, at sikre at meddelelser ikke kan læses af andre og de mest kendte/benyttede metoder (DES og RSA) er gennemgået. Endvidere er der givet en kort beskrivelse af 2DES og 3DES, da disse forøger sikkerheden af DES og der bruges 3DES i Z-Wave systemer, da en asymmetrisk krypteringsalgoritme (f.eks. RSA) er for ressourcekrævende.

I slutningen af kapitlet er forskellige angrebsmetoder og –typer beskrevet. Gennemgangen af angrebsmetoderne berettiges af, at de kan forekomme i Z-Wave netværket.

[64] Se afsnit 2.3 Sikkerhed

3.0 Design

I de efterfølgende afsnit vil de forskellige løsningsforslag blive gennemgået. Løsningernes fordele og ulemper gennemgås, og hvis nødvendigt forklares selve inkluderingsproceduren, som løsningen skal bruges i.

Komponenterne der bruges til løsningsforslagene er standard komponenter, som kan findes hos enten RS Components (http://www.rsonline.dk) eller hos Farnell InOne (http://www.farnellinone.dk). Løsningerne baseres på kommercielt tilgængelige komponenter, da kostprisen er en vigtig faktor.

Løsningsforslagene er overordnet grupperet, så vi først gennemgår de løsninger som er in-band, dvs. hvor det er den normale RF kommunikationskanal der bruges. Efter disse ser vi på en fabriksprogrammeret løsning. Den sidste overordnede gruppe er de løsninger som er out-of-band, dvs. at nøgleudvekslingen foregår på en anden måde end ved brug af RF. Vi starter med at analysere de løsninger hvor brugeren skal gøre noget aktivt så som, at dreje eller trykke på knapper. Afslutningsvis gennemgås de løsninger, hvor brugeren ikke skal gøre noget særlig aktivt andet end at holde enhederne tæt på hinanden.

For ikke at skulle beskrive alt for mange ”eksotiske” forslag af implementeringer samt anvendelse af disse, har vi besluttet os for, at holde os til en fast nøglelængde. Set fra brugersiden (brugervenlighed) er det mere intuitivt, at alle nøgler har samme længde, dvs. den samme handling skal foretages samme antal gange uanset ”udseende” af nøglen. Dvs. der er ikke forskel på antallet af handlinger brugeren skal foretage uanset om nøglen er 34 eller 169765234. Løsningsforslag som pga. anvendelsesmåden sætter begrænsninger på nøglerummet, er heller ikke medtaget, da vi mener det kan være for kompliceret, at skulle holde styr på dette.

Z-Wave secure løsning skal bruges til sikre noder, dette kan f.eks. være rumfølere, som registrerer bevægelse i lokalet og giver besked til en central alarmeringsenhed. Dette sætter begrænsning på hvor meget løsningen må fylde fysisk set, og vi går ud fra, at jo mindre det fylder, des bedre. Der kan være yderligere designmæssige krav så som, at tilgangen til komponenten skal foregå bag en klap.

Beskrivelserne af hvert løsningsforslag er bygget op på samme måde, for at lette overskueligheden. Opbygning af beskrivelserne er som følger:

- **Løsningsbeskrivelse** – Her gives en overordnet forklaring af løsningen. Det har til formål, at give et overblik over løsningsforslagene, så man kan se hvad det drejer sig om.
- **Beskrivelse af hardware** – En oversigt over komponentens vigtigste egenskaber listes. Oversigten vil typisk bestå af en kort beskrivende tekst, forventet holdbarhed, antallet af funktioner, fysiske mål, pris samt varenr. Mht. prisen, så er det prisen på den primære komponent i løsningen der vises. Ang. varenr., så angiver F, at komponenten findes hos Farnellinone, og R angiver RS-Components. Flere af komponenterne findes hos begge forhandlere, så her er den billigste angivet.
- **Opstilling af hardware** – Her vises hvordan hardwaren/komponenterne skal forbindes til Z-Wave modulet for, at nøgleudvekslingen kan finde sted. Ved nogle forslag vil der være flere muligheder for opstillinger.
- **Generelt** – Dette afsnit medtages når der findes generelle ting, som gælder for alle løsningsforslagene i samme kategori. Det kan f.eks. være en beskrivelse af en anderledes/mere optimal tilslutning af hardwaren til Z-Wave modulet.
- **Løsning 1,2,3,...** – I disse afsnit beskrives de forskellige løsningsforslag. Der beskrives hvad brugeren skal gøre for at udføre nøgleudvekslingen og diverse udregninger, som bla. vises i løsningsoversigten, udregnes.
- **Løsningsoversigt** – Denne oversigt viser nøgletal for de forskellige løsninger. Oversigten skaber et nemt overblik over hvilke løsninger, der er mere favorable end andre. I oversigten vil der typisk være listet hvor mange funktioner den pågældende komponent/løsning bruger, hvor mange handlinger brugeren skal foretage, og hvor ”effektiv” løsningen er via et mapnings-tal. Mapningstallet angiver hvor mange handlinger brugeren skal udfører i forhold til nøglens længde. Jo større værdi, des mere effektiv er løsningen. Der henvises til bilag D, hvor begrebet gennemgås nærmere.

De fleste løsninger følger ovenstående opstilling, men der er løsningsforslag, hvor det ikke har været muligt at finde de tekniske detaljer, så der gives i stedet en kort gennemgang.

3.1 Karakteristika for Secure Z-Wave

I dette afsnit beskrives først de tekniske karakteristika for ZW0102 (chippen som sikkerhedsløsningen laves for), derved fås et overblik over de tekniske muligheder. Derefter gennemgås de karakteristisk som gælder for Z-Wave's secure løsning.

Arkitekturen i chippen er baseret på en 8051 mikroprocessor med en klokfrekvens på 7,376974 MHz. Den fysiske udformning af chippen er en TQFP[65] med 52 ben, som måler 10 x 10 mm. I chippen er der 16-20 kB Flash hukommelse for OEM software alt efter hvilken applikation den kører, og der er adgang til 1 kB SRAM. I chippen er der implementeret DES/3DES hardware til kryptering, og ved brug af radioen er der også implementeret en ægte tilfældig tal generator. En sidste hardware implementering som chippen besidder, er en RTC[66] til nøjagtige tidsmålinger.

Chippen har et lavt strømforbrug. Forbruget når der sendes ved 4 dBm er 34 mA og når der modtages, bruges der typisk 20 mA. Enheden kan sættes i en power down mode (vågner ved reset) og har en lavt strømforbrug på kun 1 uA ved 25°C. Den er garanteret at være operationsdygtig inden for temperaturområdet -40 til 85°C.

Af kommunikationsmuligheder til chippen, findes der foruden radioen følgende muligheder: Standard 115 kbps RS232 UART[67], 4 analoge I/O ben med en 10 bit ADC[68] og maksimal sampling rate på 20,96 kHz og 11 digitale I/O (nogle med ekstra funktionaliteter).

Radioen opererer ved 868,42 MHz i EU og ved 908,42 MHz i USA og overfører data med 9,6 kbps. I EU er der krav på en dutycycle på 1%, dvs. her må der maksimalt sendes 3,6 kB/time. Modtageren har en høj følsomhed på –96 dBm. Sendestyrken i radioen er programmerbar fra -20 til 4 dBm (0,01 til 2,5 mW).

[65] TQFP - Thin Quard Flat Pack, "Chiphus" som er velegnet til chips med svage strømme, lille vægt og lav højde
[66] RTC – Real Time Clock, decideret hardware som holder styr på tiden, så selve kernen ikke belastes med dette
[67] UART - Universal Asynchronous Receiver-Transmitter - den gængse seriel port på enhver computer
[68] ADC - Analog to Digital Converter, bruges til at konvertere et analogt signal til en digital talværdi i mikroprocessoren

Vi har set på de tekniske detaljer for chippen og beskriver nu de krav/realiteter, som skal gælde for Secure Z-Wave.

Z-Wave secure enheder skal kunne installeres af både installatører og private, dvs. dem som sætter enheder op har forskellige niveauer af færdigheder. Da man ikke kan undgå, at have enheden fysisk i hånden når den installeres/sættes op, må vi betegne det som en kort afstand (1 - 5 m) når enhederne skal inkluderes i netværket. Det gør det muligt at bruge out-of-band[69] kommunikation ved inkluderingen.

Data pakkerne der sendes via RF, må betegnes som små, da de har en størrelse på mellem 10 og 64 bytes (typisk 20). Z-Wave teknologien bruges til kontrol og ikke til streaming, så data trafikken er meget lav. Da det er ”standard” pakker der sendes frem og tilbage, er variationen af indholdet lille, og for at opnå stor interoperabilitet, dvs. at alle enheder skal kunne snakke med alle, er formatet af de ukrypteret pakker ”offentlig kendte”. Der er mulighed for at præfabrikere modulerne fra producentens side, så evt. parametre/værdier bestemmes her.

Det er muligt at jamme/umuliggøre signalet/kommunikationen uden at det opdages, da ingen enheder kan sende og lytte samtidig. Desuden kan normal støj genere signalet, og det kan derfor være umuligt, at se forskel på om det er direkte jamming der foretages, eller om det er normal baggrundsstøj.

Der søges en fordelagtig kostpris struktur, dvs. at merudgifterne primært skal forsøges at blive lagt på kontrollerne frem for noder. Grunden til dette er, at noder typisk er højvolume produkter, hvorimod kontrollerne typisk er lavvolume produkter med en i forvejen relativ høj kostpris. Kostprisen er en vigtig parameter og derfor er der i noderne begrænsede ressourcer som f.eks. RAM, Flash, regnekraft osv.

[69] In-band er hvis RF bruges til kommunikation, Out-of-band er når kommunikationen foregår på en anden måde

3.2 Initial nøgleudveksling

Til Z-Wave secure løsning skal trafikken mellem noderne være sikret, så tredjepart ikke kan se hvad der foregår. Der er i kapitel 2 bla. gennemgået hvilke forskellige angrebstyper, som netværket kan udsættes for. Det er gjort klart, at for at sikre netværket, så er det nødvendigt at kryptere trafikken.

For at noderne i netværket kan kryptere/dekryptere trafikken, er det nødvendigt at de bruger samme nøgle. Pga. ressourcemæssige begrænsninger er det ikke muligt, at bruge asymmetrisk kryptering da det vil være for ressourcekrævende, som angivet i afsnit 2.3.3 "Opsummering af sikkerhed".

Vi ser på et scenarium, hvor hele netværket bruger symmetrisk kryptering, og hvor hele netværket bruger den samme nøgle. Dette medfører en lav administration af nøgler og det sikre, at alle noder kan kommunikere med hinanden uden, at der skal foretages en nøgleudveksling, som vil skabe et overhead på trafikken. Forskellige krypteringsalgoritmer er sammenlignet nedenfor[70]:

Algoritme	Hastighed
DES	81 Mbps
IDEA	80 Mbps
RC4	439 Mbps
TwoFish	204 Mbps
AES	241 Mbps
Crypticore	973 Mbps

Table 4: Sammenligning af forskellige Krypteringsalgoritmer

Zensys har implementeret DES/3DES i hardware i ASIC'en, så når der skal foretages kryptering/dekryptering, belaster det ikke processoren yderligere. DES/3DES er implementeret, da algoritmen ikke kræver licens, ikke kræver mange ressourcer, er sikker (3DES) samt er hurtig nok til den lave båndbredde der bruges i et Z-Wave netværk.

[70] Målingerne er foretaget på en Pentium III 450 MHz. Kilde: http://www.cryptico.com/Files/filer/produktoversigt.pdf

Problemstilling er her, hvordan nye noder får den nøgle, som resten af netværket bruger. Man vil gerne undgå, at sende nøglen i klartekst til den nye node, da det derved vil være nemt at aflytte trafikken fremover for hele netværket. Har tredjepart først fået fat i den fælles netværksnøgle, findes der ingen måde til at "fjerne" denne igen. Dette er et problem i Bluetooth[71], som man gerne vil undgå i Z-Wave.

Som beskrevet tidligere, så er er DES/3DES hardware implementeret i ASIC'en, så spørgsmålet er hvilken der skal bruges. Bruges DES er nøglelængden på 56 bits og bruges 3DES er nøglelængden på 112 bits. Sikkerheden i 3DES er væsentlig bedre end DES pga. det større nøglerum, men spørgsmålet om det favorisere 3DES frem for DES. Der har været foretaget brute force angreb, hvor det er lykkes at bryde DES, hvilket ikke er tilfældet for 3DES, men det har været distribueret angreb som har krævet mange computere.

I overvejelserne til om hvilken krypteringsform der skal bruges, skal man huske på tommelfingerreglen: "Sikkerhed er omvendt proportionalt med brugervenlighed.". Dvs. øger vi sikkerheden ved, at bruge en dobbelt så stor nøglelængde og derved bruge 3DES, "halveres" brugervenligheden. Det vil altså være mere besværlig for brugeren, at få inkluderet en ny node i netværket, da risikoen for fejl forøges. For det første er nøglelængden dobbelt så stor, så risikoen for fejl er også dobbelt så stor, og for det andet falder koncentrationen jo mere tid brugeren skal bruge. Endvidere er der begrænsninger på, hvor meget fysisk plads der må bruges til overførelsen, hvilket også kan være en faktor til, at fejlprocent forøges.

Pga. ovenstående tages der udgangspunkt i, at der bruges DES til den initiale nøgleudveksling og derefter 3DES til den videre kryptering, dvs. brugeren skal overføre/fortælle noden en 56 bits nøgle. De efterfølgende afsnit beskriver forskellige løsningsmetoder til, hvordan den initial nøgleudveksling kan foregå, hvorefter kapitlet afsluttes med en opsummering.

[71] Kilde: http://www.computerworld.com/mobiletopics/mobile/story/0,10801,89495,00.html?nas=MW-89495

3.2.1 RF (Software)

Løsningsbeskrivelse:

Den simpleste/nemmeste måde for en node, som lige er blevet inkluderet i netværket, at få netværksnøglen er ved, at denne sendes lige efter den normale inkluderingsprocedure er gennemført. Da noden på dette tidspunkt ikke har noget tilfælles med kontrolleren, dvs. at de ikke deler nogen form for hemmelighed, som andre ikke kan få fat i, er det ikke muligt at lave en sikker nøgleudveksling.

Problemet ligger i, at enhederne (noden og kontrolleren) ikke har mulighed for, at verificere det der bliver modtaget. Enhederne har ikke mulighed for, at verificere at sender og modtager er dem, som de udgiver sig for at være. Man kunne argumentere for at enhederne kunne sende oplysninger frem og tilbage, som kunne bruges til at lave en midlertidig nøgle. Det hjælper dog ikke noget for, hvis noden kan generer en midlertidig nøgle ud fra noget tilsendte data, så kan en hvilken som helst anden node (dvs. tredjepart) også gøre dette, og derved være i stand til at aflytte hele ”samtalen”.

Vi går ud fra, at inkluderingsproceduren og den efterfølgende nøgleudveksling er kendt, da gængs sikkerhedspolitik siger, at sikkerhed ikke må baseres på en fælles hemmelighed.

3.2.2 RF (Lav sendestyrke)

Løsningsbeskrivelse:

Som opgivet i afsnit "3.1 Karakteristika" så kan sendestyrken på enhederne indstilles til at sende med mellem –20 og 4 dBm uden ændringer af hardwaren. Enheder som bruges i USA sender normalt med –5 dBm og i EU sendes der med 1 dBm. Forskellen i sendestyrken skyldes regionale bestemmelser, som enhederne skal opfylde. I inkluderingsprocessen, som er gennemgået i afsnit "2.2.7.4 Inkluderingsprocessen i Z-wave", skrues der ned for sendestyrken med 1 dBm, når den nye node skal undersøge hvilke andre noder den kan se. Dette sker for, at sikre at noder som ligger lige på grænsen ikke medtages, da man heller vil have en liste med "gode"/sikre noder frem for en liste med "dårlige"/usikre noder.

Princippet med at kunne skrue ned for sendestyrken så rækkevidden mindskes og færre hører signalet, kan vi bruge i nøgleudvekslingen[72]. En test lavet for at måle på pålideligheden af Z-Wave netværket uden retransmission, har givet følgende resultat[73]:

Afstand	Frames med fejl	Fejlrate
10 meter	1	$1{,}9*10^{-6}$
20 meter	5	$9{,}6*10^{-6}$
30 meter	8	$1{,}5*10^{-5}$

Table 5: Pålidelighed af Z-Wave netværk

Testen viser, at der er en meget lav fejlrate selv på 30 meters afstand, og i praksis er rækkevidden op til 50 meter. I vores problemstilling er vi interesseret i det modsatte nemlig, at skrue så meget ned for styrken så ingen andre (eller så få som muligt) hører, hvad der sendes frem og tilbage.

Tager vi udgangspunkt i at det er muligt for normale moduler, at kommunikere med hinanden i en afstand på 50 m med en sendestyrke på –1 dBm, kan vi hurtigt give et overslag på afstanden, hvis vi skruer ned på –20 dBm. En grundregel er, at for hver 6 dBm et RF signal mindske, så halveres afstanden. Da vi kan sænke vores sendestyrke fra -5 til –20 dBm, vil afstanden mindskes med en

[72] Der henvises til Bilag A, som er et patent, der går ud på at reducere sendestyrken i Bluetooth enheder under den initiale nøgleudveksling.
[73] Kilde: 903900101, Z-Wave Reliability Application Note, February 2003

faktor 2,5 [(-5 – (-20))/6] dvs. vi kommer ned på omkring 20 meter (50 meter/2,5). Denne afstand er for stor til at vi kan bruge den til noget brugbart.

Som før omtalt opgives sendestyrken til at kunne dæmpes ned til –20 dBm direkte i den eksisterende software. Med få ændringer i softwaren kan vi dog komme ned på omkring – 50 dBm. Dette sker ved at to forstærkere på RF-udgangen slukkes fuldstændigt og ikke kører på et minimum, som ved de –20 dBm. Hvis dette bruges kan vi få rækkevidden reduceret med en faktor 9 [(-5 – (-50))/6], hvilket giver en afstand på 5½ meter (50 meter/9). Dette gælder for normale moduler, men der er intet der afholder tredjepart fra, at ombygge et modul med forstærkere samt bruge en retningsbestemt antenne. Denne løsning med reduceret sendestyrke kan ikke betegnes som værende sikker, da der er mulighed for tredjepart aflytter trafikken, selvom om løsningen besværliggøre opsnapning af nøglen.

3.2.3 RF (Retningsbestemt)

Løsningsbeskrivelse:

Når RF radiobølger skal udbredes i luften, sker det via en antenne. Antennen skal afpasses til den bølgelængde som man ønsker at modtage/sende ved for at få det mest optimale resultat. Derved minimeres støj og man får det kraftigste signal ud af det. Foruden antennens størrelse, har udformningen en stor rolle, hvad angår udbredelsen af radiobølgerne.

Ser vi på antenne typer, som skal laves på print, så er der ikke så mange valgmuligheder. Den mest benyttede type er en monopol. Denne type bruges i øjeblikket på Z-Wave slavemoduler. udformningen kan karakteriseres som én lang leder, med den ene ende forbundet til selve signalgeneratoren. Lederen kan f.eks. være en "metalpind", som det ses på transistorradioer, men den kan også være en printbane. Udbredelsen af radiobølgerne kan groft sagt antages at ske sfærisk, dvs. styrken/modtagelsen er lige kraftig/god i/fra alle retninger. En anden anvendt type er en dipol. En dipol har samme udformning som en monopol, men her "fødes" antennen med signalet midt på, i stedet for en af enderne som ved monopolen. Nedenfor ses udbredelsen i 2D og 3D[74]:

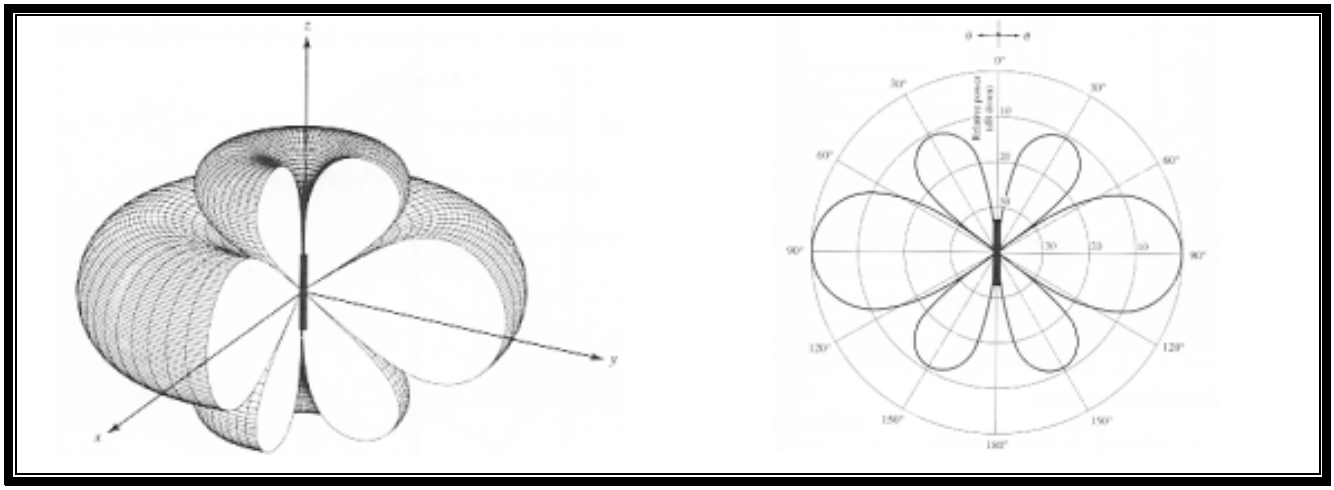


Figure 35: Udbredelse af radiosignaler i 2D og 3D

Det er meget svært at retningsbestemme radiobølger uden brug af horn eller reflektor, og disse løsninger er dyre og optager plads, som vi ikke mener, vi kan forlange/gøre krav på. Det vil være muligt, om end besværligt for tredjepart, at opfange signalet med udstyr med retningsbestemt antenne og kraftige forstærkere.

[74] Kilde: Antenna Theory - Analysis and Design, Second Edition by Constantine A. Balanis

3.2.4 Forudprogrammeret nøgle

Løsningsbeskrivelse:

Enheden er programmeret fra fabrikken/producenten med en nøgle, som aldrig udskiftes og som bliver i enheden i resten af dens levetid. Der foretages ingen udveksling af initial nøgle mellem node og kontroller i de to første løsningsforslag, da inkluderingen foregår i et "beskyttet" miljø.
I løsning 1 og 2 kan der bruges 112 bit nøgle uden, at det ændrer på brugervenligheden, men dette er ikke tilfældet for løsning 3. Der skal ikke til nogle af forslagene bruges ekstra hardware.

Løsning 1:

Denne løsning baserer sig på én fællesnøgle. Udgangspunktet er en security løsning som kun bruges til alarmudstyr og at den enkelte forbruger har indgået en servicekontrakt med et alarmselskab. Dvs. når/hvis der skal sættes nye enheder op eller gamle skal udskiftes, så kommer der en installatør fra firmaet og foretager installationen.

Vi forudsætter at alarmselskabet køber færdig lavet enheder/sensorer hos en producent (f.eks.Visionic Ltd. Israel[75]). Hos denne producent bestiller alarmfirmaet et stort antal sensorer som alle får den samme initiale nøgle brændt ned. Dvs. at alle enheder, hos alle kunderne der bruger det samme alarmselskab, har den samme initial nøgle. Dette gør det nemt at installere, da enhederne kan inkluderes i netværket på sædvanlig vis uden, at der skal foretages noget ekstra fra installatørens side, da den samme nøgle også ligger i kontrolleren som installatøren bruger.

Ulempen ved denne løsning er, at alle netværkenes sikkerhed baseres på én fælles hemmelighed, hvilket er i direkte modstrid med nutidens sikkerhedspolitik. Da alle enhederne har den samme initiale nøgle, vil netværket på ingen måde være sikkert, hvis bare én enhed bliver kompromitteret. Sker dette, kan tredje mand få kontrol over netværket, da denne hele tiden vil kunne følge med i hvad der bliver sendt frem og tilbage.

[75] Website: http://www.visonic.com/

Fremgangsmåden for tredjemand der kender til den fælles hemmelighed (initial nøgle) er:

1. Tredjemand sætter en sensor ud af drift enten ved at ødelægge den eller på anden måde gøre den utilgængelig for netværket.
2. En installatør bliver tilkaldt for at skifte enheden.
3. Når skiftet sker, følger tredjemand med ved at overvåge al trafikken.
4. Den nye enhed får tildelt den nye nøgle som resten af netværket bruger og alt dette kan tredjemand se (man-in-the-middle attack), da denne kender den initiale nøgle.

Da alle enheder får den samme initial nøgle, vil der ikke være nogen meromkostninger med denne løsning.

Løsning 2:

Ovenstående løsning tog udgangspunkt i at alle enhederne hos slutbrugerne har den samme initiale nøgle, da det er producenten af enhederne, der brænder nøglen ned. Ændrer vi på forudsætningerne, så alarmselskabet selv kan brænde nøglen ned eller bestille enheder med en specifik nøgle, er det muligt at give kunderne forskellige nøgler. Man kan tænke sig, at hver kunde så at sige får sin egen initiale nøgle, dvs. at alle enheder hos den samme kunde har samme initiale nøgle. De kundespecifikke nøgler ligger i en database hos alarmselskabet.

Når en installatør bliver tilkaldt for at sætte nye enheder op, brænder han den kundespecifikke nøgle ned i enhederne samt i sin egen kontroller. Man kan sagtens forestille sig, at han lægger nøgler ned i flere enheder til flere kunder, og at nøglerne også bliver lagt ned i kontrolleren. Når installatøren kommer ud til kunderne, har han mulighed for på sin kontroller at vælge hvilken kunde han er ude hos, derved ved kontrolleren hvilken nøgle der skal bruges i inkluderingsprocessen.

Ligesom i løsning 1 sker der hér heller ingen udveksling af den initiale nøgle ude hos kunden. Fordelen ved denne løsning frem for løsning 1 er, at tredjepart ikke kan bruge kendskab fra en initial nøgle fra ét netværk på et andet, da de er forskellige. I forhold til løsning 1 må meromkostningen her være højere, da der nu skal brændes forskellige nøgler ned i enhederne, og dem skal der holdes styr på. Derved er der ikke længere tale om en ensartet/standard vare.

Løsning 3:

En tredje mulighed er, at producenten/alarmselskabet giver hver enhed en ny/unik nøgle. Man behøver ikke kunne garantere at nøglen er unik, da sandsynligheden for at tredje part kan finde de enheder med samme initiale nøgle må siges at være ikke eksisterende, da der findes mere end 7,2 * 10^{16} mulige nøgler (ved 56 bits). Sammen med noden vedlægges et certifikat e. lign. hvor den 16 cifferet nøgle står på. Når noden skal inkluderes i netværket, skal brugen indtaste nøglen på kontrolleren, så den kan sende noden den nøgle som resten af netværket bruger.

I stedet for et certifikatet som brugen skal gemme, kunne det også tænkes at nøglen blev udskrevet på en label, som blev klistret på noden. Derved skal brugeren ikke holde styr på hvor certifikaterne er, samt til hvilken node det enkelte certifikat hører til. Desuden kan nøglen aflæses (og indtastes) med det samme, når man står med noden i hånden.

Løsningen må siges at være meget brugervenlig men sætter store krav til producenten. Ved produktionen/programmeringen, skal man sørge for at det er det rigtige certifikat/label der følger den enkelte node. Bliver der byttet ud på enhederne/certifikaterne under produktionen/programmeringen gøres enhederne ubrugelige, da de ikke kan inkluderes i et netværk. Da den initiale nøgle ikke kan hentes ud af chippen, skal enhederne sendes tilbage til omprogrammering for at få en ny nøgle samt certifikat/label. Ligesom løsning 2 må omkostningen her forventes at være højere end for "normale" enheder, da der skal holdes styr på certifikater/labels.

Løsningsoversigt:

#	Stk. pris (kr.)	Unik nøgle	Udskiftning af nøgle mulig	Installatør installation*
1	-	Nej	Nej	Ja
2	-	Kundeunik	Nej	Ja
3	-	Ja	Nej	Nej

Table 6: Løsningsoversigt ved forudprogrammeret nøgle

Stk. pris – Det er ikke muligt at udregne en styk pris, da der ikke skal tilføjes ekstra hardware til modulet. Ekstraomkostningen ligger ved programmeringen.

*** -** Der ses bort fra, at lovgivning kan begrænse hvad private må installere.

3.2.5 Ledning

Løsningsbeskrivelse:

Alle noder og kontrollere har ét stik som bruges til at overføre den initiale nøgle med. Noden, der skal inkluderes i netværket, forbindes med en ledning til en kontroller, som overfører nøglen resten af netværket bruger. Der er ikke brug for nogen initial nøgle, da opsnapning af data fra kablet, må betegnes som umuligt.

Beskrivelse af hardware:

3,5 mm Jack stik og fatning:

3,5 mm 2 polet Jack stik og fatning

Fatning: Indkapslet chassisfatning med forsølvede sluttekontakter.

Stik: Isoleret, men ikke afskærmet.

Fysiske mål Fatning: B: 9 mm H: 10,5 mm D: 20 mm

Fysiske mål Stik: L: 11 mm D: 52 mm

Varenr. (Fatning/Stik): RS 454-233/449-994

Pris (Fatning/Stik): 3,50/3,65 kr.

Modular stik og fatning:

4/4 stik og fatning har 4 kontaktpositioner med 4 kontakter isat

Fatning: Lav-profil type for horisontal printmontage

Stik: Kontakter er af forgyldt fosforbronze. Der skal bruges fladt telefonkabel.

Fysiske mål Fatning: Ikke opgivet

Fysiske mål Stik: Ikke opgivet

Varenr. (Fatning/Stik): F 153-084/RS 143-0042

Pris (Fatning/Stik): 3,17/0,63 kr.

Strømforsyningsstik og -fatning:

2,1 mm 2 polet strømforsyningsstik og fatning

Fatning:

Stik:

Fysiske mål Fatning: B: 11,0 mm H: 14,5 mm D: 14,3 mm

Fysiske mål Stik: L: 21,8 mm D: 8,0 mm

Varenr. (Fatning/Stik): RS 486-662/486-628

Pris (Fatning/Stik): 4,90/4,00 kr.

DIN-stik og fatning:

2 polet DIN stik og fatning

Fatning:

Stik:

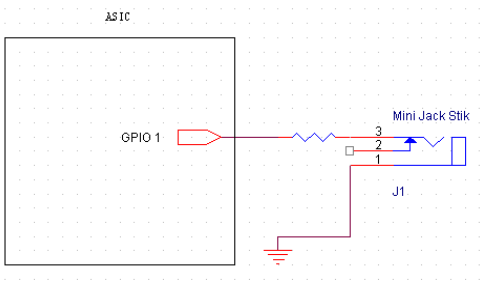
Fysiske mål Fatning: B: 17,6 mm H: 29,0 mm D: 15,5 mm

Fysiske mål Stik: L: 28,5 mm D: 15,5 mm

Varenr. (Fatning/Stik): F 150-852/150-851

Pris (Fatning/Stik): 2,90/4,00 kr.

Opstilling af hardware:



ASIC er forbundet til mini jack stikket via en modstand, for at beskytte indgangen på GPIO' en.

Figure 36: Viser ASIC tilsluttet til stik

Generelt:

På figuren som viser opstillingen af hardware, kan det ses at den GPIO som er tilsluttet til stikket på noden, også kan fungere som Ekstern Interrupt. En node der ikke er inkluderet i netværket er i dvale mode og forbruger derfor minimal strøm. Når noden skal inkluderes i netværket forbindes den via en ledning til en kontroller. Kontrolleren sættes i inkluderingstilstand og hiver derfor forbindelsen til noden høj. Dette bevirker at noden bliver vækket, og efter lidt tid, som noden skal bruge til at vågne, er den klar til at modtage den endelige nøgle.

Det anbefales at ledningen mellem de to enheder holdes kort for minimering af støj. Den eneste forskel der er på løsningsforslagene, er de fatninger og stik der bliver brugt. Man må generelt sige, at disse løsningsforslag er meget brugervenlige, da der bare skal forbindes en ledningen mellem enhederne og derefter skal kontrolleren have at vide, at den skal inkludere en node. Det skal dog bemærkes, at der skal være adgang til stikket efter eller under montagen af AC powered enheder. Placeringen af enheden kan evt. også besværliggøre montage af ledningen.

Løsning 1:

Løsning 1 bruger mini Jack stik til at overføre nøglen til noden. Mini Jack stik bruges primært indenfor audio området, så som headsets til mobiltelefoner. Skulle en bruger få den idé at tilslutte et headset til noden, vil der ikke ske noget, da ingen af enheder udsender strøm. Det samme gælder for kontrolleren, da der kun sendes en strøm afsted, når der inkluderes en node. Desuden er spændingen så lav (maks. 3,3 V) og strømmen så svag (maks. 2 mA), at den ingen skade kan ske.

Løsning 2:

Ved denne løsning bruges 4 polet modular stik og fatninger. Disse stik bruges primært indenfor telefon området. Ligesom i løsning 1 kan der ikke ske skade ved at tilslutte telefoner til hverken noder eller kontrollere, pga. den lave spænding samt lille strøm der arbejdes med.

Løsning 3:

Stikkene her går under kategorien strømforsyningsstik, da de primært bruges ved de små sorte strømforsyninger, som vi alle har siddende rundt omkring i hjemmet. Hvis denne type stik skal anvendes, må det anbefales, at der bruges yderligere elektronik til at beskytte indgangene så disse ikke tager skade, hvis der skulle være nogen, der kunne finde på at tilslutte en strømforsyning til stikket.

Løsning 4:

2 polet DIN stik blev førhen meget brugt ved tilslutning af højtalere, men gør det ikke længere. Ligesom ved løsning 1 og 2 kan der ikke umiddelbart ske nogen skade pga. det tilsluttedes karakter.

Løsningsoversigt:

#	Stiktype	Stk. pris	Ledere
1	Jack	3,50 / 3,65	2
2	Modular	3,17 / 0,63	4
3	Strømforsyning	4,90 / 4,00	2
4	DIN	2,90 / 4,00	2

Table 7: Løsningsoversigt ved brug af ledning

3.2.6 SIL/DIL kontakter

Løsningsbeskrivelse:

På alle noder er der monteret én række af SIL/DIL[76] kontakter samt en tryk knap. Vi antager, at kontrolleren har en eller anden form for display. Kontrolleren bestemmer en midlertidig nøgle og viser denne i displayet, dvs. der vises hvordan kontakterne skal stilles. Når brugeren har indstillet kontakterne, trykkes på en knap på noden samt på kontrolleren. Dette medfører, at kontrolleren viser hvordan næste indstilling skal se ud. Sådan fortsættes indtil alle bits i nøglen er indstillet. Ved brug af 112 bits nøgle fordobles antallet af indstillinger der skal foretages, og dermed også risikoen for fejlindstilling.

Beskrivelse af hardware:

SIL kontakter 1:

Single-In-Line kontakter.
Kontakterne er af forgyldt beryllium-kobber.
Forventet holdbarhed: 1.000 operationer pr. kontakt.
Funktioner: 2/4
Fysiske mål: B: 3,2 mm H: 5,0 mm L: 7,6/12,6 mm
Varenr.: RS 665-095/665-102
Pris: 8,40/10 kr.

SIL kontakter 2:

Single-In-Line kontakter.
Kontakterne er af forgyldt nikkel.
Forventet holdbarhed: 500 operationer pr. kontakt.
Funktioner: 6/8
Fysiske mål: B: 3,2 mm H: 5,0 mm L: 17,7/22,8 mm
Varenr.: F 430-7239/430-7240
Pris: 8,97/9,38 kr.

[76] SIL = Single-In-Line, DIL = Dual-In-Line

DIL kontakter:

Dual-In-Line kontakter (top actuated)
Kontakterne er forgyldte.
Funktioner: 2/4/6/8/10/12
Fysiske mål: B: 6,9 mm H: 9,9 mm L: ??? mm
Varenr.: F 607-502/285-950/285-961/285-973/607-514/607-526
Pris: 3,04/3,59/4,28/4,69/5,52/6,49 kr.

SIL og DIL kontakter er en række af kontakter, som er meget små og derfor sidder tæt på hinanden. SIL og DIL kontakter er meget ens, men der er to forskelle. Den ene er måden de bliver monteret på printet. SIL kontakter er én lang række ben, hvor DIL består af to rækker, og derfor betegnes DIL at være mere stabile i forhold til SIL monteringsmæssigt set.

Den anden forskel er, at SIL kontakter har én fælles forbindelse/ben som alle kontakter er forbundet til, hvor DIL kontakter ikke deler en fælles forbindelse og kan derfor bruges uafhængigt af hinanden. Selve kontakten som skal vippes op/ned eller frem/tilbage er omkring 1,4 mm, så det må antages at det er nødvendigt at bruge en eller anden form for spids genstand til at skifte stilling. Kontakterne skifter mellem sluttet og ikke-sluttet, så der skal yderligere bruges en pull-up/down modstand, som vises på nedenstående figur. For at spare på GPIO kan der bruges en ekstra chip.

Opstilling af hardware:

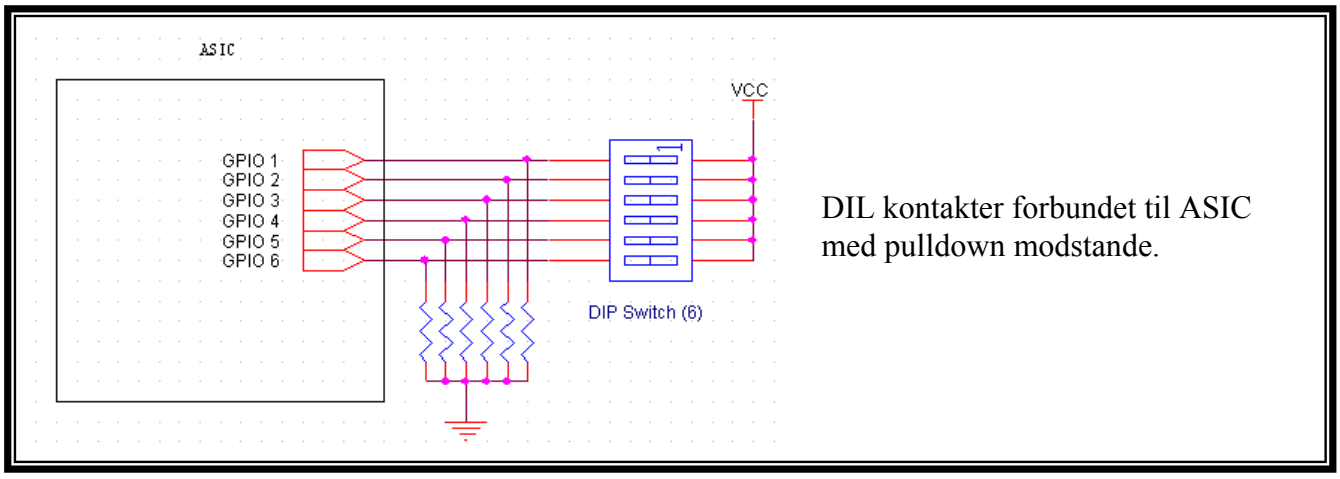


Figure 37: DIL kontakter tilsluttet ASIC

Generelt:
Kontakterne er enten sluttet eller afbrudt, dvs. at kontakten enten er i den ene tilstand eller i den anden. For en bit gælder det samme, enten er den 0 eller 1. Derfor kan man sige at kontakterne er binære, og vi har derfor at én kontakt repræsenterer én bit. Bruger vi to kontakter, repræsenterer disse to bits osv. Vi ser, at vi har en lineær sammenhæng mellem antallet af kontakter og antallet af bits der bliver repræsenteret.

Løsning 1:
Her bruges en 2 polet SIL-kontakt, dvs. mellem hver tryk på tryk-knappen, kan vi repræsentere 2 bits af nøglen. Kontakterne skal derfor indstilles 28 (56/2) gange før hele nøglen er overført og mapningen er på 2 (28-til-56).

Løsning 2, 3 og 4:
Disse løsninger er magen til løsning 1, med den forskel at der bliver brugt henholdsvis 4, 6 og 8 kontakter. Derved skal kontakterne indstilles henholdsvis 14 (56/4), 10 (56/6) og 7 (56/8) gange og mapningen bliver på: 4 (56/14), 5,6 (56/10) og 8 (56/7).

Løsning 5, 6, 7, 8, 9, 10:
Disse løsninger er magen til løsning 1-4 med den lille forskel, at der hér bliver brugt DIL kontakter i stedet for SIL. Den ene ende af DIL kontakterne forbindes til stel, så de fungerer som SIL kontakter funktionalitetsmæssigt. Udregningerne af hvor mange gang kontakterne skal sættes samt hvad mapningen bliver, er foretaget ovenfor og da de er meget simple, vil vi ikke her vise dem, blot henvis til ovenstående løsningsoversigt.

Løsningsoversigt:

#	Type	Stk. pris	Kontakter	Indstillinger	Mapning
1	SIL	8,40	2	28	2
2	SIL	10	4	14	4
3	SIL	8,97	6	10	5,6
4	SIL	9,38	8	7	8
5	DIL	4,55	2	28	2
6	DIL	3,59	4	14	4
7	DIL	4,28	6	10	6
8	DIL	6,76	8	7	8
9	DIL	5,52	10	6	9,3
10	DIL	6,49	12	5	11,2

Table 8: Løsningsoversigt ved brug af SIL/DIL kontakter

Indstillinger – Angiver hvor mange gange kontakterne skal sættes under den initiale nøgleudskiftning for at ”få” de 56 bits.

3.2.7 Flere trykknapper

Løsningsbeskrivelse:

Alle noder har foruden en kontrolknap monteret flere knapper. Vi antager, at kontrolleren har en eller anden form for display. Kontrolleren bestemmer en midlertidig nøgle og viser denne i displayet, dvs. der vises hvordan kontakterne skal stilles. Når noden skal inkluderes i netværket, vises i displayet, hvilke knapper der skal holdes nede. Brugeren holder de respektive knapper nede, som kontrolleren viser, og trykker på kontrolknappen. Derefter trykkes på en knap på kontrolleren, som viser hvilke knapper på noden, der nu skal holdes nede. Sådan fortsættes, indtil alle bits i nøglen er indstillet. Det maksimale antal ekstra knapper er 4, da man skal kunne gøre handlingen med kun én hånd, og den sidste finger skal bruges til kontrolknappen. Ved brug af 112 bits nøgle fordobles antallet af gange knapperne skal holdes nede, og dermed øges risikoen for fejl.

Beskrivelse af hardware:

Tryk knap:

PCB monteret tryknap

Fysiske mål: B: 6 mm H: 5 mm D: 6 mm

Forventet levetid: 100.000 tryk

Varenr.: F 535-916

Pris: 1,93 kr.

Opstilling af hardware:

Tre trykknapper tilsluttet til ASIC via pulldown modstande

Figure 38: Trykknap tilsluttet ASIC vha. pulldown modstand

Løsning 1:

Ved brug af én ekstra trykknap foruden kontrolknappen, kan vi repræsentere én bit ad gangen, og der skal derfor bestemmes 56 gange, om knappen skal være trykket ned eller ej. Dette giver en mapning på 1.

Løsning 2:

Bruges to ekstra trykknapper, kan 2 bits repræsenteres af gangen, og brugeren skal derfor i alt 28 (56/2) gange holde ingen, én eller begge knapper nede mens der trykkes på kontrolknappen. Mapningen er på 2 (56/28).

Løsning 3:

Med tre ekstra trykknapper skal brugeren bruge 19 (56/3) indtastninger og mapningen er på 2,95 (56/19).

Løsning 4:

Når fire ekstra trykknapper tages i brug, skal der foretages 14 (56/4) indtastninger og mapningen er op 4 (56/14).

Løsningsoversigt:

#	Antal trykknapper	Stk. pris	Tryk	Mapning
1	1	2,48	56	1
2	2	4,96 (2*2,48)	28	2
3	3	7,44 (3*2,48)	19	2,95
4	4	9,92 (4*2,48)	14	4

Table 9: Løsningsoversigt ved brug af trykknapper

3.2.8 Tastatur

Løsningsbeskrivelse:

Et tastatur er monteret på alle noder. Vi antager, at kontrolleren har en eller anden form for display. Kontrolleren bestemmer en midlertidig nøgle og viser denne i displayet, og brugeren taster kombinationen ind på tastaturet på noden. Brug af 112 bits nøgle fordobler antallet af tastetryk.

Beskrivelse af hardware:

Tastatur 1:

Selvklæbende membrantastatur som giver føleligt kvitteringsklik ved aktivering.
Taster er udformet som bobler, hvilket letter lokalisering.
Forventet holdbarhed: Ikke oplyst
Funktioner: 4 x 1 = 4
Fysiske mål: B: 38 mm H: 95 mm T: 0,15 mm
Varenr.: F 442-9746
Pris: 113,57 kr.

Tastatur 2:

Tastatur med sort ramme og hvide taster.
Forventet holdbarhed (min.): 10^6 operationer
Funktioner: 4 x 3 = 12
Fysiske mål: B: 51 mm H: 64 mm T: 10,6 mm
Varenr.: F 467-200
Pris: 45,95 kr.

Tastatur 3:

Tastatur med sort ramme og hvide taster.
Forventet holdbarhed (min.): 10^6 operationer
Funktioner: 4 x 4 = 16
Fysiske mål: B: 65 mm H: 64 mm T: 10,6 mm
Varenr.: F 467-212
Pris: 57,55 kr.

Opstilling af hardware:

Det har ikke været muligt, at finde skematiske tegninger der viser tilslutning af tastatur.

Figure 39: Tastatur tilsluttet til ASIC

Generelt:

For at aflæse hvilken tast der bliver trykket ned, sker dekodning på matrix form, dvs. tastaturet er inddelt i søjler og rækker ligesom den fysiske udformning. For tastatur nr. 1 bruges der 5 (4 + 1) forbindelser, for nr. 2 bruges der 12 (4 x 3) forbindelser og for nr. 3 skal der bruges 16 (4 x 4) forbindelser. Man finder ud af hvilken tast der er aktiveret ved, at undersøge hvilken søjle der er forbundet med hvilken række. Dette kan ASIC´en selv gøre eller man kan bruge en ekstra chip, der decideret kun står og laver dette. Da der på ASIC'en kun er adgang til 11 GPIO, skal der bruges ekstra chip for at dekode tastatur med 12 og 16 taster. Bruges en ekstra chip skal der kun bruges én GPIO til forskel for henholdsvis 5, 12 og 16. Brug af ekstra chip vil forøge omkostningerne på noden, hvilket ikke er ønskværdigt.

Løsning 1:

Til denne løsning bruges der et tastatur med i alt 4 taster. På tasterne kan der f.eks. stå 1, 2, 3, 4 Kontrolleren genererer en midlertidig nøgle på 56 bits og viser denne i displayet via nedenstående repræsentation:

Bits i nøgle	Tast på tastatur
00	A
01	B
10	C
11	D

Table 10: Sammenhæng mellem nøglebits og tastaturtryk

Dvs. der i alt skal foretages 28 (56 bits/2 bits pr. tast) tastetryk. Efter brugeren har indtastet alle bits, så kender noden til den midlertidige nøgle, og kan derfor modtage den "rigtige" nøgle. Antallet af tastetryk er 28 har vi en mapning på 2 (56/28).

Løsning 2:

Til denne løsning bruges der et tastatur med i alt 12 taster. På tasterne kan der f.eks. stå: A, B, C, D, E, F, G, H, I, J, K, L. Repræsentationen for bits'ne i nøglen kan se således ud:

Bits i nøgle	Tast på tastatur	Bits i nøgle	Tast på tastatur	Bits i nøgle	Tast på tastatur
000	A	100	E	-	I
001	B	101	F	-	J
010	C	110	G	-	K
011	D	111	H	-	L

Table 11: Sammenhæng mellem nøglebits og tastaturtryk

Da der kun er 12 taster på tastaturet, kan hver tast kun dække over 3 bits (8 muligheder) og derved er der 4 taster der ikke bruges. Der skal i alt foretages 19 (56 bits/3 bits pr. tast) tastetryk. Mapningen er på 2,95 (56/19).

Løsning 3:

Denne løsning bruger et tastatur med i alt 16 taster, hvor der f.eks. kunne stå: 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G. Repræsentationen for bits'ne i nøglen kan se således ud:

Bits i nøgle	Tast på tastatur	Bits i nøgle	Tast på tastatur	Bits i nøgle	Tast på tastatur	Bits i nøgle	Tast på tastatur
0000	1	0100	5	1000	9	1100	D
0001	2	0101	6	1001	A	1101	E
0010	3	0110	7	1010	B	1110	F
0011	4	0111	8	1011	C	1111	G

Table 12: Sammenhæng mellem nøglebits og tastaturtryk

Som det ses så bruges alle 16 taster og hver tast repræsenterer 4 bits, dvs. at der skal tastes i alt 14 (56 bits/ 4 bits pr. tast) gange for at få nøglen overført til noden. Mapningen her er på 4 (56/14).

Løsningsoversigt:

#	Taster	Stk. pris	Tastetryk	Mapning
1	4 (4 x 1)	118	28	2
2	12 (4 x 3)	45,95	19	2,95
3	16 (4 x 4)	57,55	14	4

Table 13: Løsningsoversigt ved brug af tastatur

Tastetryk – Angiver hvor mange tastetryk der skal udføres for at ”overføre” hele nøglen. (56 bits)

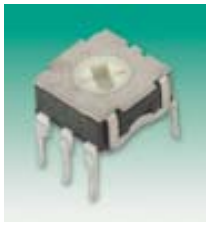
3.2.9 Drejeomskifter

Løsningsbeskrivelse:

Alle noder har én drejeomskifter samt én trykknap. Vi antager, at kontrolleren har en eller anden form for display. Kontrolleren bestemmer en midlertidig nøgle og viser denne i displayet, dvs. der vises hvordan drejeomskifteren skal stilles. Når noden skal inkluderes i netværket, vises i displayet, hvilken position drejeomskifteren skal stilles i. Brugeren sætter omskifteren i den viste position, og trykker på kontrolknappen. Derefter trykkes på en knap på kontrolleren, som viser den nye position, der ikke behøver at være en anden. Sådan fortsættes, indtil alle bits i nøglen er indstillet. Ved brug af 112 bits nøgle fordobles antallet af gange omskifteren skal sættes, og dermed øges risikoen for fejl.

Beskrivelse af hardware:

Omskifter 1:



DIL drejeomskifter til printmontage.

Positioner: 2/3/10/16

Drejningsvinkel: 120°/60°/36°/22,5°

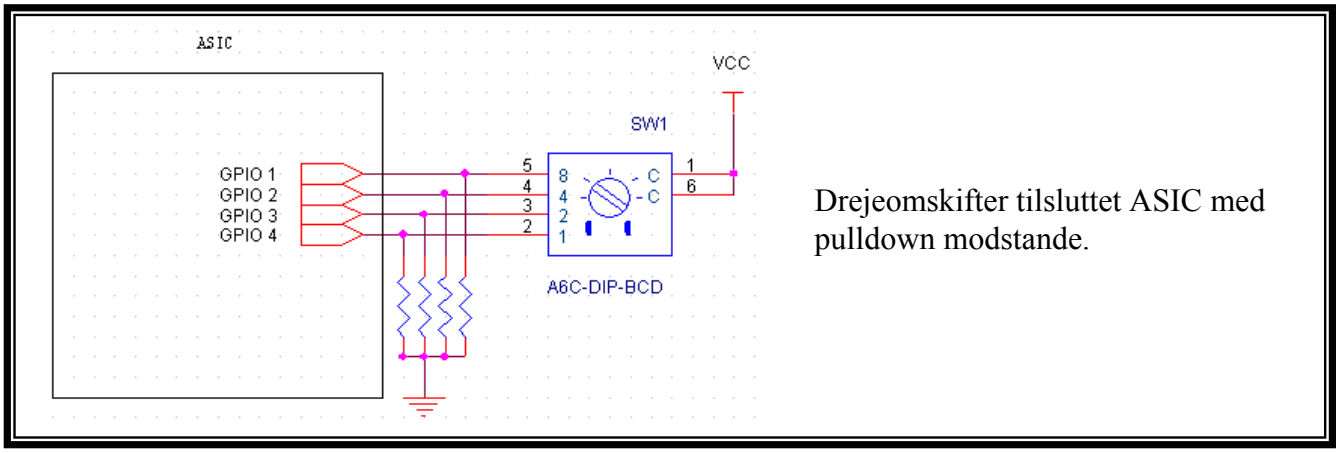
Forventet levetid: 1.500/1.500/20.000/20.000 skift

Fysiske mål: L: 8,0 mm B: 7,4 mm H: 4,25 mm

Varenr.: F 415-6572/415-6626/415-6638/415-6640

Pris: 8,97/8,97/14,63/14,90 kr.

Opstilling af hardware:



Drejeomskifter tilsluttet ASIC med pulldown modstande.

Figure 40: Drejeomskifter tilsluttet ASIC

Løsning 1:

Til denne løsning bruges en omskifter med kun to positioner. Derved kan der kun overføres én bit af nøglen ad gangen. Mellem hver position drejeomskifteren kan sættes i, er der 120 grader. Der skal drejes 56 gange på omskifteren for, at få "overført" de 56 bits, så mapningen er på 1.

Løsning 2:

Løsningen hér bruger en omskifter med tre positioner, hvor der er 60 grader mellem hver position. Vha. tabellen findes der frem til, at der skal foretages 36 indstillinger af omskifteren og mapningen er på 1,56 (56/36).

Løsning 3:

Udskiftes omskifteren med en der indeholder 10 positioner, skal omskifteren stilles 17 gane, hvilket resulterer i en mapning på 3,29 (56/17). Der er 36 grader mellem hver position.

Løsning 4:

Den sidste mulighed er, at bruge en omskifter med 16 positioner med 22,5 grader mellem hver position. Derved kan vi repræsentere 4 bits ad gangen (16 muligheder) og der skal derfor maksimalt foretages 14 drejninger. Dette giver en mapning på 4 (56/14).

Løsningsoversigt:

#	*Positioner*	*Stk. pris*	*Drejningsvinkel*	*Antal drejninger*	*Mapning*
1	*2*	*8,97*	*120*	*56*	*1*
2	*3*	8,97	*60*	*36*	*1,56*
3	*10*	*14,63*	*36*	*17*	*3,29*
4	*16*	14,90	*22,5*	*14*	*4*

Table 14: Løsningsoversigt ved brug af drejeomskiftere

Drejningsvinklen – Angiver vinklen mellem hver position på omskifteren.

Drejninger – Angiver det absolut maksimale antal drejninger der skal til for, at få lavet en kombination der dækker alle bits i nøglen.

3.2.10 Trimmer

Løsningsbeskrivelse:

På alle noder er monteret én trimmer samt én trykknap. "Opsætningen" og fremgangsmåden for inkluderingen foregår som ved drejeomskifteren. Forskellen er, at omskifteren drejes i ryk, da der her er fast positioner, mens drejningen på trimmeren sker trinløst/flydende.

Beskrivelse af hardware:

Trimmer 1:

Single-turn, lukket udførelse, kulbane, miniature, top-/sidejustering
100R – 10M Ω (±20 % modstandstolerance)
Drejningsvinkel: 235 °
Fysiske mål vertikal: B: 10,3 mm H: 12,1 mm D: 4,5 mm
Fysiske mål horisontal: B: 10 mm H: 4,5 mm D: 10,3 mm
Varenr. Vertikal: F 614-683
Varenr. Horisontal: F 614-555
Pris: 1,70 kr.

Trimmer 2:

Single-turn, lukket udførelse, kulbane, miniature, top-/sidejustering
100R – 1M Ω (±10 % modstandstolerance)
Drejningsvinkel: 310 °
Fysiske mål vertikal: B: 9,53 mm H: 4,8 mm D: 9,53 mm
Fysiske mål horisontal: B: 9,53 mm H: 9,5 mm D: 4,83 mm
Varenr. Vertikal: 346-410
Varenr. Horisontal: 346-548
Pris: 4,80 kr.
Denne trimmer findes i en udgave med fingerskrue (pris: 12,25 kr.)

Multi-turn trimmer, dvs. trimmere som kan drejes flere gange rundt, ser vi bort fra da de ikke er brugervenlige. De er ikke brugervenlige idet det tager relativ lang tid at skrue fra den ene til den anden ende, samt at det kan være svært for brugeren at huske om trimmeren nu er drejet 6 eller 7 gange rundt.

Opstilling af hardware:

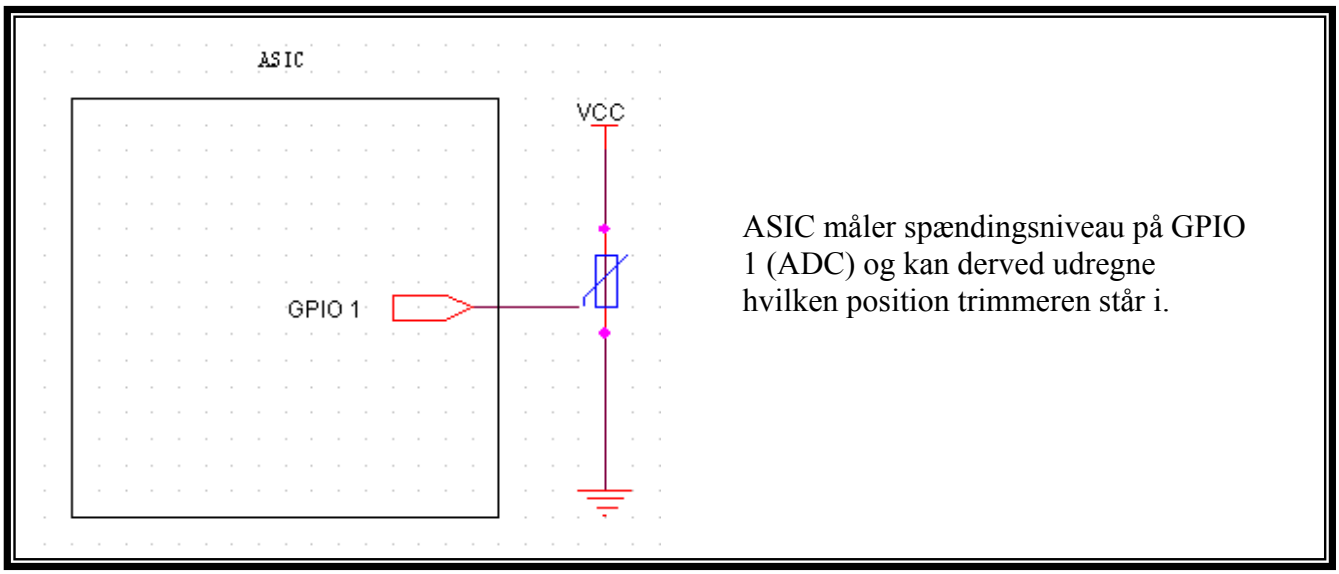


Figure 41: Trimmer tilsluttet ASIC

Løsning 1:

Løsningen her baserer sig på brug af trimmer 1, som ”inddeles” i 2 positioner. Dvs. den ene position er ved helt åben (helt til venstre), mens den anden er ved helt lukket (helt til højre).
Da det er ydrepositionerne der bruges, er der 235 grader mellem positionerne med en tolerance på 117 grader. Sådan som der rent faktisk bør måles på trimmeren, er om den er over eller under halvdelen. Denne løsning med kun to positioner kræver mindst muligt af brugeren hvad angår præcisionen, men til gengæld kræver den flest mulig antal indstillinger. Der skal maksimalt drejes 56 gange på trimmeren for at få ”overført” de 56 bits, så mapningen er på 1 (1-til-1).

Løsning 2:

Samme løsning som løsning 1 men med trimmer 2 i stedet for. Pga. den større drejningsvinkel, er vinklen mellem positionerne på 310 grader med en tolerance på 155 (310/2) grader.

Løsning 3:

I løsning 2 bruges trimmer 1, men den inddeles nu i 3 positioner, som er: åben, halvvejs og lukket.
Dvs. der er 117 (235/2) grader mellem hver position med en tolerance på 39 (235/6) grader.

Løsning 4:

Samme løsning som løsning 3 men med trimmer 2 i stedet for. Pga. den større drejningsvinkel, er vinklen mellem positionerne på 165 (310/2) grader med en tolerance på 52 (310/6) grader.

Løsning 5, 7, 9 og 11:

Trimmer 1 inddeles i 4, 5, 6, og 7 positioner. Dvs. der er henholdsvis 78 (235/3), 58 (235/4), 47 (235/5) og 39 (235/6) grader mellem hver position med en tolerance på 29 (235/8), 24 (235/10), 20 (235/12) og 17 (235/14) grader.

Løsning 6, 8, 10 og 12:

Bruges trimmer 2 som inddeles i 4, 5, 6, og 7 positioner. Dvs. der er henholdsvis 78 (235/3), 58 (235/4), 47 (235/5) og 39 (235/6) grader mellem hver position med en tolerance på 29 (235/8), 24 (235/10), 20 (235/12) og 17 (235/14) grader.

Løsning 13:

Trimmer 1 inddeles i 8 positioner, så der er 34 (235/7) grader mellem hver position med en tolerance på 15 (235/16) grader. Med de 8 positioner kan vi "overføre" 3 bits af nøglen ad gangen, så der skal i alt foretages 19 (56/3) drejninger. Det giver en mapningen på 3 (19-til-56).

Løsning 14:

Samme løsning som løsning 13 men med trimmer 2 i stedet for. Pga. den større drejningsvinkel, er vinklen mellem positionerne på 44 (310/7) grader med en tolerance på 19 (310/16) grader.

Løsningsoversigt:

#	Positioner	Stk. pris	Drejningsvinkel	Drejningstolerance	Drejninger	Mapning
1	2	1,70	235	117	56	1
2	2	4,80	310	165	56	1
3	3	1,70	117	39	38	1,56
4	3	4,80	165	52	38	1,56
5	4	1,70	78	29	28	2
6	4	4,80	103	39	28	2
7	5	1,70	58	24	25	2,24
8	5	4,80	78	31	25	2,24
9	6	1,70	47	20	22	2,55
10	6	4,80	62	26	22	2,55
11	7	1,70	39	17	20	2,8
12	7	4,80	52	22	20	2,8
13	8	1,70	34	15	19	2,95
14	8	4,80	44	19	19	2,95

Table 15: Løsningsoversigt ved brug af trimmer

3.2.11 Farvet lysdioder

Løsningsbeskrivelse:

For nedenstående løsningsforslag gælder der, at alle noder har én lysdiode samt en trykknap. På kontrolleren findes der farvede knapper i samme farve, som lysdioden på noden kan vise. Når en node skal inkluderes i netværket, er det noden der genererer en midlertidig nøgle. Nøglen bliver overført til kontrolleren ved, at lysdioden viser en farve/sekvens af blink. Brugeren trykker på den knap med samme farve på kontrolleren og derefter på kontrolknappen på noden. De fleste af løsningerne bruger blinkesekvenser, hvor brugeren på kontrolleren skal gentage denne sekvens, noden viser vha., de farvede knapper. Efter et stykke tid, hvor brugeren ikke har trykket på knappen på noden, gentages sekvensen. Når brugeren har trykket på kontrolknappen på noden, vises den næste farve/sekvens.

Beskrivelse af hardware:

Lysdiode 1:

3/5 mm lysdiode.
2 forbindelser (ben): rød.
Udstrålingsvinkel: 30°
Farver: 1
Fysiske mål: D: 3/5 mm H: 4,6/8,6 mm
Varenr.:F 656-471/656-689
Pris: 0,40/0,39 kr.

Lysdiode 2:

3/5 mm to farvet lysdiode.
2 forbindelser (ben): rød, grøn
Udstrålingsvinkel: 60°
Farver: 2 (rød, grøn)
Fysiske mål: D: 3/5 mm H: 4,6/8,6 mm
Varenr.: F 637-180/637-416
Pris: 0,966/0,97 kr.

Lysdiode 3:

3/5 mm tre farvet lysdiode.

3 forbindelser (ben): rød, grøn, rød + grøn = orange

Udstrålingsvinkel: 60°/24°

Farver: 3 (rød, grøn og orange)

Fysiske mål: D: 3/5 mm H: 4,6/8,6 mm

Varenr.: F 637-210/637-282

Pris: 0,828/0,966 kr.

Lysdiode 4:

5 mm Full color RGB lysdiode med diffus udstråling.

6 forbindelser (ben): består af 2 blå, 1 grøn og 1 rød lysdiode.

Udstrålingsvinkel: 60°

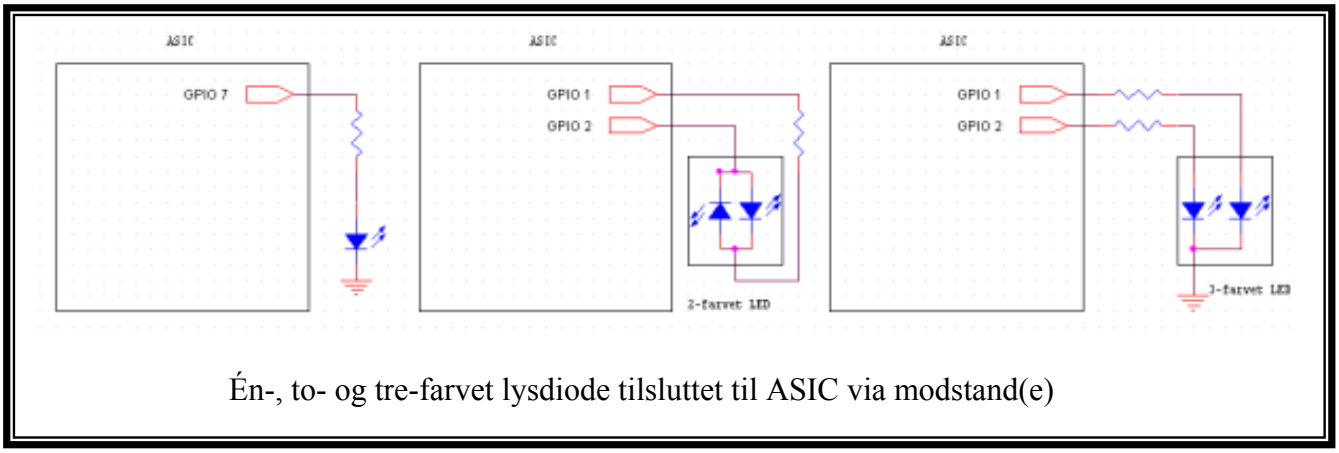
Farver: Full color (alle farver)

Fysiske mål: D: 5 mm H: 8,6 mm

Varenr.: F 621-419

Pris: 20,56 kr.

Opstilling af hardware:



Én-, to- og tre-farvet lysdiode tilsluttet til ASIC via modstand(e)

Figure 42: Farvet lysdiode tilsluttet ASIC

Løsning 1:

Der bruges en énfarvet lysdiode. Dioden blinker én eller to gange. Ét blink repræsenterer bit-værdien 0 og to blink repræsenterer bit-værdien 1. Antallet af aflæsninger er 56 og mapningen er derfor på 1 (56/56).

Løsning 2:

To-farvet lysdiode (f.eks. rød(R)/grøn(G)). Den ene farve repræsenterer bit-værdien 0 mens den anden repræsenterer bit-værdien 1. Da det er en 1-til-1 (1) mapning der foregår, skal der altid foretages 56 tastetryk. Da en bit kun kan repræsentere to værdier og da der her kun bruges to farver, vil vi aldrig kunne opnå en bedre mapning end 1 (56/56), hvorfor der ikke er flere løsningsforeslag med to-farvede lysdioder.

Løsning 3:

Tre-farvet lysdiode (f.eks. rød(R)/gul(Y)/grøn(G)). Den ene farve repræsenterer bitkombinations-værdien 01, den anden farve repræsenterer bitkombinations-værdien 10, den tredie farve repræsenterer bitkombinations-værdien 00/11. For at fortælle om der er tale om 00 eller 11, er det nødvendigt at tilføje ekstra bits. Skema over farvesekvens og bitkombination:

Farve	*Bitkombination*
R	*01*
G	*10*
Y	*00/11*

Table 16: Sammenhæng mellem nøglebits og farve

For at fortælle om Y repræsenterer bitkombinationen 00 eller 11, tilføjer vi ekstra bits til at fortælle dette. Vi har i alt 28 (56/2) bitkombinationer og i værste tilfælde er de alle enten 00 eller 11, derfor skal vi bruge ekstra bits for, at fortælle hvilke der er hvad. Vi skal snart finde ud, at vi ikke vinder noget ved denne mapning. Ser vi på to par ad gangen, findes der følgende sammensætninger: 0000, 0011, 1100 og 1111. Dvs. der er fire parkombinationer, men vi kan kun ”vise” tre med vores tre-farvet lysdiode. Derfor er vi nød til at bruge én farve der gælder for 00 og én der gælder for 11.

Havde vi haft mulighed for at bruge flere farver, kunne vi have nøjes med én farve for hver dobbelt parkombination, i stedet for den ene vi er tvunget til nu. Dette resulterer i at vi skal bruge 14 ekstra bits for at fortælle, hvordan de 14 parbits ser ud og vi kommer op på totalt 42 (28+14) bits i alt. Dette medfører en mapning på 1,33 (56/42). Man kunne vælge at udelukke enten 00 eller 11, men det frarådes da det giver begrænsinger på antallet af nøgler.

Løsning 4:

Tre-farvet lysdiode (f.eks. rød(R)/gul(Y)/grøn(G)). Skema over farvesekvens og bitkombination:

Farvesekvens	*Bitkombination*
RR	*000*
RG	*001*
RY	*010*
GR	*011*
GG	*100*
GY	*101*
YR	*110*
YG	*111*
YY	*-*

Table 17: Sammenhæng mellem nøglebits og farvesekvens

Hver tofarve kombination repræsenterer tre bits, derfor skal der foretages 38 (2*(56/3)) tastetryk og mapningen bliver derfor på 1,47 (56/38). Det ses, at der er én farvekombination der ikke bruges.

Løsning 5:

Tre-farvet lysdiode (f.eks. rød(R)/gul(Y)/grøn(G)). Skema over farvesekvens og bitkombination:

Farvesekvens	*Bitkombination*	*Farvesekvens*	*Bitkombination*
RRR	*0000*	*GGY*	*1110*
RRG	*0001*	*GYR*	*1111*
RRY	*0010*	*GYG*	*-*
RGR	*0011*	*GYY*	*-*
RGG	*0100*	*YRR*	*-*
RGY	*0101*	*YRG*	*-*
RYR	*0110*	*YRY*	*-*
RYG	*0111*	*YGR*	*-*
RYY	*1000*	*YGG*	*-*

GRR	*1001*	*YGY*	*-*
GRG	*1010*	*YYR*	*-*
GRY	*1011*	*YYG*	*-*
GGR	*1100*	*YYY*	*-*
GGG	*1101*	*-*	*-*

Table 18: Sammenhæng mellem nøglebits og farvesekvens

Hver trefarve kombination repræsenterer fire bits, så der skal foretages 42 (3*(56/4)) tastetryk. Det ses, at der er 11 farvekombinationer der ikke bruges og at mapningen giver 1,33 (56/42).

Løsning 6:

Tre-farvet lysdiode (f.eks. rød(R)/gul(Y)/grøn(G)). Skema over farvesekvens og bitkombination:

Farvesekvens	*Bitkombination*	*Farvesekvens*	*Bitkombination*	*Farvesekvens*	*Bitkombination*
RRRR	*000000*	*GRRR*	*011011*	*YRRR*	*110110*
RRRG	*000001*	*GRRG*	*011100*	*YRRG*	*110111*
RRRY	*000010*	*GRRY*	*011101*	*YRRY*	*111000*
RRGR	*000011*	*GRGR*	*011110*	*YRGR*	*111001*
RRGG	*000100*	*GRGG*	*011111*	*YRGG*	*111010*
RRGY	*000101*	*GRGY*	*100000*	*YRGY*	*111011*
RRYR	*000110*	*GRYR*	*100001*	*YRYR*	*111100*
RRYG	*000111*	*GRYG*	*100010*	*YRYG*	*111101*
RRYY	*001000*	*GRYY*	*100011*	*YRYY*	*111110*
RGRR	*001001*	*GGRR*	*100100*	*YGRR*	*111111*
RGRG	*001010*	*GGRG*	*100101*	*YGRG*	*-*
RGRY	*001011*	*GGRY*	*100110*	*YGRY*	*-*
RGGR	*001100*	*GGGR*	*100111*	*YGGR*	*-*
RGGG	*001101*	*GGGG*	*101000*	*YGGG*	*-*
RGGY	*001110*	*GGGY*	*101001*	*YGGY*	*-*
RGYR	*001111*	*GGYR*	*101010*	*YGYR*	*-*
RGYG	*010000*	*GGYG*	*101011*	*YGYG*	*-*

RGYY	*010001*	*GGYY*	*101100*	*YGYY*	-
RYRR	*010010*	*GYRR*	*101101*	*YYRR*	-
RYRG	*010011*	*GYRG*	*101110*	*YYRG*	-
RYRY	*010100*	*GYRY*	*101111*	*YYRY*	-
RYGR	*010101*	*GYGR*	*110000*	*YYGR*	-
RYGG	*010110*	*GYGG*	*110001*	*YYGG*	-
RYGY	*010111*	*GYGY*	*110010*	*YYGY*	-
RYYR	*011000*	*GYYR*	*110011*	*YYYR*	-
RYYG	*011001*	*GYYG*	*110100*	*YYYG*	-
RYYY	*011010*	*GYYY*	*110101*	*YYYY*	-

Table 19: Sammenhæng mellem nøglebits og farvesekvens

Hver firefarve kombination repræsenterer seks bits. Der skal foretages 40 ($4*\lceil 56/6 \rceil$) tastetryk, hvilket giver en mapning på 1,4 (56/40). Det ses, at der er 17 farvekombinationer, der ikke bruges.

Løsning 7:

Fire-farvet lysdiode (f.eks. rød(R)/grøn(G)/blå(B)/gul(Y)). Hver farve repræsenterer to bitværdier. Skema over farve-/bitkombination:

Farve	*Bitkombination*
R	*00*
G	*01*
B	*10*
Y	*11*

Table 20: Sammenhæng mellem nøglebits og farve

Mapningen er på 2 (56/28), da der skal bruges 28 (56/2) tastetryk. Da antallet af farver er lig med et helt multiplum af bitkombinationen, kan vi ikke effektiviserer yderligere ved at mappe flere farver i mod flere birkombinationer. Vi kan kun få forskellig mapning ved ”skæve” farvekombinationer, som ved tre-farvede lysdioder.

Løsning 8:

Fire-farvet lysdiode (f.eks. rød(R)/grøn(G)/blå(B)/gul(Y)). Hver tofarve kombination repræsenterer fem bitværdier. Skema over farvesekvens og bitkombination:

Farvesekvens	*Bitkombination*
RR	*0000*
RG	*0001*
RB	*0010*
RY	*0011*
GR	*0100*
GG	*0101*
GB	*0110*
GY	*0111*
BR	*1000*
BG	*1001*
BB	*1010*
BY	*1011*
YR	*1100*
YG	*1101*
YB	*1110*
YY	*1111*

Table 21: Sammenhæng mellem nøglebits og farvesekvens

Mapningen er igen på 2, da der ligesom i løsning 7 skal bruges 28 (2*56/4) tastetryk.

Da vi har en 4-værdi farve repræsentation over for en 2-værdi bit repræsentation, får vi ikke noget effektivt ud af at oversætte/mappe flere kombinationer sammen. Grunden til dette er, at 4 er et helt antal multipla af 2. Vi kan kun få forskellig mapning ved ”skæve” farvekombinationer, som ved tre-farvede lysdioder.

Løsning 9:

Seks-farvet lysdiode (f.eks. rød(R)/grøn(G)/blå(B)/gul(Y)/hvid(W)/violet(V)). Hver tofarve kombination repræsenterer fem bitværdier. Skema over farvesekvens og bitkombination:

Farvesekvens	*Bitkombination*	*Farvesekvens*	*Bitkombination*
RR	*00000*	*YR*	*10010*
RG	*00001*	*YG*	*10011*
RB	*00010*	*YB*	*10100*
RY	*00011*	*YY*	*10101*
RW	*00100*	*YW*	*10110*
RV	*00101*	*YV*	*10111*
GR	*00110*	*WR*	*11000*
GG	*00111*	*WG*	*11001*
GB	*01000*	*WB*	*11010*
GY	*01001*	*WY*	*11011*
GW	*01010*	*WW*	*11100*
GV	*01011*	*WV*	*11101*
BR	*01100*	*VR*	*11110*
BG	*01101*	*VG*	*11111*
BB	*01110*	*VB*	-
BY	*01111*	*VY*	-
BW	*10000*	*VW*	-
BV	*10001*	*VV*	-

Table 22: Sammenhæng mellem nøglebits og farvesekvens

Mapningen er på to-til-fem (0,4) og der skal i alt bruges 24 ($2*\lceil 56/5 \rceil$) tastetryk. Det ses, at der er 4 farvekombinationer, der ikke bruges.

Løsningsoversigt:

#	*Antal farver*	*Stk. pris (kr.)*	*Antal tastetryk*	*Mapning*
1	*1*	*0,40 / 0,39*	*56*	*1*
2	*2*	*0,966 / 0,97*	*56*	*1*
3	*3*	*0,828 / 0,966*	*42*	*1,33*
4	*3*	*0,828 / 0,966*	*38*	*1,47*
5	*3*	*0,828 / 0,966*	*42*	*1,33*
6	*3*	*0,828 / 0,966*	*40*	*1,4*
7	*4***	*- / 20,56*	*28*	*2*
8	*5***	*- / 20,56*	*28*	*2*
9	*6***	*- / 20,56*	*24*	*2,5*

Table 23: Løsningsoversigt ved brug af farvet lysdioder

*** = Lysdioden er en full color RGB lysdiode, men bliver "begrænset" i denne løsningen.*

3.2.12 7-segment

Løsningsbeskrivelse:

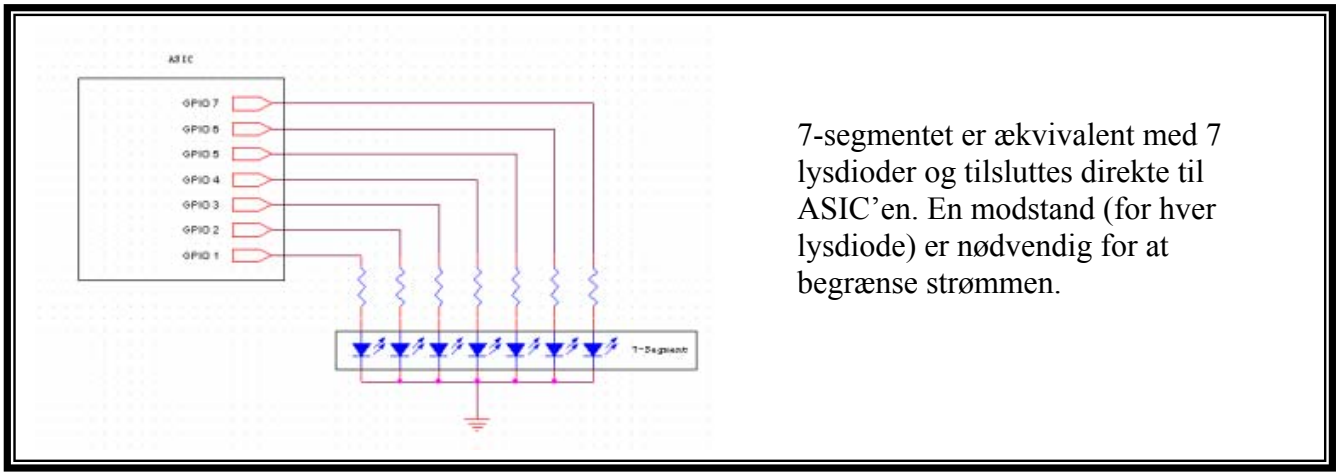
Noderne har et eller flere 7-segment(er) samt en trykknap, og kontrollerne har taster med de tal/symboler, som 7-segmentet kan vise. Noden bestemmer en midlertidig nøgle, som indtastes på kontrolleren. Brugeren trykker på den knap på kontrolleren, som 7-segmentet(erne) viser. Derefter trykkes på kontrolknappen på noden, som viser næste tal/symbol, som skal indtastes på kontrolleren.

Beskrivelse af hardware:

7-segment:

Grønt lysende 7-segment med fælles anode/katode.
Display størrelse: 13,2 mm.
Format: 8.
Fysiske mål: B: 12,40 mm H: 17,55 mm D: 7,00 mm
Varenr.: F 622-175/622-187
Pris: 3,31/3,31 kr.

Opstilling af hardware:



7-segmentet er ækvivalent med 7 lysdioder og tilsluttes direkte til ASIC'en. En modstand (for hver lysdiode) er nødvendig for at begrænse strømmen.

Figure 43: 7-segment tilsluttet ASIC

Løsning 1:

7-Segmentet på noden viser det tal, som skal trykkes på kontrolleren. Når tallet er indtastet, trykker brugeren på knappen på noden, som så viser det næste tal. I denne løsning bruges kun tallene 0, 1, 2, 3, 4, 5, 6, 7, 8 og 9, i alt 10 tal. Når der ikke skal indtastes flere tal, viser noden "-" i 7-segmentet, som tegn på at hele overførelsen er sket. Vha. mappingstabellen findes der frem til, at der skal foretages 17 indtastninger, hvilket giver en mapning på 3,29 (56/17).

Løsning 2:

Udvider vi repræsentationen af nøglen til at bruge hexidecimal tal, bruger vi 16 forskellige symboler/tal i 7-segmentet. Symbolerne/tallene er: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e og f. Der er i alt 16 tal/symboler. Hvert tal repræsenterer 4 bitværdier. Antallet af visninger af tal/symboler i 7-segmentet er på 14 (56/4) og vi har en mapning på 4 (56/14).

Løsning 3:

Ved brug af to stk. 7-segmenter og decimal tal repræsentationen, kan vi vise værdier fra 00 til 99 ligesom i løsning 4. Man skal bruge 9 ($\lceil 17/2 \rceil$) indtastninger for at få overført hele nøglen, og det giver en mapning på 6,22 (56/9). De 17 kommer fra antallet af indtastninger i løsning 1, da vi nu har dobbelt antal 7-segmenter.

Løsning 6:

Repræsenteres nøglen vha. hexidecimal tal, kan vi med 2 stk. 7-segmenter vise 256 (16^2) forskellige kombinationer fra 00 til ff. Dvs. at hver kombination repræsenterer 8 bitværdier. Antallet af visninger af tal/symboler i 7-segmenterne er på 7 ($\lceil 56/8 \rceil$) og vi har en mapning på 8 (56/7).

Løsningsoversigt:

#	Antal af tal/symboler	Stk. pris	Visninger	Mapning
1	10	8,83	17	3,29
2	16	8,83	14	4
3	20 (2 x 10)	17,66	9	6,22
4	32 (2 x 16)	17,66	7	8

Table 24: Løsningsoversigt ved brug af 7-segmenter

3.2.13 Smart Card

Løsningsbeskrivelse:

Alle kontrollere og noder har en kortenhed. Når en node skal inkluderes i netværket, sættes først et smart card i kortenheden på kontrolleren. Denne genererer en midlertidig nøgle og lægger denne ned på kortet. Kortet tages ud af kontrollen og sættes ind i noden, som aflæser den initiale nøgle og den endelige nøgleudveksling kan nu foregå.

Beskrivelse af hardware:

Smart Card:

256 Bytes E^2prom, 2 lednings I^2C bus, ISO7816 for placering af kontaktflader
Forventet levetid (min.): 1 mio. slette/skrive cykler
Valid data uden strøm: 10 år
Klok frekvens: 100 kHz
Fysiske mål: B: 54 mm H: 85 mm D: 0,8 mm
Varenr.: F 301-8430
Pris: 16,56 kr.

Smart Card enhed:

Læser og skriver Smart Card, 16 kontaktflader, ISO7816 standard
Ved indsættelse af kort afbrydes en kontakt (kortdetektering)
Forventet levetid (min.): 30.000 cykler
Fysiske mål: B: 59 mm H: 7 mm L: 42,1 mm
Varenr.: F 430-2643
Pris: 12,28 kr.

Opstilling af hardware:

Figure 44: Smart Card enhed tilsluttet ASIC

Løsning 1:

Til hver kontroller medfølger et kort, som skal bruges ved inkluderingen af nye noder. Kortet er ikke knyttet specielt til den pågældende kontroller, men kan bruges på kryds og tværs af hinanden. Når en node skal inkluderes i netværket, sættes et kort i kortenheden i kontrolleren. Dette opdager kontrolleren, da kortdetekteringsforbindelsen i kortenheden bliver afbrudt (sker ved indsættelse af kort). Kontrolleren genererer en midlertidig nøgle og lægger denne ned på kortet. Kortet tages ud af kontrolleren og sættes i noden. Noden opdager dette, aflæser og sletter derefter nøglen fra kortet. Den endelige overførelse af netværksnøglen kan nu begynde. Da der er kortdetektering i selve kortenheden, er det ikke nødvendigt med en trykknap på noden som i andre løsningsforslag.

Løsning 2:

Samme som i løsning 1 med den forskel, at den midlertidige nøgle er 112 bits i stedet for 56 bits. Skrivning og læsning fra kortet sker så hurtigt, at det ikke vil tage mærkbart længere tid ved brug af nøgle på 112 bits, som der skal bruges til 3DES.

Generelt:

Ser vi lidt nærmere på selve inkluderingsproceduren, så starter vi med, at kontrolleren genererer en nøgle og lægger denne ned på kortet. Kontrolleren bør vise i displayet, at den er klar til at inkludere en ny node. Da vi går ud fra at man befinder sig i nærheden af den node man vil inkludere, bør der være en tidsbegrænsning på, hvor lang tid nøglen kan leve.

Selve skrivning og læsning fra og til kortet foregår ved 100 kHz, og da der kun skal overføres 56/112 bits, kan dette klares på under ét sekund, når alt overhead[77] er taget med. Hvis levetiden sættes til 10 – 15 sekunder, må man betegne det som værende umuligt for tredje part at kunne franarre kortet fra personen, der er igang med inkluderingen, aflæse nøglen og så "give" kortet tilbage igen. Alt sammen uden at personen opdager dette.

For at bruge ovenstående metode med begrænset levetid, er det ikke nødvendigt at både kontroller og node kender til/har den samme tid, det er nok at kontrolleren holder styr på dette. Tager vi udgangspunkt i den nuværende inkluderingsprocess, så skal den nye foregå således:

1. Kort indsættes i kontrolleren
2. Kontrolleren genererer midlertidig nøgle og lægger denne ned på kortet
3. Derefter går kontrolleren i "learning mode" og starter en tæller
4. Kortet tages ud af kontrolleren og indsættes i noden
5. Noden aflæser én byte fra kortet og sletter denne med det samme
6. Noden aflæser næste byte fra kortet og sletter denne. Dette gøres til hele nøglen er aflæst
7. Noden sender en broadcast pakke afsted og afventer svar fra kontrolleren
8. Kontrolleren modtager pakken, stopper tælleren og kontrollerer at udvekslingen af nøglen er foregået inden for det definerede tidsinterval
9. Noden og kontrolleren følger den sædvanlige inkluderingsprocedure, hvor noden får tildelt Home og Node ID, og bagefter kan den endelige netværksnøgle overføres til noden.

Løsningsoversigt:

#	*Stk. pris*	*Nøglelængde*
1	*12,28 (+ 16,56)*	*56*
2	12,28 (+ 16,56)	112

Table 25: Løsningsoversigt ved brug af Smart Card

[77] Overhead er de yderligere informationer, som overføres til kortet foruden de egentlige informationer/bits. Af yderligere informationer kan f.eks. nævnes adresse (hvor nøglen ligger) samt diverse handshakes, der bruges af I^2C protokollen.

3.2.14 Fingeraftryk

En mulighed for at overføre den initiale nøgle er, at bruge en fingeraftrykslæser. Et dansk firma (Quard Technology)[78] har udviklet en fingeraftrykslæser, som har samme dimensioner som et normalt kreditkort. Kortet indeholder en processor, et batteri, en sensor, et display og betegnes som et biometrisk smartcard.

Kortet fungerer ved, at brugeren først skal oplærer kortet, så det kun reagerer på netop den bruger. Det gøres ved, at brugeren fører en finger hen over sensoren tre gange. Der er ingen krav til hvilken finger der skal bruges blot, at den føres over sensoren i samme retning hver gang. Sensoren har en brede på 1 mm og en længde på 1 cm. Hver gang brugeren fører sin finger hen over sensoren, vises et engangskodeord i displayet.

Figure 45: Fingeraftrykslæser

Selve kortet har ikke den store interesse for Z-wave security løsningen, men det har selve teknologien. Man kan forestille sig, at man fra kortet bruger sensoren og evt. processoren. Programmellet skal laves om, så det ikke bindes til én specifik person, men genererer en kode ud fra fingeraftrykket. Ved at placere en sensor (og tilhørende processor) på både slave og kontroller, skal brugeren blot føre den samme finger hen over sensoren i samme retning, og der vil blive dannet samme nøgle begge steder. Kontrolleren kan evt. lave en MD 5 på nøglen, sende den til slaven, som svarer tilbage om nøglerne passer sammen. Passer de kan den endelige overførelse af netværksnøglen ske, ellers må brugeren igen føre en finger hen over sensoren.
Efter flere henvendelser til firmaet, er det ikke lykkedes at få yderligere oplysninger om selve kortet eller den teknologi der ligger bag det, hvorfor dette løsningsforslag kun er kort beskrevet.

[78] Hjemmeside: http://www.quard.dk/

3.2.15 Stregkode

På næsten alle produkter som handles har en stregkode på sig. En stregkode er, som navnet siger, en kodet lavet af streger. Idéen med at bruge stregkoder i detailhandlen for at lette administration ved salg af varer, blev første gang taget i brug i 1973[79] med UPC[80]. Stregkoder indeholder ikke nogle direkte beskrivende data om produktet eller om folk (det der er på sygesikringskort eller anden ID-kort) men er blot et reference nummer, som computeren bruger til at hente de relevante data frem. På billederne nedenunder ses eksempler på stregkoder:

Figure 46: Eksempler på forskellige stregkoder[81]

Som det ses på billederne består en stregkode af en serie af streger med varierende tykkelse og med varierende afstand imellem. Kombinationer af forskellige stregtykkelser og forskellige afstande imellem repræsenterer forskellige karakterer. Hvilken karakter der er repræsenteret afhænger af, hvilken standard der bruges. De forskellige standarder har deres fordele og ulemper, eller eksisterer pga. politiske årsager. Tabellen på næste side viser en oversigt over de forskellige standarder, om variabel længde tillades samt, hvilke karakterer de kan repræsentere.

Det ses i tabellen, at nyere standarder understøtter variable længde samt hele ACSII sættet, hvilket de første og mest benyttede UPC og EAN ikke gør. Den mest benyttede af de nyere standarder er Code 39.

[79] Kilde: http://ask.yahoo.com/ask/20030630.html
[80] UPC: Universal Product Code (http://www.barcode-1.com/pub/russadam/upccode.html)
[81] Kilde: http://www.barcodehq.com/primer.html

System	Variabel længde	Karaktér repræsentation
Ældre systemer:		
Code 11	Ja	0-9
Codabar	Ja	0-9,$+.:/
Plessey	Ja	0-9,A-F
MSI	Ja	0-9
2 of 5	Ja	0-9
UPC og EAN	Nej	0-9
Nyere systemer:		
Code 39	Ja	0-9,A-Z./+-%$ (sammensat 2 karakter for hele ASCII)
Code 128	Ja	Hele ASCII
Code 93	Ja	0-9,A-Z./+-%$ (sammensat 2 karakter for hele ASCII)

Table 26: Oversigt over stregkode systemer[82]

En nyere udvikling indenfor stregkode er opfindelsen og anvendelse af 2D stregkoder. Dette er gjort da ovenstående standarder er begrænset hvad plads angår, og i stedet for at stregkoden kun er et reference nummer, vil man nu bruge stregkoden til at være en beskrivende tekst for produktet. I dag findes der over 20 forskellige 2D stregkode standarder. De vil ikke blive gennemgået her, blot vises eksempler på nogle af de mest benyttede:

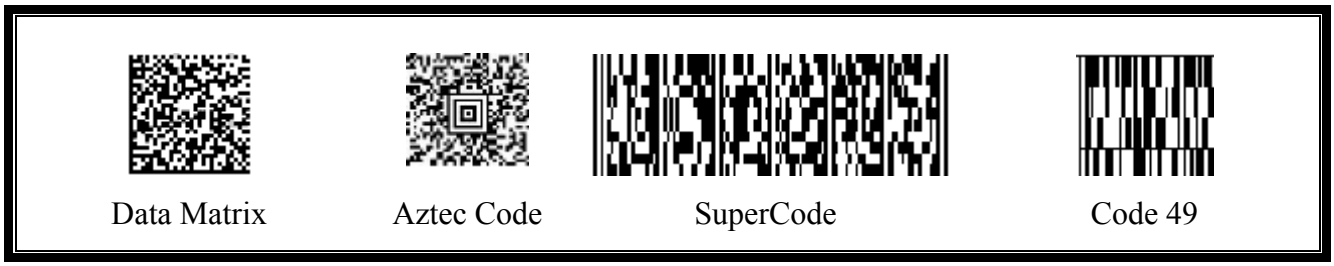


Figure 47: Eksempler på forskellige 2D stregkoder

Fordelene ved 2D stregkoderne er, at de kan indeholde langt mere data end de konventionelle stregkoder samt, at de indeholder flere fejlrettende koder. Efter at have undersøgt Internettet samt

[82] Kilde: http://www.barcodehq.com/primer.html

kontakt med et firma[83] som sælger stregkode læsere, har vi fundet frem til at stykprisen er på 18.780 kr. Derfor kan denne løsning ikke bruges.

[83] Kilde: http://www.symbol.com/products/oem/cse600.html

3.2.16 Magnetkort

Magnetkort er et kort som i dag er meget benyttet, pga. den billige teknologi. De fleste af os går rundt med flere magnet kort i vores pung, så som: sygesikringskort, telefonkort eller betalingskort som f.eks. Dankortet. Magnetkort er en hjælp til brugeren, da det bruges til at lagre oplysninger, som brugeren ellers skulle gå rundt og huske på. Fælles for disse kort er, at de blive skrevet på én gang og derefter aflæst mange gange. Det er de færreste applikationer, som ændre på de oplysninger der ligger på kortet.

Et magnet kort er et plastik kort som skal opfylde nogle helt bestemte krav. Disse krav er standardiseret efter ISO standarder. Nedenfor ses nogle af kravene samt hvilke ISO standarder, som skal følges:

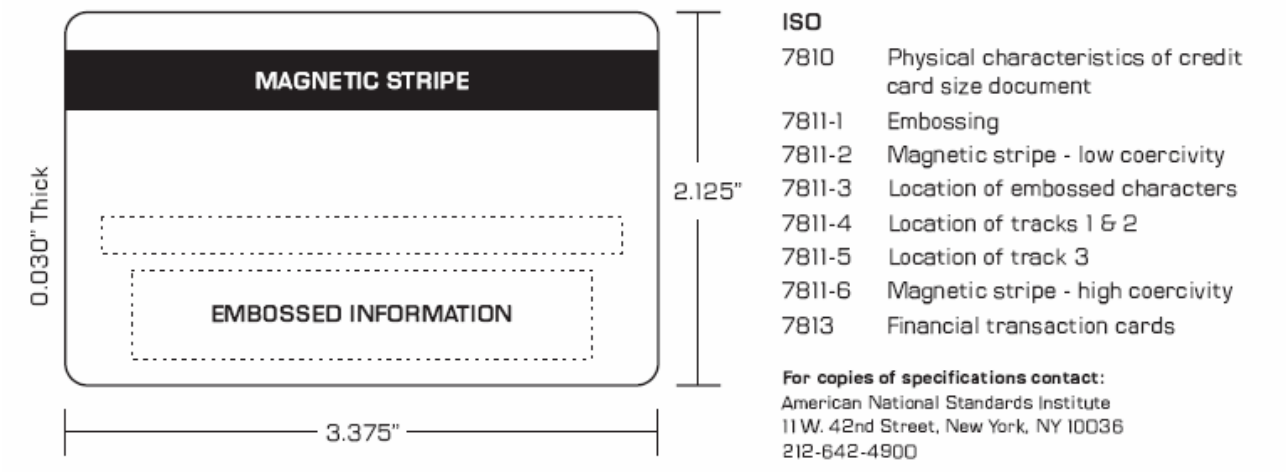


Figure 48: Magnetkort samt ISO standarder, som skal følges[84]

I magnetstriben er der defineret tre spor, som har forskellige karakteristika. Disse vises i nedenstående tabel:

Spor	Bittæthed (bits pr. tomme)	Karakter kodning (inklusiv paritets bit)	Informationsindhold
1	210	7 bits pr. karakter	79 karakterer
2	75	5 bits pr. karakter	40 karakterer
3	210	7 bits pr. karakter	107 karakterer

Table 27: Magnetkort karakteristika

[84] Kilde: http://www.magtek.com/documentation/public/99800004-1.pdf

De tre spor har forskellige formål, hvorfor de ikke er ens opbygget. Det første spor indeholder kortholderens navn, konto nr., udløbsdato, PIN-kode på krypteret form m.m. Her bruges der 7 bits pr. karakter, hvor den sidste bit er en paritets bit, så énbit fejl kan detekteres og rettes. Dvs. i dette spor findes i alt 474 (79 karakter * 6 bits/karakter) uafhængige bits.

Spor to er defineret af ABA (American Bank Association)[85] og er det mest benyttede spor. Det bruges i alle hæveautomater og hvor der ellers bruges kreditkort. Dette spor kan indeholde 40 4-bits karakterer (plus en paritetsbit), så der er i alt 160 (40 karakterer * 4 bits/karakter) uafhængige bits.
Det tredje spor er et lidt specielt spor. Det bruger den samme karakterkodning som spor et, men kan indeholde op til 107 karakterer. Meningen med dette spor var, at det både skulle læses samt skrives. Det skulle bruges til, at kontoinformationer skulle opdateres direkte på kortet. Derved kunne man verificere kontoen uden at maskinerne skulle være online, dvs. forbundet til en central. I dag er alle kreditkort maskiner/automater på en eller anden måde forbundet til en central, så anvendelsen af dette spor er ikke-eksisterende.

Den overordnede struktur for hvordan sporene skal læses er ens og følger en standard. Ligesom der for hver karakter er en paritetsbit, så alle enbit fejl kan detekteres og rettes, er der for hvert spor også en LRC (Longitudinal Redundancy Check). Derved kan flerbit fejl detekteres, men ikke rettes.
Måden en magnetkort læser skulle bruges på i dette projekt, er ved at både kontroller samt node har en dertil hørende aflæser. Når en node skal inkluderes i netværket, skal brugeren køre et vilkårligt magnetkort igennem på begge enheder. Ud fra de data som ligger på kortet, genereres en midlertidig nøgle som bruges til at overføre den endelige nøgle. Det smarteste er, at lave et kort som brugeren kan bruge. Dette gøres for, at brugeren ikke skal blive forvirret over at skulle bruge et Dankort eller lignende for at inkludere en node, men det er absolut ikke et krav. Man kan sige, at jo flere forskellige kort der bliver brugt/kan bruges, jo sværere er det for udefra kommende person at finde frem til de data, som er blevet brugt til at generere den midlertidige nøgle.

Efter at have undersøgt Internettet samt haft kontakt med et firma[86] som sælger disse magnetkort læsere, har vi fundet frem til at stykprisen (ved stor volumen) ligger på omkring 70 kr.

[85] Kilde: http://www.chez.com/mosfet/cread2.txt
[86] Sælger af magnetkort læser: http://www.adcomdata.dk

3.2.17 IR

Løsningsbeskrivelse:

Noder har én IR diode (emitter) og en trykknap og alle kontrollere har en IR modtager. Inkluderingen af noden ske som på sædvanlig vis ved, at en knap holdes nede på kontrolleren, mens der trykkes på noden. Når inkluderingen er fuldendt og noden skal have netværksnøglen, genererer noden en tilfældig nøgle, som den sender til kontrolleren via IR emitteren. Kontrolleren skal være placeret foran noden, så den kan modtage nøglen via sin IR modtager. Ved brug af 112 bits nøgle frem for 56 bits forlænges overførelsestiden, men det vil ikke være mærkbart, da der f.eks. overføres med 9.600 kbps. Forskellen i de forskellige løsninger ligger i, hvor direkte kontroller og node skal være placeret over for hinanden.

Beskrivelse af hardware:

IR Emitter 1:

5 mm IR diode
Udstrålingsvinkel: 16°
Bølgelængde: 880 nm
Fysiske mål: D: 5 mm H: 8,6 mm
Varenr.:F 212-672 (SFH484)
Pris: 2,69 kr.

IR Emitter 2:

5 mm IR diode.
Udstrålingsvinkel: 80°
Bølgelængde: 880 nm
Fysiske mål: D: 5 mm H: 8,6 mm
Varenr.: F 212-684 (SFH485)
Pris: 4,55 kr.

IR Detektor 1:

5 mm IR fototransistor

Åbningsvinkel: 20°

Bølgelængde: 850 nm

Fysiske mål: D: 5 mm H: 8,6 mm

Varenr.:F 881-934

Pris: 2,29 kr.

IR Detektor 2:

5 mm IR fotodiode

Åbningsvinkel: 150°

Bølgelængde: 850 nm

Fysiske mål: D: 5 mm H: 8,6 mm

Varenr.:F 212-726

Pris: 4,00 kr.

Opstilling af hardware:

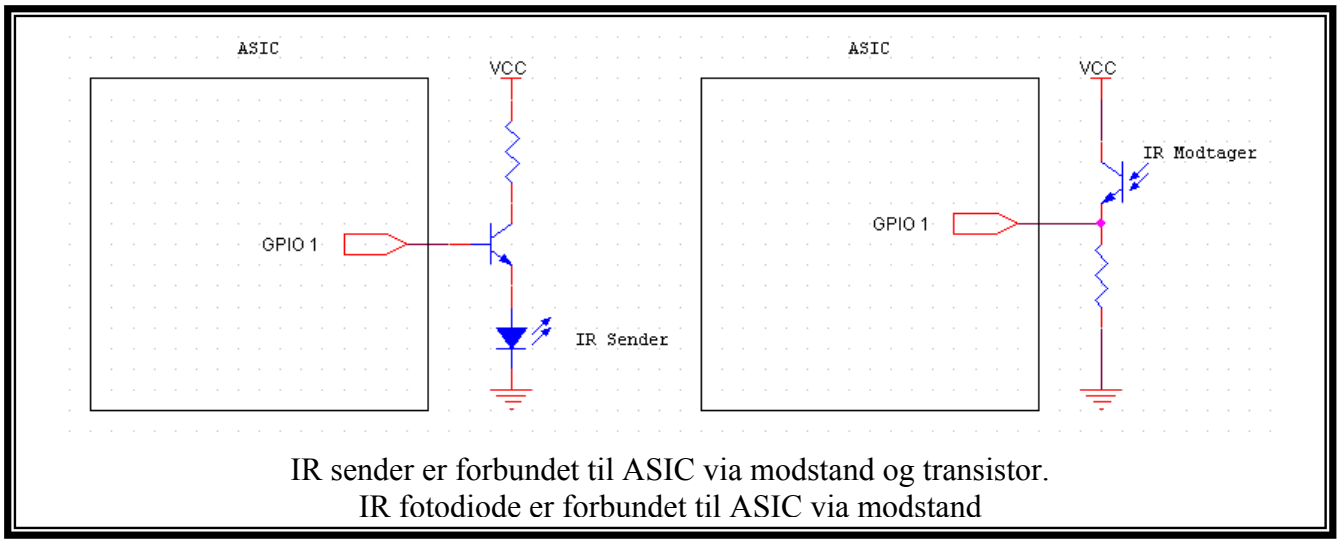


Figure 49: IR emitter og modtager tilsluttet ASIC

Baggrund for undersøgelse af løsningsforslag ved IR dioder:

Nedenfor gennemgås de forskellige parametre på løsningerne:

Vinkel:

For modtager og emitter opgives henholdsvis åbnings- og spredningsvinkel. Vinklen der opgives, er den samlede vinkel, som der spændes over. For at der ikke skal være nogen tvivl, er dette vist på nedenstående figur:

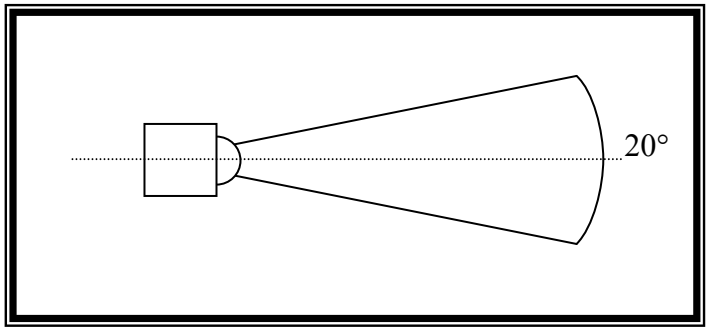
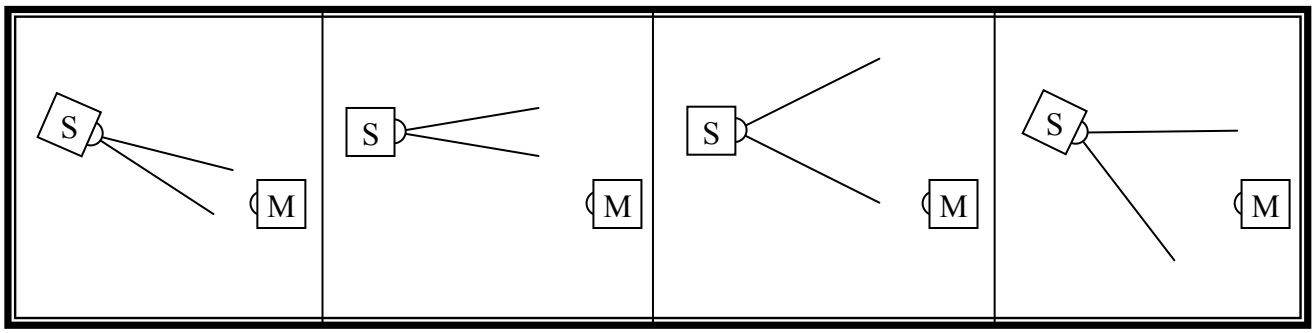


Figure 50: Åbnings- og spredningsvinkel

Ser vi først på emitteren, så angiver spredningsvinklen hvor stor en vinkel den udsendte stråling spredes over. Jo større spredningsvinkel, jo mindre retningsbestemt er senderen, dvs. des mindre præcist skal man pege mod modtageren for at denne kan opfange signalet. Dette vises på nedenstående figurer, hvor vi ser bort fra modtagerens åbningsvinkel:



S: Sender/Emitter, M: Modtager

Figure 51: Figurerne viser konsekvenserne af, når emitteren har stor eller lille spredningsvinkel

Ser vi på modtageren, så fortæller åbningsvinklen indenfor hvilket område modtageren er i stand til at opfange signaler fra. Med en lille åbningsvinkel skal emitteren være placeret mere ligefor for at kunne opfange, hvad der sendes. Har modtageren en stor åbningsvinkel, så kan den opfange

signaler der sendes mere skråt fra. Nedenstående figurer viser dette, hvor der ses bort fra emitteren's spredningsvinkel:

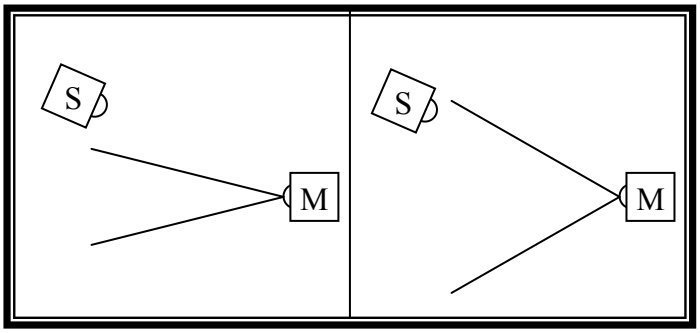


Figure 52: Figurerne viser konsekvenserne af, når modtageren har stor eller lille åbningsvinkel

Dvs. vi har, at åbningsvinklen siger noget om inden for, hvilket område emitteren skal befinde sig i for, at modtageren kan opfange signalet, hvorimod spredningsvinklen siger noget om, hvor præcist emitteren skal peges/rettes mod modtageren.

For nemmere at kunne sammenligne de forskellige løsninger, vil vi se lidt på frihedsgraden/ brugervenligheden. En stor frihedsgrad fås hvis både emitter samt modtager har store vinkler (spredning og åbning), da de (emitter og modtager) ikke absolut skal være placeret lige præcist over for hinanden "i lige linie". Omvendt er frihedsgraden lav, hvis emitter og modtager skal være i lige linie for at de kan kommunikere med hinanden.

Frihedsgraden udregnes ved at lægge den halve spredningsvinkel sammen med den halve åbningsvinkel . Da de begge ligger i intervallet mellem 0 og 180 grader, ligger frihedsgraden mellem 0 og 180 grader. Dette fremgår af følgende figur:

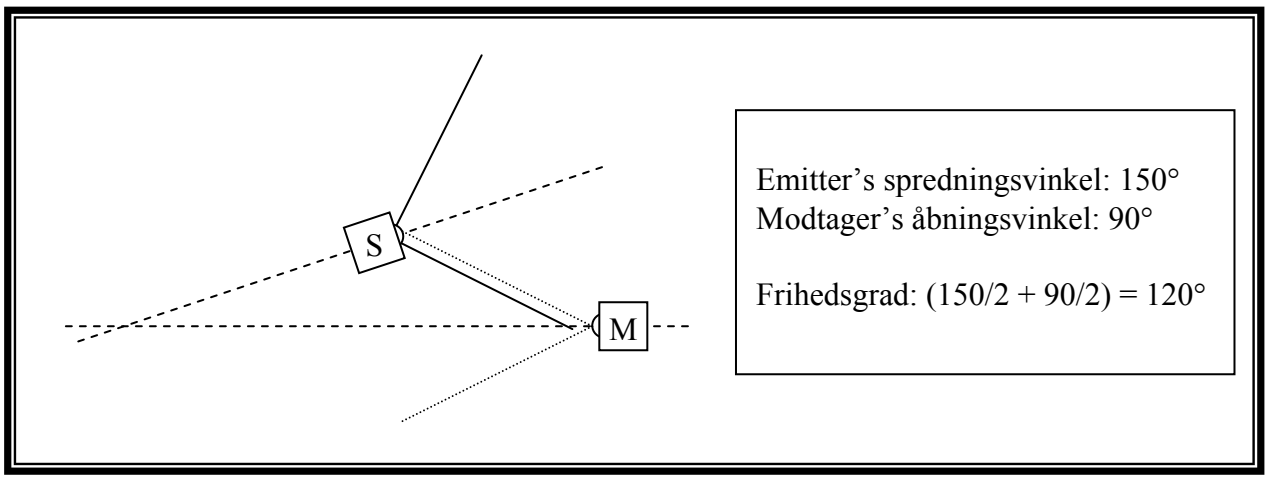


Figure 53: Eksempel på udregning af frihedsgrad

Ser vi på de sikkerhedsmæssige aspekter samt brugervenligheden, ser det således ud:

	Sikkerhed	*Brugervenlighed*
Stor spredningsvinkel	*Større risiko for at en angriber kan opfange signalet.*	*Modtageren skal ikke være placeret direkte foran emitteren.*
Lille spredningsvinkel	*Begrænset risiko for at en angriber kan opfange signalet.*	*Emitter og modtager skal være placeret direkte foran hinanden.*
Stor åbningsvinkel	*Større risiko for at en angriber kan jamme/forstyrre signalet.*	*Flere signaler/kilder kan opfanges samtidigt, som kan virke støjende.*
Lille åbningsvinkel	*Begrænset risiko for at en angriber kan jamme/forstyrre signalet.*	*Begrænset hvor mange signaler/kilder modtageren kan opfange.*

Table 28: Fordele og ulemper mht. sikkerhed og brugervenlighed ved stor/lille emitter-/åbningsvinkel

Som det fremgår at ovenstående oversigt, så er valget en afvejning af om man vægter sikkerhed eller brugervenligheden højest. En ting man også skal have i tankerne er, at man kun inkluderer én enhed af gangen, så der vil reelt ikke være andre kilder (signaler fra andre noder) som kan virke støjende.

Bølgelængde:

Denne værdi angiver hvor i det infrarøde spektrum, at emitteren afgiver mest energi og hvor modtageren er mest følsom. Generelt kan man sige, at jo større overensstemmelse der er mellem senderens og modtagerens spidsbølgelængde, des bedre er kvaliteten af signalet, da mere af det den afsendte energi opfanges af modtageren. Nedenfor ses følsomheden for detektor 1 samt udstrålingsspektret for emitter 1, begge som funktion af bølgelængden (nm).

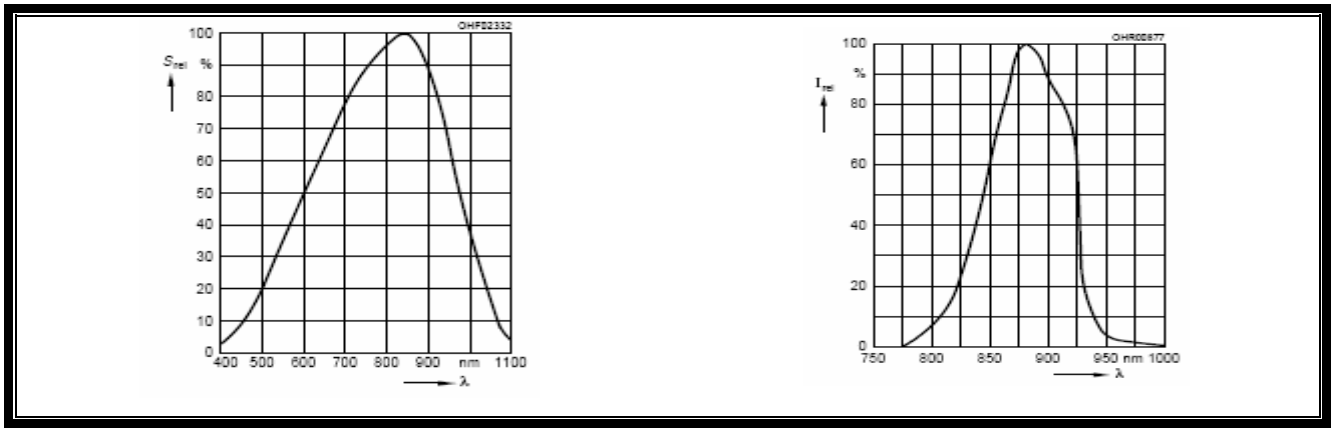


Figure 54: Følsomheds- og udstrålingsspektre for IR modtager og sender

Det ses på figurerne, at spredningen er stor, så selvom der bruges en modtager, som er mest følsom ved 850 nm, og en emitter der udsender mest ved 880 nm, så er overlappet mellem spektrene så stort, at der ikke vil være nogen problemer.

Udstrålingseffekt:

Jo større effekt, des længere rækkevidde har emitteren, men der kommer også flere/kraftigere refleksioner. Hvis vi tager udgangspunkt at rækkevidden kun skal være på et par meter, har dette ingen praktisk betydning.

Filter:

For at skærme mod sollys, som indeholder infrarødt lys, kan både emitter og modtager have filter. Modtager uden filter kan have svære ved at modtage IR signaler, hvis solen "står lige på". Tager vi udgangspunkt i, at de fleste sikre noder vil befinde sig indendørs samt at den initiale nøgleudveksling kun foretages én gang, har filtret ikke den store betydning.

Løsningsforslagene er lavet på baggrund af gennemgangen af retningsbestemtheden:

Løsning:	*Spredningsvinkel*	*Åbningsvinkel*
1	*Lille*	*Lille*
2	*Lille*	*Stor*
3	*Stor*	*Lille*
4	*Stor*	*Stor*

Table 29: Oversigt over sprednings- og åbningsvinkel for løsninger

Løsning 1:

Til denne løsning bruges en SFH484 som sender og en SFH313FA som modtager.

Det første vi ser på er, at spidsbølgelængden på senderen er på 880 nm og for modtageren er den på 870 nm. Den lille forskel på kun 10 nm har ingen praktisk betydning, så hvad dette angår, må man betegne løsningen som værende god. Ser vi på emissionsvinklen så er den på 16 grader og for modtagerens vedkommende er åbningsvinklen på 20 grader. Derved bliver frihedsgraden lav, da sender og modtager højest må være forskudt med 18 (16/2 + 20/2) grader for, at vi er sikker på en god forbindelse. Emitteren udsender fra 80 til 160 mW/sr ved I_f = 100 mA, og betegnes derved som værende kraftig, dvs. at vi har mulighed for at opnå en stor rækkevidde.

Løsning 2:

Med udgangspunkt i løsning 1, skiftes modtageren ud med en anden model (SFH203P), som har en meget stor åbningsvinkel. Spidsbølgelængden for modtageren er på 850 nm, hvor den for emitteren er på 880 nm. I løsning 1 var forskellen på 1,1 % (10 nm) mens den her er 3,5 % (30 nm). Om det i praksis har nogen betydning er svært at sige, men det er værd at bemærke. Emissionsvinklen er på 16 grader mens åbningsvinklen er på 150 grader, dette medfører en frihedsgrad på 83 (16/2 + 150/2) grader.

Løsning 3:

Til forskel fra løsning 1, bruges der her en anden emitter men med den samme modtager. Emitteren er en SFH485. Denne har en emissionsvinkel på 40 grader og derved fås en frihedsgrad på 30 (40/2 + 20/2) grader. Forskellen på spidsbølgelængden er på 1,1 % (880 nm for emitter og 870 for modtager). Emitteren udsender mellem 25 og 50 mW/sr ved I_f = 100 mA, hvilket sætter en begrænsning på rækkevidden.

Løsning 4:

Denne løsning tager udgangspunkt i stor spredningsvinkel og stor åbningsvinkel, derfor bruges der en SFH485 som emitter og en SFH203P som modtager. Ligesom i løsning 2 er der her en forskel 3,5 % hvad angår forskellen på spidsbølgelængden. Frihedsgrad er på 95 (40/2 + 150/2) grader og er den største.

Løsningsoversigt:

#	*Vinkel*	*Stk. Pris*	*Bølgelængde (nm)*	*Udstrålingseffekt (mW)*	*Modtagerfilter*
1	*16 / 20*	*2,69 / 2,29*	*880 / 850*	*80 – 160*	*Nej*
2	*16 / 150*	*2,69 / 4,00*	*880 / 850*	*80 – 160*	*Nej*
3	*80 / 20*	*4,55 / 2,29*	*880 / 850*	*25 – 50*	*Nej*
4	*80 / 150*	*4,55 / 4,00*	*880 / 850*	*25 – 50*	*Nej*

Table 30: Løsningsoversigt ved brug af IR

Stk. pris – Pris for emitter/modtager

3.2.18 Laser

Laser står for ”Light Amplification by Stimulated Emission of Radiation” og er meget anvendt i hverdagen, f.eks. i ”pegepinde”, dvd- og cd-afspillere, hastighedsmålere m.m. Laser er koncentreret lys på et meget lille område. Idéelt set har laser ingen spredning, så den vil ”oplyse” det samme område uanset afstand. I virkeligheden sker der en spredning, men den er så lille, at den kan negligeres.

Da lasere har mange anvendelsesmuligheder, findes der mange forskellige måder at generere lyset på. Størrelsen på lasere variere fra store anlæg, hvor laseren skal bruges til at skære med og derfor kræver flydende kvælstof til nedkøling og ned til laser dioder, som bruges i cd-afspillere og pegepinde. Det er de sidstnævnte vi er interesseret i.

3.2.19 Ultralyd

Ultralyd er lydbølger med en frekvens som er større end 20 kHz, dvs. det er lyde som det menneskelige øre ikke kan opfatte. Selv ved høje intensiteter har ultralyd minimal effekt på mennesker[87]. Dyr som f.eks. hunde og katte kan godt høre ultralyd. Dette kunne man se ved gamle B&O fjernsyn, som brugte ultralyd i fjernbetjeningen.

Ultralyd har vundet stor udbredelse i industrien, men bruges ikke til kommunikation pga. den lave båndbrede. Ultralyd bruges primært til at undersøge forskellige objekter for hvordan overgangen er mellem forskellige materialer. Ved hver overgang sker der både en transmission samt en reflektion af lydbølgerne. Foruden materialeundersøgelse bruges ultralyd til afstands- og hastighedsmåling. Ved hastighedsmåling udnyttes Doppler effekten[88], som er en forskydning af frekvensen af den bølge som rammer et legme i bevægelse.

Som sagt bruges ultralyd ikke til kommunikation, kun meget simple/gamle applikationer har brugt ultralyd til at tænde/slukke. Foruden B&O´s fjernbetjening findes andre eksempler så som bevægelsesdetektorer.

Det har ikke været muligt ved søgning på Internettet, at finde noget brugbart materiale om brug af ultralyd til kommunikationsformål, hvorfor vi må udelukke dette som en løsning.

[87] Kilde: http://www.encyclopedia.com/html/u1/ultrason.asp
[88] Kilde: http://ej.rsna.org/ej3/0079-98.fin/doppler.htm

3.2.20 PAN

PAN står for Personal Area Network og dækker over udveksling af data mellem to mennesker via kroppen. PAN er udsprunget fra forskning ved MIT's (Massachusetts Institutes of Technology) Media Laboratorium og hovedmanden bag idéen er Thomas G. Zimmerman[89]. Idéen går ud på at bruge kroppen som kommunikationsmedium ved at føre en meget lille strøm igennem den. Kroppens naturlige saltindhold gør den velegnet til at lede strømmen. Strømstyrken er på en nano Ampere, hvilket er mere end tusind gange mindre end den strøm der genereres, når man reder sit hår.

PAN er baseret på ISO 7498 netværksstandarden og består af en sender og en modtager. Enhederne får strøm fra batterier og har to elektroder. Den ene elektrode er i direkte kontakt med huden mens den anden ikke må have forbindelse med kroppen. Senderen sender en moduleret strøm gennemkroppen med en frekvens, der er mindre end 1 MHz. Pga. den lille strøm og lave frekvens, er spredning af signalet meget beskedent, hvilket vanskeliggøre aflytning, og det forstyrre andre PAN minimalt. Returvejen for strømmen er gennem luften samt andre ledende eller dielektriske materialer, som findes i nærheden af kroppen. Returvejen for strømmen skal være isoleret fra resten af kroppen, da man ellers vil få en kortslutning, og ingen kommunikation kan finde sted.

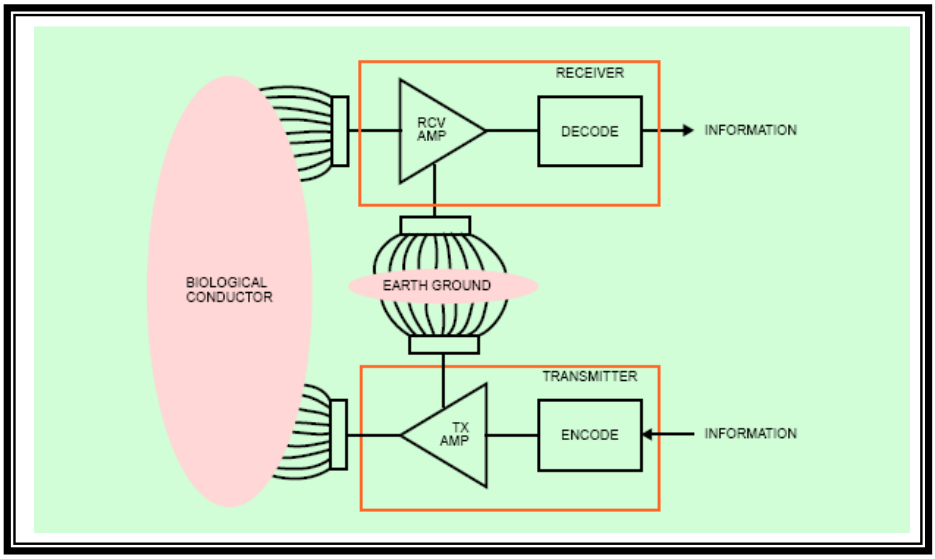


Figure 55: Skematisk oversigt over hvordan PAN virker gennem kroppen (Biological conductor) sammen med det omkringværende medium (earth ground)[90]

[89] Website: http://www.almaden.ibm.com/cs/user/pan/pan.html
[90] Figuren er kopieret fra artiklen af Thomas G. Zimmerman "Personal Area Networks: Near-field intrabody communication"; IBM Systems Journal, Vol 35, NOS 3&4, 1996

Den effektive båndbrede opgives til at være omkring 2400 bits/s, hvilket er mere end rigeligt til vores anvendelse, da nøglen er 56/112 bits. For at bruge PAN til vores formål vil det være med en ledende flade/plade på både slave og kontroller. Når inkluderingen så skal foretages, skal brugeren bare rører begge plader på samme tid. Denne løsning er meget attraktiv, da strømmen er meget svag og frekvensen lav, så aflytning vil i praksis være umuligt.

Forskningen inden for dette område er stoppet og det har ikke været muligt at få yderligere informationer fra Thomas G. Zimmerman, som kan realiseres til en implementering.

3.2.21 Opsummering af initial nøgleudveksling

I de ovenstående afsnit er gennemgået forskellige forslag til, hvordan den initiale nøgleudveksling mellem kontroller og node kan foregå. Først blev to in-band metoder beskrevet og dernæst out-of-band metoder.

Den første metode gør brug af reduceret sendestyrke i forhold til normal styrke, mens i den anden er antennen på kontrolleren retningsbestemt. Fælles for disse to metoder er, at det er muligt for tredjepart at aflytte kommunikationen vha. retningsbestemt antenne samt god forstærkning. De to metoder kan kombineres, men det besværliggøre kun aflytningen, ikke umuliggøre. Det anbefales ikke at bruge nogle af disse metoder.

De efterfølgende metoder er alle out-of-band metoder, hvor fordele og ulemper er beskrevet. Ved brug af forudprogrammeret nøgle lægges der op til, at det er installatører der foretager installationen og inkluderingsproceduren forgår i et ”lukket” miljø. Løsningen kræver ingen yderligere komponenter men derimod administration af, hvilke noder der har hvilken nøgle. Løsningen binder hele setup´et til, at brugeren ikke selv kan købe nye noder og sætte op, hvilket gør løsningen ufleksibel.

I næste metode der gennemgås, bruges en ledning til at forbinde de to enheder for at få overført nøglen. Løsningen er sikker, da det ikke vil være muligt at aflytte trafikken og udvekslingen kan foregå meget hurtigt. Der er set på forskellige stiktyper, hvor løsningen der bruger mini jack er værd at overveje, da det er nemt, hurtigt og småt. Designmæssig vil det ikke være strengt nødvendigt, at gemme stikket bag en klap e. lign. Stiktypen ”Modular 4/4” har den ulempe, at det låser stikket til fatningen og tappen, der låser, kan nemt brække af. Problemet er kun af kosmetisk karakter, da overførelsen foregår så hurtig, så stikket skal ikke være i fatningen i lang tid.

En anden metode at overfører den initiale nøgle er ved, at bruge SIL/DIL kontakter, som brugeren skal sætte i forskellige stillinger. Kontakterne er meget små, især SIL kontakterne, så det kan være svært at sætte dem rigtigt med en skruetrække eller kuglepen. På den anden side skal dette kun gøres én gang i enhedens levetid. Producenten af enheder vil nok foretrække, at placere disse bag en klap, da de er lidt skrøbelige og ikke videre kønne.

Noderne kan være udstyret med flere trykknapper, som skal holdes nede mens der trykkes på en anden knap. Det kan være svært, at afgøre om ”de rigtige” knapper er trykket ned især, hvis det er fire knapper det skal gøres med. Jo flere knapper der skal holdes nede, des større er risikoen for at lave fejl, men jo færre gange skal det gøres.

Løsningerne hvor et tastatur bliver brugt, kan ikke bruges i denne sammenhæng af flere årsager. For det første fylder de meget og for det andet er de utrolig dyre.

En anden måde at overføre den initiale nøgle på er ved, at gøre brug af en drejeomskifter. Drejeomskifteren har både fordele og ulemper. Den er lille, så det er designmæssigt ikke strengt nødvendigt at skjule hullet til den, og den flytter sig i ryk når der skiftes position. Bruges den med få stillinger, skal den stilles mange gange, hvorimod dem med mange positioner ikke skal sættes så mange gange, men det kan være svært at afgøre hvilken position den står i.

Løsningerne med trimmer minder meget om førnævnte bortset fra, at de stilles flydende og ikke rykker i hak. Der er to versioner med hver deres præcision samt vinkel, der kan drejes over. Hvis trimmeren kun skal indstilles i få positioner kan den billige model bruges (med ”dårlig” præcision og mindre drejningsvinkel), men skal den inddeles i mange positioner skal den dyre model bruges (som er mere præcis og spænder over en større vinkel). Ser vi designmæssigt på det, så kan tilgangen til trimmeren f.eks. foregå vha. en knap. Bruges en version hvor der skal bruges skruetrækker, skal hullet til den ikke være særlig stort, og det vil ikke være nødvendigt at gemme den bag en klap.

Farvet lysdioder på noden kan bruges til at vise, hvad brugeren skal indtaste på kontrolleren. Løsningerne der baserer sig på lysdioder kræver minimal plads pga. den lille størrelse, og de kan fås i flere forskellige farver, så de designmæssigt passer til resten af applikationen, hvorved der ikke er nødvendigt, at gemme den af vejen. Op til tre-farvet (rød/grøn/blå) dioder er billige, hvorimod full-color dioden er dyr. Bruges denne diodetype til at vise mange farver, er det vigtigt at der er stor kontrast mellem farverne, så brugere ikke er i tvivl. Løsningerne her er absolut værd at overveje.

I stedet for lysdioder, kan man bruge 7-segmenter, som kendes fra bla. klokradioer. 7-segmenter fylder mere end lysdioder og kan kun fås i rødt eller grønt. Funktionsmæssigt løser de opgaven på en tilfredsstillende måde, hvorimod de dessignmæssigt ikke er særlig pæne, hvorfor de typisk vil blive gemt bag en klap.

De næste løsninger vi har set på, er løsninger hvor brugeren kun skal foretage sig noget aktivt én gang. Den første løsning baserer sig på, at bruge Smart Card. Her skal både kontroller og node have en kortenhed. Overførelsen af nøglen foregår hurtigt og nemt. På kortet lægges den midlertidige nøgle, som kun er gyldig i et lille tidsrum. Fysisk set fylder kortenheden meget og vil sandsynligvis blive gemt af vejen, så den kun ses/bruges ved inkluderingsproceduren.

Aflæsning af fingeraftryk er en anden mulighed. Et dansk firma har udviklet en fingeraftrykslæser, som kan være i et Smart Card. Bruges denne metode skal alle noder samt kontrollere have en aflæser. Det er dog ikke lykkes, efter flere forsøg, at få nogle brugbare informationer om teknologien.

En anden form for aflæsning er aflæsning af stregkoder. Forskellige systemer er beskrevet og der er fundet priser for små moduler, som desværre må konstateres at være for dyre til at komme i betragtning. Det samme gør sig gældende for den næste løsning der er beskrevet; magnetkort.

De sidste fire metoder har alle det tilfældes, at de bruger luften som overføringsmedium. Den første foregår vha. IR. I beskrivelsen er gennemgået hvilke konsekvenser forskellige kombinationer af emitter/modtager har mht. hvor brugervenlige de er. Der skal ikke meget ekstra hardware end selve IR-enhederne (som i forvejen er små) og det er muligt, at bestemme hvor retningsbestemt det hele skal være. Denne løsningsmetode er værd at overveje.

De to næste metoder, er ved at bruge Laser eller Ultralyd. Laser har den fordel, at den er meget retningsbestemt, men den ulempe at den kan gøre skade hvis den rettes mod øjnene. Det har ikke været muligt, at finde noget brugbart om ultralyd, men det kan konstateres at det er en gammel teknologi som ikke er særlig retningsbestemt og aflytning vil absolut være muligt. Disse to metoder bør ikke bruges. Den sidste mulighed er PAN, men forskning indenfor dette område er afsluttet og det har ikke været muligt, at få noget brugbart materiale, hvorfor metoden ikke kan bruges.

Ovenfor er opsummeret hvilke fordele og ulemper, der er ved de forskellige metoder, men prisen er ikke taget med i betragtning, derfor vil vi nu se nærmere på sammenhængen mellem prisen og mapningen. De metoder, hvor overførelsen sker på én gang, er der ingen mapning hvorfor de ikke kan medtages. Nedenstående figur viser sammenhængen mellem mapning og pris:

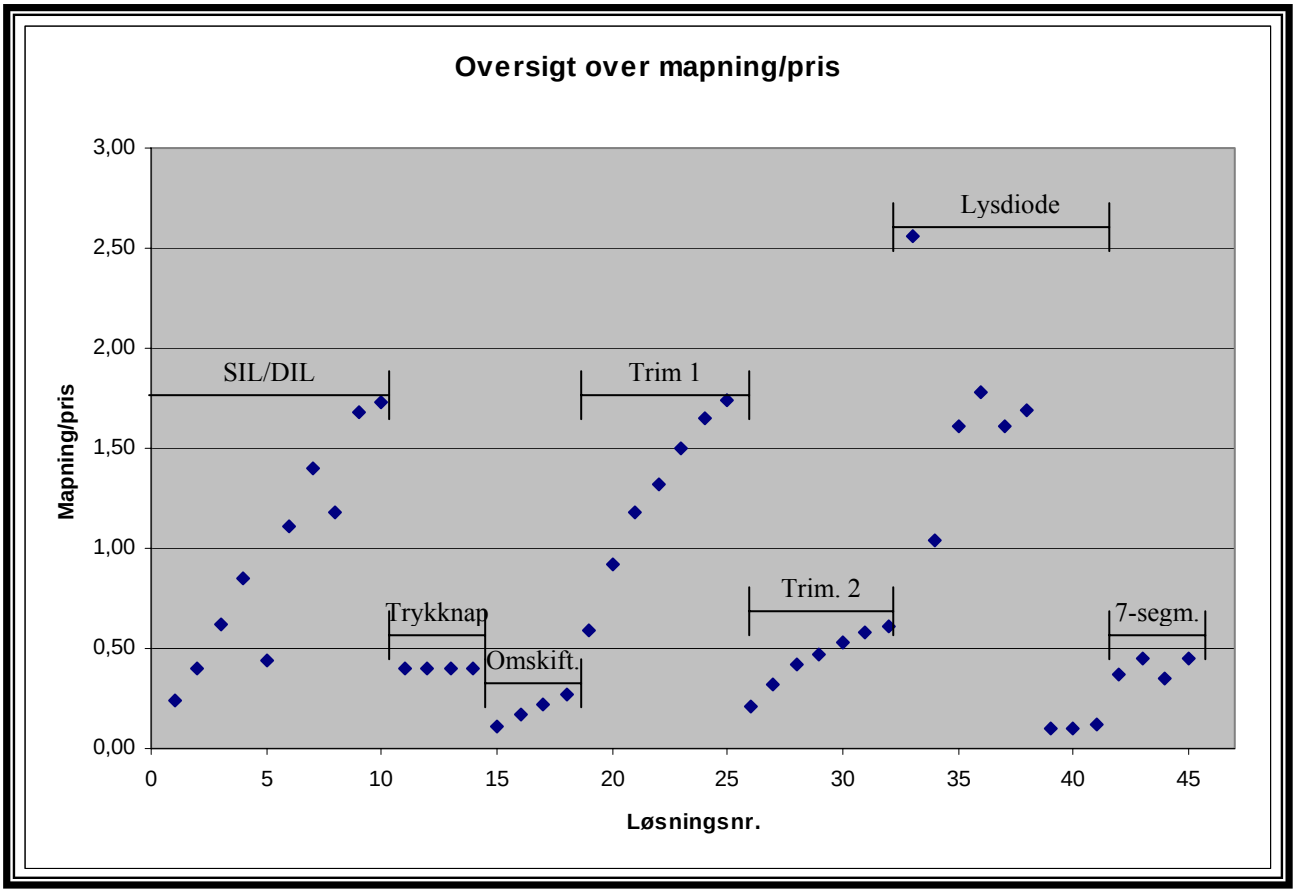


Table 31: Sammenhæng mellem mapning og pris

De løsninger hvor mapningen er stor i forhold til prisen, er placeret øverst på figuren, og de løsninger som er dyre i forhold til mapningen, er nederst. Tager vi dem fra en ende af, kan vi se at SIL/DIL kontakterne stiger mere end lineært, da mapning/pris forholdet stiger hver gang der kommer flere kontakter på. Vi starter med et forhold på omkring 0,25 og ender ved omkring 1,75. Prisen for de første fire (SIL) går fra omkring 8,50 til 9,50 kr., hvor det for DIL kontakterne (de sidste 6) går fra omkring 3,50 til 6,50 kr.

Den næste samling er trykknapper, hvor vi har en lineær sammenhæng. Det har vi, da hver gang der bliver tilføjet én kontakt til løsningen, kan vi repræsentere én bit mere af gangen. Vi lægger mærke til, at forholdet er meget lavet på 0,4 og stykprisen ligger på omkring 2,50 kr.

Næste gruppe er omskiftere, der er er meget dyre (9 til 15 kr.) hvorfor mapning/pris forholdet er lavt fra 0,1 til 0,3.

De næste to grupper består af to modeller af trimmere. Den billige model (mindre præcis, mindre drejningsvinkel) koster 1,70 kr. og forholdet går fra 0,6 til 1,75. Da den totale drejningsvinkel er lille er det dog tvivlsom, hvorvidt de sidste tre løsninger kan bruges, da det simpelthen vil være for svært at ramme den rigtige position. Den anden trimmer koster 4,80 kr. med et forhold der spænder fra 0,2 til 0,6. Denne model er mere realistisk at bruge, hvis mange positioner skal bruges, så kun få indstillinger skal foretages.

Farvet lysdioder rangerer højt på figuren. Især den første løsning skiller sig ud fra mængden og det skyldes den lave pris på kun 40 øre. Den næste har en pris på omkring en krone med en mapning på kun 1, hvorfor den ligger længere nede. Derefter er der en mindre gruppe på 4, hvor det er den samme lysdiode der bruges med en stk. pris på 83 øre og en mapning på omkring 1,4. De sidste tre lysdiode løsninger bruger en meget dyr diode som koster omkring 20 kr. hvorfor forholdet er så lavt.

Sidste metodegruppe er ved brug af 7-segmenter, hvor forholdet ligger på 0,4. Dette skyldes stykprisen på omkring 9 kr.

Vi må konkludere, at løsningerne hvor en én/to-farvet lysdiode bruges er meget attraktiv, da prisen er meget lav og de hardware- og designmæssige ændringer af en eksisterende applikation er minimale. Fysisk set er denne løsning, den der fylder mindst af alle gennemgået out-of-band løsninger.

Er man ikke interesseret i at bruge lysdiode, vil den bedste løsning være, at bruge DIL kontakter. De spænder fra omkring 3,50 til 6,50 kr. og har et højt mapning/pris forhold. Bruges den længste række (12 kontakter) skal den kun sættes 5 gange. DIL kontakterne fylder noget mere end en lysdiode og skal nok sættes bag en klap.

Ønsker man en løsning, hvor brugeren skal foretage sig minimalt, må IR-løsningen anbefales. Som udgangspunkt er den ”billige” del lagt på noden med en pris på 2,65/4,15 kr. (emitter), mens modtageren på kontrolleren er på 4,65/7,10 kr. for, at holde den lave pris på ”high volume” området (noderne) mens den høje pris er på ”low volume” området. Løsningen har den fordel, at man ikke nødvendigvis skal stå med begge enheder (kontroller og node) i hånden ved inkluderingen. Valg af emitter og modtager kan gøre løsningen meget retningsbestemt, som beskrevet.

3.3 Replay attacks

Mange angreb mod netværkssikkerhed er baseret på opsnapning (eavesdroping) af meddelelser/data under transaktion mellem to kommunikerende parter, sådan en kommunikation kan f.eks. foregår mellem en klient og en server. En af angrebsformerne er replay attack. Et Replay attack foregår ved, at en angriber opsnapper en meddelelse under en transaktion, og derefter bruger denne eller dele af meddelelsen igen på et senere tidspunkt (Se figuren nedenunder). Et replay attack finder generelt sted uden, at de to kommunikerende parter opdager det da de tror, at de kommunikerer med hinanden.

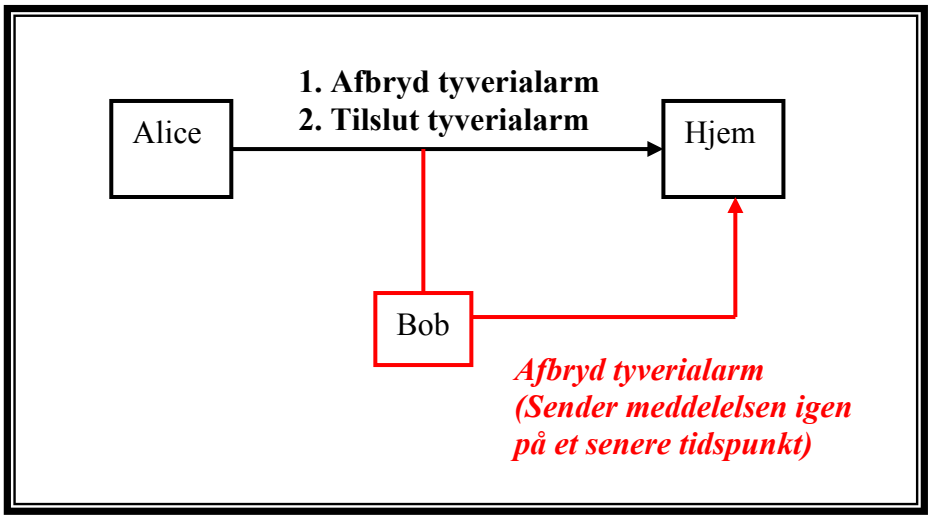


Figure 56: Replay Attack

Figuren ovenpå viser et sandsynligt replay attack. Alice opdager, efter hun tilsluttede tyverialarmen, at hun har glemt sin mobiltelefon, så hun afbryder tyverialarmen for at hente sin mobiltelefon, og efter hun har hentet den tilslutter hun tyverialarmen igen. Bob er en angriber, som opsnapper data/meddelelsen fra de to kommunikerende parter. Han kan vælge, at sende samme meddelelse igen efter et stykke tid (f.eks. efter nogle minutter eller timer), eller han kan bruge dele af meddelelsen og sende den videre (modifikation).

Der findes metoder til at undgå/modvirke Replay attacks, som kan hjælpe med at opdage om en meddelelse er replay eller ej og dermed afvise de replayed meddelelser. En af metoderne til at modvirke Replay attacks er Challenge/Response teknikken, som bliver detaljeret beskrevet i næste afsnit. Her skal man være opmærksom på, at de to kommunikerende parter først og fremmest skal

sørge for modifikation af de afsendte meddelelser, som kan gøres ved at bruge kryptering. Ved brug af kryptering kan man nemlig gøre det svært, for andre personer (f.eks. angriber), som har adgang til kommunikationskanalen, at forfalske en tidsvarierende parameter (tilfældigt nummer, sekvensnumre eller Time Stamp) som adderes i meddelelsen for at undgå replay attacks. Den tidsvarierende parameter er et tal, som skal findes i alle meddelelser og som hele tiden skal være forskellig.

De tre forskellige metoder til at modvirke replay attacks introduceres i det næste afsnit, hvorefter de beskrives i dybden i de efterfølgende afsnit.

3.3.1 Generelle metoder til at forhindre replay attacks

For at forhindre replay attacks, skal alle meddelelser som modtages undersøges for at afgøre om meddelelsen er replay eller ej. De tre brugbare metoder til detektering af replay attacks er:

- ***Udfordring/Svar (Challenge/Response)***
  Denne metode går ud på, at afsender af meddelelsen vedhæfter et nummer eller en streng. Modtageren af meddelelsen svarer tilbage med sit svar vedhæftet det oprindelig afsendte nummer (challenge). Da meddelelsen er krypteret, kan tredjepart ikke se hvad udfordringen er, og kan derfor ikke svarer tilbage med dette/denne nummer/streng. Hvad udfordringen er, er ikke vigtigt. Det kan f.eks. være et *Time-Stamp,* en *nonce* (et *tilfældigt tal*), et *sekvensnummer* eller en *tidsafhængig funktion*.

- ***Sekvensnummer (Sequence Number)***
  Et sekvensnummer er et unikt nummer, f.eks. et fortløbende serienummer, der adderes til hver meddelelse og bruges til at identificere afsendte data. Dvs. vha. sekvensnummeret, kan modtageren af data holde styr på om de modtagne meddelelser er replay meddelelser eller ej. Sekvensnummeret forøges med én ved hver ny meddelelse, så er alle parter (afsender og modtager) klar over hvilket tal den næste ”legale” meddelelse vil indeholde. Meddelelser med andet nummer end det forventede ses der bort fra.

- ***Tidskode (Time-Stamp).***
  En sidste metode for at modvirke replay attack fungere ved, at den tidsvarierende parameter, består af et time-stemp. Dvs. den indeholder en angivelse af tiden, da meddelelsen blev sendt afsted. Når meddelelsen bliver modtaget af modtageren, sammenligner modtageren tiden i meddelelsen med sin egen tid da meddelelsen blev modtaget. Derefter bedømmer han om meddelelsen er et replay eller ej. Her skal der holdes styr på tiderne, dvs. urene på enhederne skal synkroniseres, så de går ens inden for rimelighedens grænser.

De ovenstående metoder/løsninger til, at undgå replay attacks bliver beskrevet i detaljer i de efterfølgende afsnit. Først beskrives Challenge/Response metoden, derefter sekvensnummer metoden og til sidst Time-Stamp metoden.

3.3.1.1 Udfordring/Svar (Challenge/Response)

En Challenge/Response[91] protokol virker ved følgende;

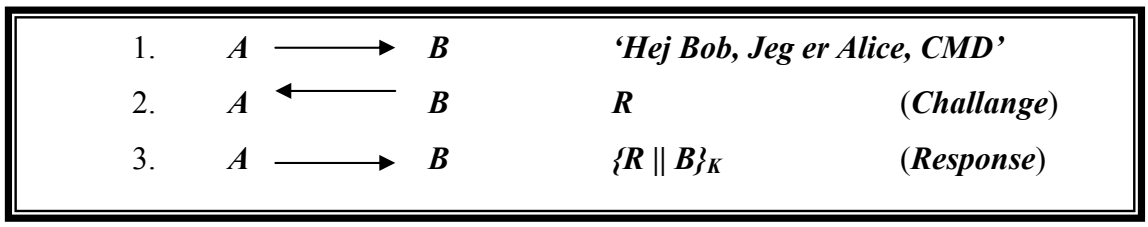


Figure 57: Challenge/Response protokol

Her gælder det, at Alice ***A*** (Klient) og Bob ***B*** (Server) starter med, at dele en hemmelig nøgle ***K*** mellem sig. Hvordan dette foregår, er i denne sammenhæng underordnet. Alice sender en initial meddelelse til Bob, indeholdende hendes, Bobs identifikation og en **CMD,** kommando (ønske). Så sender Bob en ***Challange*** meddelelse ***R*** til Alice og Alice svarer tilbage med et ***Response,*** nemlig ***{R || B}$_K$*** (Meddelelse ***R*** er sammensat med ***B*** og krypteret med nøglen ***K***). Bob dekrypterer Alices Response og kontrollerer om han får ***R || B***. Hvis han får ***R || B,*** så accepterer han Alice og hendes CMD (kommando) ellers ikke.

I denne protokol kan en angriber "se" ***R*** og ***{R || B}$_K$,*** men har ikke kendskab til nøglen ***K*** og dermed ***R || B*** og kan derfor ikke svare korrekt tilbage til Bob's ***Challange***. Bob får sit ***R*** tilbage i en krypteret form, som kun kan krypteres af Alice, da hun er den eneste der kender nøglen ***K***, og dette sørger for meddelelsesintegritet. Det skal ikke være muligt for en angriber X, at kunne forudsige Bob's challenge ***R***, da han ellers kan spille Alice over for Bob (dvs. man-in-the-middle). Et eksempel på hvordan angrebet fungere i praksis, følger her:

Angriberen lytter med i kommunikationen og opsnapper meddelelserne mellem A og B. De meddelelser han samler, kan han på et senere tidspunkt bruge til at sende igen. Figuren på næste side viser et replay attack i Challange/Response protokollen (hvis Challenge/Response ikke bruges for hver sendte meddelelse) for udfordringen er hele tiden den samme. (På figuren nedenunder betyder ***X(A)*** at angriberen ***X*** spiller ***A*** over for ***B***).

[91] Kilde: Handbook of applied cryptography, A. Menezes, P. van Oorschot, S. Vanstone, CRC Press, 1997
og Cryptography: Theory and Practive, D. R. Stinson, CRC Press, 1995

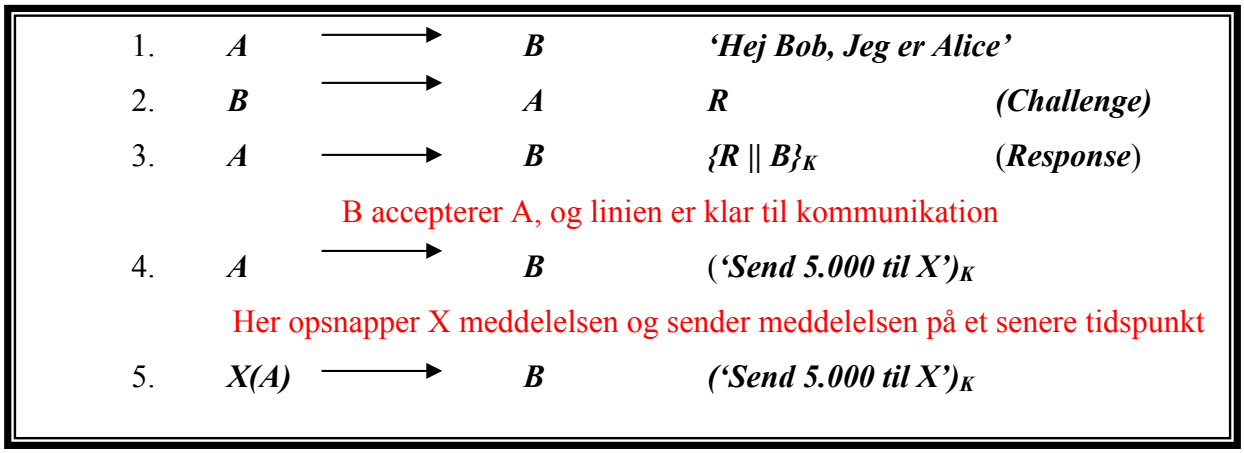


Figure 58: Replay Attack i Challenge/Response

Det ovenstående replay attack viser, at meddelelsesintegritet ikke er nok for at forhindre angrebet, derfor skal man kontrollere at meddelelsen ikke er blevet brugt før og at meddelelsen er genereret indenfor en accepteret tidsramme (kaldes Freshness). Derfor kan Challenge/Response protokollen bruges til, at beskytte sig mod replay atttacks, da dette kan nås ved, at bruge en challenge værdi som er tidsafhængig. Generelle tidsafhængige challenge værdier er: et nonce (et tilfældigt tal), et sekvensnummer eller en tidsfunktion Time-Stamp. Alle disse tidsafhængige challenge værdier har en ting til fælles nemlig, at de kun bruges én gang, så man kan afgøre om de modtagne meddelelse er replay eller ej. Det ovenstående replay attack kan undgås, hvis man bruger Challenge/Response teknikken for hver sendte meddelelse, da man altid benytter en ny tidsafhængig challenge værdi for hver meddelelse, kan man så udfra protokollen se, at angriberen har brugt en meddelelse som kommer fra en foregående "samtale"/session. Altså meddelelsen er en gammel meddelelsen og bliver ikke accepteret.

3.3.1.2 Sekvensnumre (Sequence Numbers)

Et Sekvensnummer[92] er et unikt nummer, som kan bruges som en tidsvarierende parameter. Sekvensnumre kan bruges til at identificere en meddelelse og til, at afgøre om en meddelelse er en replay meddelelse eller ej. Sekvensnumre bruges ligesom tilfældige tal eller nonce i Challenge/Response protokollen. Et sekvensnummer er specifikt knyttet til et specielt par af entiteter og skal enten være eksplicit eller implicit tilknyttet til både afsender og modtager af en meddelelse.
En simpel sekvensnummer regel er, at et sekvensnummer starter med nul, stiger i rækkefølge og hver efterfølgende meddelelsessekvensnummer er større end den foregående modtaget meddelelses sekvensnummer. Et Sekvensnummer bruges ved, at vedhæfte sekvensnummeret med en meddelelse vha. kryptering og derefter sende denne meddelelse til modtageren. En sådan afsendt meddelelse accepteres af modtageren kun og kun hvis:

- Sekvensnummeret tilfredsstiller de ovenstående sekvensnummer regler, altså at et sekvensnummer starter med nul og stiger i rækkefølge
- Sekvensnummeret i meddelelsen er ikke blevet brugt før, enten i alle foregående meddelelser eller i et specificeret tidsvindue

Der findes et par ulemper ved sekvensnumre, da brugen af disse kræver følgende:

- Modtagere af meddelelser skal gemme de modtagne sekvensnumre i lange perioder, så det kan afgøres, om de brugte sekvensnumre i nye meddelelser er gyldige og man dermed kan afgøre om de modtagne meddelelserne er replay meddelelser eller ej
- Speciale procedure f.eks. resynkronisering af sekvensnummer kan være nødvendigt, hvis der sker forhold som f.eks. systemfejl der gør, at en enhed sender ét sekvensnummer afsted, mens modtageren forventer et andet
- Generelt er ”tvungne” forsinkelser meget svære at opdage ved brug af sekvensnumre, og kræver en protokol med mindst to udvekslinger
- Som en overordnet konsekvens, hvor synkronisering er nødvendig, er sekvensnumre mest passende for små og lukkede netværk

[92] Kilde: Handbook of applied cryptography, A. Menezes, P. van Oorschot, S. Vanstone, CRC Press, 1997

3.3.1.3 Tidskodet (Time Stamps)

Den sidste metode til at modvirke replay attacks, er ved at gøre brug af Time Stamps[93]. Man kan, vha. de modtagne Time Stamps i meddelelserne, forsyne en klok eller tidstabel (for sammenligning af de optagne meddelelser med Time Stamps), som bruges til at afgøre, om en modtaget meddelelse er en replay meddelelse. Dermed kan man forhindre de mulige replay attacks i ens kommunikation. Time Stamps kan også bruges til, at opdage de tvungne forsinkelser i systemet, som f.eks. kan forekomme ved systemfejl. En anden applikation for Time Stamps er, at tidsbegrænse adgang til system eller privilegier.

Time Stamps virker ved;:

Figure 59: Time Stamp

Der sker det, at Alice sender en enkelt meddelelse til Bob, indeholdende Alice's identitet, en ***CMD*** kommando og $\{T \| B\}_K$, hvor ***T*** er et Time Stamp sammensat med ***B*** (Bob's identitet), og derefter er krypteret med nøglen ***K***. Bob dekrypterer meddelelsen med nøglen ***K*** og kontrollerer, om meddelelsen er ny ved at sammenligne det nye ***T*** med de andre modtagne ***T´er.*** Alice og Bob er de eneste der kender nøglen ***K***.

Der sker generelt følgende, når en afsender gerne vil sende en meddelelse, så genererer systemet først et Time Stamp (hvis det er en host/server kommunikation, så genereres tiden fra hostens lokale ur). Dette Time Stamp kan vha. kryptering adderes til en meddelelse. Modtageren åbner denne Time Stamped meddelelse vha. dekryptering, kontrollerer sin egen lokal tid og finder forskellen mellem den sendte tid og den modtagne tid af meddelelsen.

[93] Kilde: Handbook of applied chryptography, A. Menezes, P. van Oorschot, S. Vanstone, CRC Press, 1997

Meddelelsen accepteres hvis og kun hvis:

- Time Stamps forskellen er indenfor de acceptable tidsvindue, som generelt er på 10-20 millisekunder[94]

- Ingen meddelelse med samme Time Stamp er modtaget fra samme afsender. Dette kan der holdes styr på vha., at modtageren gemmer alle de/den senest modtagne Time Stamps i en tabel, og ved hver ny meddelelse sammenlignes det nye Time Stamp med det i tabellen

Sikkerheden af Time Stamp baseret verifikation ligger ved brug af samme tidsreference. Dette kræver, at alle ure er tilgængelige, er synkroniseret samt, at de er sikret mod modifikation[95]. Synkronisering er nødvendig for Time Stamp metoden, da man sammenligner tidsværdierne ved afsendelsen og modtagelsen, og meddelelsen accepteres kun i et lille tidsvindue. Hvis de to kommunikerende parters ure ikke er synkroniseret, dvs. der er en større tidsforskel på deres ure, så er der stor risiko for, at meddelelserne bliver opfattet som replay meddelelser og bliver forkastet. Utilgængelighed for modifikation af meddelelsen er også vigtigt i Time Stamp metoden, da alle som har adgang til kommunikationskanalen kan forfalske et Time Stamp i en meddelelse og dermed snyde de to kommunikerende parter. Dette kan sikres ved at bruge kryptering.

[94] Kapitel 10 i Handbook of applied cryptography
[95] Se kapitel 2.3.1.4 Angrebsmetoder.

3.3.2 Løsningsforeslag for at undgå/modvirke replay attacks

Som beskrevet, så findes der flere muligheder for, at forhindre replay attacks kan gøre skade og disse er blevet beskrevet i de foregående afsnit. Hver af disse løsninger blev beskrevet mht., hvordan den tidsvarierende parameter (Time Stamp, Sequence number, nonce) bliver genereret og, hvordan de bliver kontrolleret ved meddelelsens modtagelse. I det følgende beskrives de tre løsninger i et Z-Wave netværk.

Challenge/Response

Indledningsvis beskrives kort, hvordan Challenge/Response metoden virker i Z-Wave netværket. Kontrolleren genererer et tilfældigt tal ***R*** (Challenge) og sender det til noden, som skal returnere med et response $(R||T)_K$, hvor ***T*** er meddelelsen. $(R||T)_K$ betyder ***R*** sammensat med ***T*** og krypteret med nøglen ***K***. Hele idéen er, at en angriber ikke kan replay tidligere meddelelser, da kontrolleren vælger et nyt ***R***, hver gang der skal sendes en ny meddelelse.

Denne løsning er sikker men sandsynligvis ikke realistisk i Z-Wave netværket, da løsningen indebærer et stort overhead, da man skal bruge hele 3 ekstra meddelelser for at ens meddelelse accepteres. Yderligere medfører udvekslingen af beskeder mellem kontroller og noder ekstra kommunikations- og behandlingstid.

Sequence Numbers

Afsenderen skal for hver modtager, holde øje med et sekvensnummer, som er indføjet som ***R*** og stiger med hver sendte meddelelse. Modtageren skal huske for hver afsender, hvilket sekvensnummer sidst er modtaget og accepterer kun et nyt ***R***, hvis og kun hvis det er én større end den gemte værdi. Dette forhindrer replay attacks, eftersom angriberen kun har meddelelser med værdier af ***R,*** som er for lille. Hvis vi antager, at meddelelserne under normale omstændigheder genereres i den rækkefølge de bliver sendt, så vil gyldige meddelelser (altså ikke replay meddelelser) ikke blive afvist.

Inkluderingsprocessen i Z-Wave antages at være god. Ved fornyelsen af nøgler i Z-Wave netværker, fornyer man også sekvensnummeret, dvs. sekvensnummer starter fra nul. Når en ny node er blevet inkluderet i netværket (eller muligvis i tilfælde af fejl), skal sekvensnummeret synkroniseres, dette gøres ved at sætte sekvensnummeret til nul.

Sekvensnummer afbrydning giver problemer, derfor skal sekvensnummer synkroniseres ved afbrydning. Dette kan finde sted ved følgende:
Ved kommunikation mellem kontroller og node, ”opdager” noden at sekvensnummeret ikke passer med det forventede, og giver besked tilbage til kontrolleren om dette. Kontrolleren sender et nyt sekvensnummer, med et tilfældigt tidsvarierende komponent ***R*** tilbage til noden**.** Noden returnerer en autentisk meddelelse indeholdende ***R*** sammen med det aktuelle sekvensnummer i payloaden. Noden har nu accepteret svaret og tilpasser dets aktuelle sekvensnummer, det samme gælder for kontrolleren, hvis og kun hvis svaret er rigtigt autentificeret indeholdende det korrekte ***R***. Her skal kontrolleren skal have en liste over de brugte ***R´er***, for sammenligning og for at et ***R*** ikke bruges mere end en gang. Dette forhindrer replay attacks af tidligere synkroniserede meddelelser.

Brug af sekvensnummer kan medføre meget administration, da hver node skal gemme en tabel af sekvensnumre for alle andre noder, som den kommunikerer med. Dette kræver et højt hukommelsesforbrug.

Time Stamps

Vi starter med at give en beskrivelse af Time Stamp løsninger i Z-Wave og giver derefter forslag og anbefalinger for, hvordan de kan bruges i Z-Wave netværket. I et Time Stamp system sættes ***R*** til at være den aktuelle systemtid og et modtaget ***R*** kan kun accepteres af modtageren, hvis og kun hvis det modtagne ***R*** ikke ligger alt for langt væk fra nodens egen aktuelle tid, efter tilføjelsen af en forventet transporttid, som forudsætter at den kan beregnes.

Det vil ikke fungere, hvis senderens og modtagerens ure er for langt væk fra hinanden: Givet at modtageren kun accepterer ***R***’er inden for et bestemt tidsinterval omkring dens aktuelle tid, angriberen vil kun have meddelelser tilgængelige med forældede værdier af ***R*** medmindre han

handler hurtigt. Hvis der er god synkronisering af urene og transporttiden af meddelelser kan blive forudsagt nogenlunde præcist, så kan replay attacks undgås.

Generelt bliver løsning med Time Stamp nød til at leve med en opvejning mellem nødvendigheden af at acceptere gode meddelelser på den ene side og nødvendigheden for at afvise dårlige meddelelser på den anden side. Det rigtige valg af et tidsvindue hvor en meddelelse accepteres, kan kun blive lavet med præcis viden, om applikationen (hvis vi f.eks. kigger på det eksempel om at Alice glemmer sin mobiltelefon i huset efter hun tilslutter alarmen, hvor lang tid skal der så gå før Alice kan afbryde alarmen igen og hente sin mobiltelefon), om hvor god synkroniseringen er, og hvor præcis vi kan forudsige tiden for en meddelelse tager for at ankomme (transporttiden). Fordelen ved denne løsning er, at den ikke har brug for ekstra hukommelse eller ekstra udveksling af beskeder.

Generelt set, er replay af meddelelser muligt indenfor det valgte tidsvindue, og umuligt udenfor dette tidsvindue, da modtageren kun accepterer en meddelelse inden for dette tidsvindue. Man skal være opmærksom på, at replay af meddelelser inden for tidsvinduet ikke behøver at være et seriøst problem, det afhænger helt og holdent af applikationen. Med hensyn til Z-wave bliver følgende anbefalet:

Formatet af Time Stamp'et skal være følgende:

- Først en ur-værdi som starter med den byte, som veksler hurtigst
- Efterfulgt med afsenderens node id og modtagerens node id

Hvis resultatet er kortere end 8 bytes, så tilpasses resultatet til 8 bytes – foretrækkende med tilfældige bytes, og hvis det ikke er gennemførlig, brug et tilfældigt fast byte mønster.

Transporttiden i Z-Wave kan være op til 30 sekunder, men den kan også være meget kortere. I dårlige tilfælde holder dette for meddelelser, som bruges til at synkronisere tiden. Derfor kan en god meddelelse som skal accepteres have et time stamp, som er op til et minut væk fra afsenderens lokal tid. Desuden er der ikke en fuldstændig pålidelig metode til at forudsige transporttiden, så man bliver nødt til at acceptere en sådan forskel på 1 minut.

I værste tilfælde bliver det svært at regne/bestemme transporttiden, dels pga. hvor stor netværket er, dels pga. hvor meget information der skal transporteres og dels pga. man ikke kan vide hvor meget netværket er belastet.

I bedste tilfælde kan transporttiden regnes ud fra:

> Protokollen vil prøver op til fire forskellige ruter, i værste tilfælde for alle ruter bruger fire hops til base-stationen, se figuren neden under:

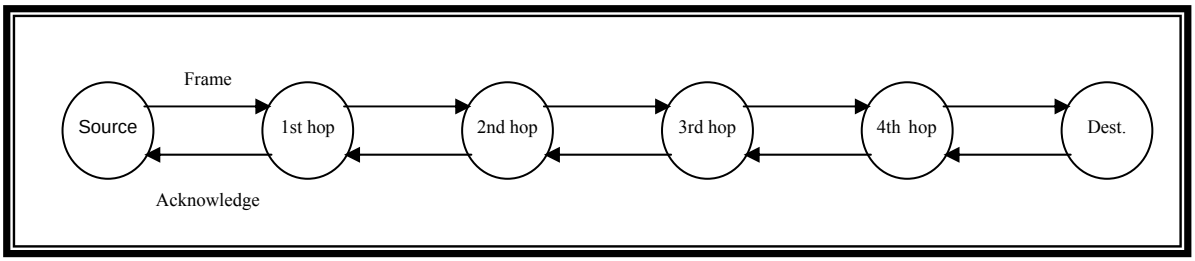


Figure 60: Bruger fire hop til at nå destination

Hvis direkte (ingen hop) kommunikationen fejler, så er protokollen programmeret til timeout efter 620 ms. Derfor kan transport tiden udregnes som:

$$T_{Delay} = N_{Number\ of\ routes} * (N_{Number\ of\ hops} + 1) * T_{0\ hop\ timeout}$$

$$T_{Delay} = 4 * (4 + 1) * 620\ ms = 12400\ ms \sim 15\ seconds$$

Et ur ”taber eller vinder” tid efterhånden som tiden går, men et tidsvindue på 1 minut vil normalt være ’meget’ stort i forhold til dette. Man kan evt. tænke sig, at det er nødvendigt, at synkronisere urene en gang om måneden. Hvis det er nødvendigt at synkronisere hyppigere, kunne man vælge i stedet at forøge værdien af tidsvinduet. Hvorvidt sådan et tidsvindue er tilfredsstillende eller ej, afhænger i høj grad af anvendelsen. Hvis et stort tidsvindue er nødvendigt, må man fraråde brugen af Toggle kommandoer, da disse ellers vil være et oplagt angrebsmål.

Synkronisering af Tid

Synkronisering kan ske ved, at nye noder synkroniserer med kontrolleren lige efter de er blevet inkluderet i netværket. Synkroniseringen af tiden skal også finde sted ved strømsvigt, hvis dette medfører at noden taber tiden. Periodisk synkronisering af tiden er også nødvendigt, da urene i de forskellige noder vil tabe/vinde forskelligt. En god mulig begivenhed vil være en månedlig nøgle fornyelse, hvis dette interval er passende. Synkronisering mellem to noder kan også foregå, hvis en node modtager en besked med forkert time stamp. Dette kan så foretages ved en Challenge/Response protokol, som også indeholder en ny værdi for uret.

Synkronisering af tid kan gøres ved følgende:

1. Noden sender en pakke til kontrolleren, der i payloaden har en tilfældig 8 byte block ***R,*** og efterspørger kontrollerens aktuelle tid
2. Kontrolleren returnerer en autentisk pakke, som indeholder ***R*** som tidsvarierende komponent og den aktuelle tid i payloaden
3. Noden accepterer svaret og justerer sin aktuelle tid, hvis og kun hvis svaret er korrekt autentificeret og indeholder det korrekte ***R***. Dette forhindre replay attacks af tidligere synkroniseringsmeddelelser.

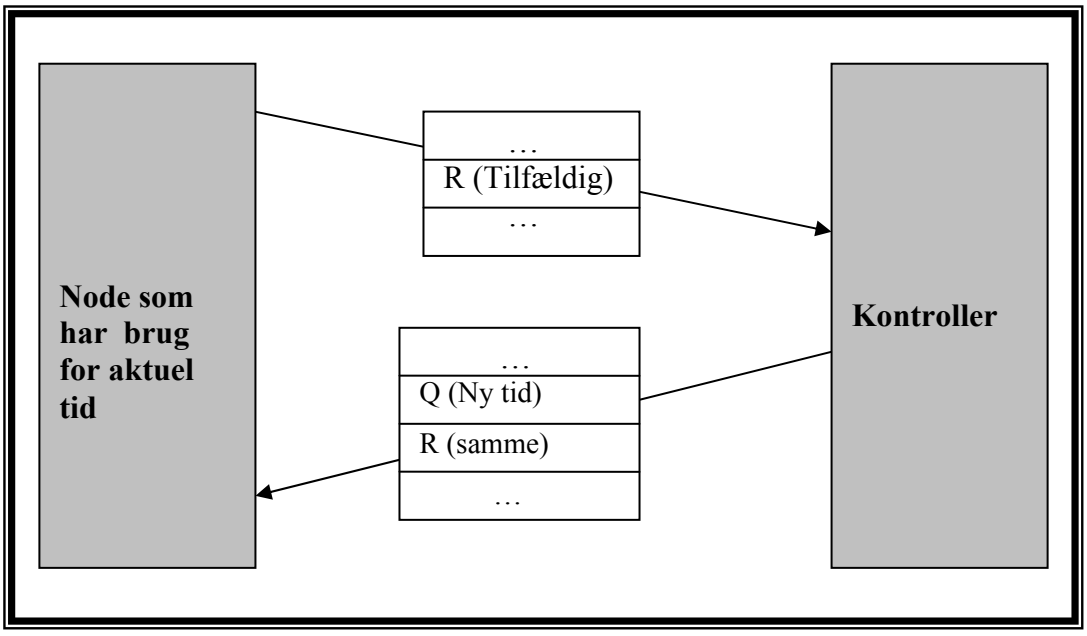


Figure 61: Tidssynkronisering af node

Bemærk, at det betyder at noden skal huske ***R*** fra trin 1 til trin 3, og at det her kræver en speciel kode til at kontrollere den tidsvarierende komponent. Den skal sammenlignes med den gemte ***R***-værdi og ikke behandles som et Time Stamp.

Inkludering af ny node

En ny sikker node bliver inkluderet i netværket ved at gå i learning mode, hvor den vil acceptere nye nøgler. Dette sker ved følgende:

Noden sender først en meddelelse gennem en out-of-band kanal til en bærbar kontroller. Denne sender en meddelelse tilbage, som indeholder Home- og Node ID til den nye node samt et nøglesæt KS, som består af en krypteret nøgle og en autentificeret nøgle. Nøglesættet KS er tilfældig genereret til noden, imens den er i learning mode. Bagefter gemmer noden nøglesættet KS. Vi antager, at i dette øjeblik, hvor noden bliver inkluderet i netværket, kan noden kommunikere trådløst med kontrolleren. Midlertidig vil noden ignorere al kommunikation indtil kontrolleren sender den en speciel pakke, som vil blive beskrevet senere.

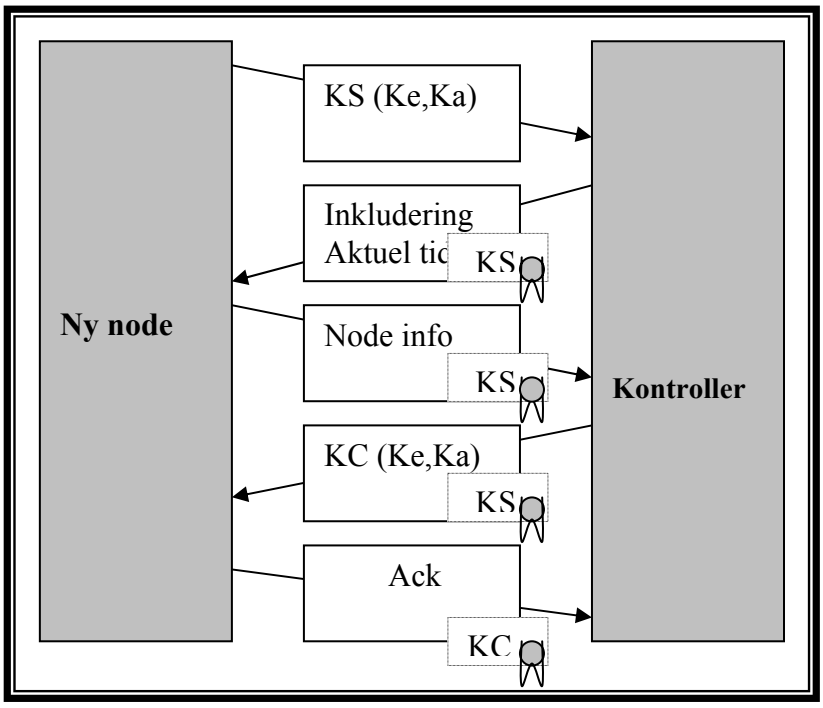


Figure 62: Procedure ved inkludering af ny node

1. Kontrolleren sender en speciel ”join” pakke til den nye node. Denne pakke er autentificeret med nøglesættet KS. Noden vil kun svare tilbage, hvis sådan en pakke er modtaget.
2. Noden svarer tilbage med en pakke, som indeholder dens identitet og er autentificeret ved brug af nøglesættet KS.
3. Kontrolleren kan nu vise på sit display identiteten af den nye node til installatøren, derved kan installatøren kontrollere, om noden som er ved at blive inkluderet har et ID, der passer

med den node han fysisk har installeret. Hvis installatøren accepterer, sender kontrolleren en ny pakke, som indeholder det aktuelle nøglesæt KC til netværket og sit versionsnummer. Pakken er både krypteret og autentificeret med nøglesættet KS.

4. Efter modtagelsen af nøglesættet KC, erstatter noden nøglesættet KS med det modtagne nøglesæt KC. Noden sender en bekræftelsespakke til kontrolleren, som er autentificeret med nøglesættet KC.
5. Dette tillader kontrolleren at fortælle installatøren, at inkluderingen af den nye node var vellykket.
6. Den nye node skal nu synkronisere tiden med nøglesættet KC. Dette er blevet beskrevet tidligere.

Bemærk at punkt 1 - 5 sker før nodens tid er synkroniseret. Derfor kan et Time Stamp ikke blive genereret eller bruges for disse meddelelser. Dette er ikke et sikkerhedsproblem, da en node kun bliver inkluderet en gang i sit liv, så der er intet at vinde ved replaying af sådanne meddelelser.

Det skal bemærkes at fasen, hvor den nye node kommunikerer med kontrolleren er den mest sårbar del af designet. Hvis denne kommunikation og den følgende meddelelse fra kontrolleren er blevet aflyttet, bliver nøglesættet til hele netværket kompromitteret. Alligevel, ved givet det krav, at vi kun har ét nøglesæt for hele netværket, og at noden ikke kan blive preinstalleret med nøglerne, er dette problem uundgåelig. Den eneste løsning til dette problem er, at holde fasen kort og at gøre aflytning fysisk svært.

Dialogen, hvor den nye node sender sit ID til kontrolleren før den får nøglesættet KC, vil forhindre angriberen i at installere en anden enhed på netværket, uden at installatøren opdager det. Beskyttelsen af denne dialog er sikker, så længe angriberen bruger en eksisterende enhed uden ændring. Selvfølgelig kan protokollen ikke forhindre en enhed fra at lyve om sin identitet, men sådan et angreb er svært at forhindre, da enheden kan finde på en identitet, som kan se overbevisende ud for installatøren.

Bemærk, hvis kontrolleren fejler og en ny kontroller skal tilsluttes til netværket, kan den ovennævnte procedure ikke bruges, da den antager at kontrolleren fungerer perfekt. Dette kan løses, enten ved at have en backup kontroller, som er tilsluttet netværket samtidig med den primære

kontroller, eller ved at tilslutte kontrolleren i en direkte forbindelse med nøglesæt KC, så den kan lære det aktuelle nøglesæt.

Kombination af to metoder for at modvirke replay attacks

I foregående afsnit er der givet en beskrivelse af, hvordan de tre metoder, nemlig Challenge/Response, Sequensnumbers og Time Stamps virker for at forhindre replay attacks. I dette afsnit kigger vi på en løsning, som kan fås ved at kombinere to af disse løsningsmetoder, nemlig Time Stamps og Challenge/Response. Man kan nemlig benytte de to metoder på engang. Man kan starte med først at bruge Time Stamp løsningen i et lille tidsvindue på f.eks. X sekunder (kan være ¼ af hele tidsvinduet), og derefter kan man benytte Challenge/Response løsningen i det resterende acceptable tidsvindue i Z-Wave netværket (se figuren nedenunder). Det acceptable tidsvindue sættes til Y sekunder.

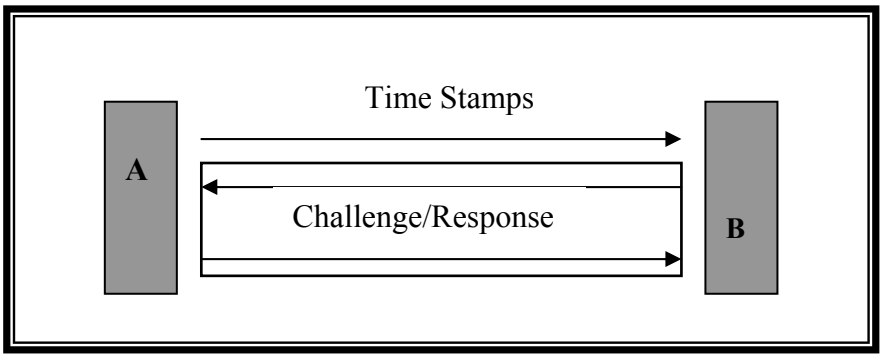


Figure 63: Kombination af de to løsningsforslag

A starter med at sende en meddelelse med Time Stamp til ***B***. ***B*** accepterer meddelelsen hvis og kun hvis den modtagne meddelelsen med Time Stamp ligger indenfor det acceptable tidsvindue, altså X sekunder. Hvis meddelelsen ikke ligger indenfor det acceptable tidsvindue, så bruges Challenge/Response[96] metoden. Dvs. ***B*** sender en Challenge til ***A*** og vil gerne have et response tilbage. ***B*** accepterer meddelelsen hvis og kun hvis det response, der fås tilbage, er rigtig samt at det er inden for det resterende tidsvindue, altså indenfor Y sekunder.

Meddelelsen med Time Stamp bliver ikke accepteret enten pga. forkert Time Stamp (en meddelelse fra en angriber) eller pga. meddelelsen er udenfor det acceptable tidsvindue pga. stor transporttid). Challenge/Response metoden bruges til, at være sikker på at den der kommunikeres med, er den som den giver sig ud for at være, så man kan afgøre om meddelelsen med Time Stamp, man har

[96] Se afsnit 3.4.1.1 Challenge/Response

modtaget, ikke kommer fra en angriber. En angriber kan nemlig ikke svare rigtig til challengen fra ***B***, da det er kun er de to kommunikerende parter ***A*** og ***B,*** som kender nøglen.

Denne metode er meget sikker, da man begrænser det totale tidsvindue ned til X sekunder (1/4 af det totale tidsvindue), hvilket bliver svært for angribere at opsnappe en meddelelse og sende den videre igen på X sekunder. Det er også svært for angriberen at spille ***B*** over for ***A*** i det acceptable tidsvindue, fordi ***A*** vil ikke accepterer angriberen, da han ikke kender angriberen og hans Challenge.

En fordeling til denne løsning kan ses på figuren neden under, som viser fordelingen mellem Time Stamps og Challenge/Response. Figuren viser, at 80% af de Time Stamped meddelelser, der bliver sendt, accepteres og de resterende Time Stamped meddelelser som ikke bliver accepteret, bliver accepteret efter brug af Challenge/Response metoden.

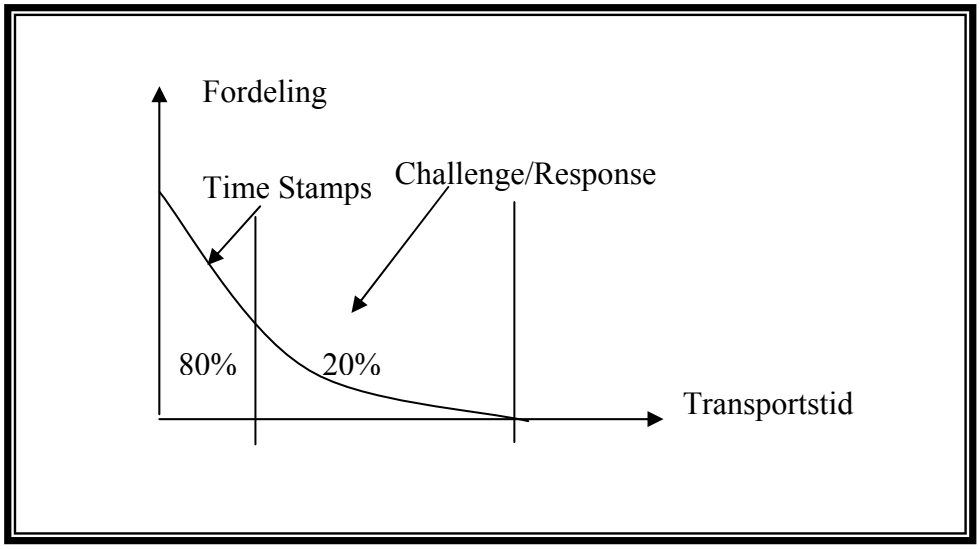


Figure 64: Løsningstabel mod Replay attacks

3.3.3 Tildeling af ny netværksnøgle

Nøglerne i Z-Wave netværket skal skiftes, så netværket hele tiden forbliver sikkert. For at bestemme hvor tit nøglerne i Z-Wave netværket skal skiftes, kigges der på Unicity distance. Unicity distance n_o er en tilnærmelse til den mængde af ciffertekster, så summen af den ægte information (Entropi, bruges til at bestemme hvor dårlig en nøgle distribution er. Defineret som, summen af -pK $\log_2(pK)$, hvor pK er sandsynligheden af nøgle K) i tilsvarende ciffertekst og krypteret nøgle, er lig nummeret af de brugte ciffertekst bits. Altså ved et perfekt sikkert ciffer er $n_o = \infty$.

Unicity distance giver en probabilistisk (dvs. at resultatet ikke er 100% sikkert) resultat, nemlig minimum tilnærmelsen af ciffertekster, for hvilket det er sandsynligt, at der kun er én eneste forståelig klartekst svarende til cifferteksten, når alle mulige nøgler prøves ved dekryptering.
Unicity distance virker ved, at man skal have alle de mulige ciffertekst karaktere. Dvs. hvis unicity distancen af et ciffertekst består af X karakter, og hvis man opsnapper mindre end X ciffertekst karaktere, så er der ikke nok informationer til, at adskille den rigtige nøgle fra et sæt af mulige nøgler.

I Z-Wave kender man hele cifferteksten og noget af klartekstens indhold. Meddelelserne i Z-Wave netværk er korte og det fører til, at flere nøgler ser ud som om de passer, hvilket giver en høj Unicity distance.

Grunden til at vi kigger på unicity distance er, at 3DES er hardware implementeret i Z-Wave. DES har unicity distance på 8.2 bytes[97], dvs. hvis man bruger brute-force angreb på DES så har man brug for to ciffertekst blokke, da DES' bloklængde er på 8 bytes. Man dekrypterer den første ciffertekst blok med en nøgle og hvis den fundne klartekst ligner et engelsk ord/noget fra Z-Wave protokollen, så dekrypterer man den anden ciffertekstblok med samme nøgle. Ligner denne også ligner et engelsk ord/noget fra Z-Wave protokollen, har man fundet den rigtige nøgle. Er det ikke tilfældet prøver man med en ny nøgle. For en 128-bit ciffer er unicity distancen på 19 bytes.

[97] Kilde: http://www.packetstormsecurity.org/mag/crypto-gram/crypto-gram-9812.html

I 1996 har man vist hvor langt tid det tager for at knække DES, se figuren[98] nedenunder:

Investment	*Time to Break*
$10,000	18 months
$300,000	19 days
$300,000,000	12 seconds

Table 32: Viser hvor lang tid det tager for at knække DES

Man kan se i tabellen, at ved at investere nogle millioner dollars kan knække DES på 12 sekunder, men når det gælder 3DES, så kommer man nok til at bruge mindst tre gange så meget tid (grov sagt, da man også skal tage hensyn til nøglen, som vokser fra 56 til 112), dvs. 3*12 sekunder = 36 sekunder til at knække nøglen. Hvis vi antager, at man vil bruge 10.000$ til at knække nøglen i Z-Wave netværket, så tager det mindst 3*18 = 54 måneder eller 4,5 år. Hvis man bruger 300.000$ så tager det 3*19 dage = 57 dage = ca. 2 måneder.

Det er usandsynligt at nogen vil bruge millioner eller tusind af dollars til at knække nøglen i et Z-Wave netværket, da her er kun tale om et smart hus netværk. Og hvis vi antager dette, så ser netværket ud til at være ret sikker, dvs. man behøver ikke at skifte nøglerne i netværket ret tit. Hvis vi antager at nogen prøver at knække nøglen i netværket med de simpleste værktøj, så går vi udfra at personen bruger mindst et par år. Men for en sikkerheds skyld så anbefaler vi at man skifter netværks nøglerne så tit som muligt, dermed gør man endnu mere besværligt for angribere at knække nøglerne i netværket.

Det koster altså ikke noget at skifte nøglerne i netværket, man kommer kun til at bruge tid på det og ikke andet. Men der er også en ulempe ved skiftning af nøgler, nemlig at man belaster netværket. Hvis vi tager hensyn til disse tre ting så ville vi anbefale at man skifter nøglerne en gang om måneden. Men der er ikke nogen risiko, hvis man ikke skifter nøglerne hver måned. Alt efter alt er det personens egen interesse og ønske hvor tit han vil skifte nøglerne i netværket, men i værste tilfælde vil det være en god ide at skifte dem mindst en gang om året.

[98] Kilde: www.sans.org/rr/papers/20/726.pdf

Som beskrevet ovenfor, så er det nødvendigt at skifte netværksnøglen for at sikre at netværket stadigvæk er sikkert. Hvordan denne nøgleudveksling kan foregå, samt hvilke fordele og ulemper der findes, gennemgås nedenfor. For at kunne kontrollere netværket bedst muligt antager vi, at det kun er den primære kontroller, som kan igangsætte nøgleudvekslingen. Den primære kontroller står for nøgleudvekslingen, så der ikke kan være nogen tvivl om, hvem der er blevet opdateret samt hvilken nøgle disse har fået. Vi tager desuden udgangspunkt i nedenstående store scenarium:

Netværket består af:

- *Én kontroller*
- *232 noder:*
    - *32 noder nås direkte fra kontrolleren (succestid: 50 ms)*
    - *50 noder nås via 1 hop (succestid: 200 ms)*
    - *50 noder nås via 2 hop (succestid: 310 ms)*
    - *50 noder nås via 3 hop (succestid: 420 ms)*
    - *50 noder nås via 4 hop (succestid: 535 ms)*

Vi forudsætter, at der kommunikeres succesfuldt med alle noder første gang, dvs. der skal ikke retransmitteres og der sker ingen ekstra routing. Vi kan udrenge den maksimale tid, for hvor lang tid det har taget for kontrolleren at få kommunikeret med alle noder i netværket:

Samlet tid = 32 * 50 ms + 50 * (200 ms + 310 ms + 420 ms + 535 ms) = 74,850 sekunder.

Vi forudsætter desuden, at selvom en node har fået en ny nøgle, så beholder den stadigvæk den gamle i en time. Dette gøres for at sikre, at noderne kan kommunikere sammen, også i et worst case scenarium.

3.3.3.1 Nutidig nøgleudskiftning

Når nøgleudskiftningen skal finde sted, sender kontroller den nye nøgle til de individuelle noder, én ad gangen (singlecast). Noderne bruger den nye nøgle med det samme og ser vi på belastningen af netværket, må denne løsning siges at være effektiv. Det eneste kontrolleren sender, er den nye nøgle (samt en checksum). Når noden modtager den nye nøgle, kontrolleres det om checksummen passer, og er det tilfældet svares der tilbage med er en Acknowledge (krypteret med den nye nøgle). Passer checksummen ikke, svarer noden ikke tilbage. Dette vil få kontrolleren til igen at sende den nye nøgle, dog højest tre gange i alt. Lykkes det ikke efter de tre gange, må kontrolleren prøve at sende til noden via nogle andre noder (routing). De routende noder **skal** bruge den nye nøgle, så de forstår hvad kontrolleren sender til dem. Dette er ikke noget problem, da kontrolleren ved hvilke noder der har fået den nye nøgle. Ovenstående antyder at den smarteste og mest effektive fremgangsmåde er, først at udskifte nøglen hos de noder der kan nås direkte, derefter dem som kan nås ved 1 hop, så dem med 2 hop osv.

Ulempen ved denne fremgangsmåde vokser med netværkets størrelse, fordi jo større netværket bliver, des længere tid tager det for kontrolleren at få opdateret alle noderne. Problemet består i, at vi kan komme ud for at to noder bruger forskellige nøgler og derfor ikke lige umiddelbart kan kommunikere med hinanden. For at belyse dette nærmere opstilles et scenarium og nogle løsningsforslag gennemgås:

Vi tager udgangspunkt i et netværk, som består af adskillige noder samt en kontroller. Vi vælger at se på to af noderne og kalder dem 1 og 2. Kontrolleren står for at uddele nye nøgler, node 1 har fået den nye nøgle og bruger denne, mens node 2 stadigvæk bruger den gamle. Node 1 prøver at kommunikere med node 2 vha. af den nye nøgle, men får ikke noget svar tilbage. Følgende fremgangsmåder kan bruges for, at noderne fortsat kan kommunikere med hinanden:

1. Node 1 får ikke noget svar tilbage fra node 2 (tre forsøg), så det kan være at denne stadigvæk bruger den gamle nøgle. Derfor forsøges der igen at sende til node B, men nu med den gamle nøgle. Kommer der stadigvæk ikke noget svar tilbage, er der en mulighed for, at node 2 i mellemtiden har fået den nye nøgle, så der sendes igen med den nye nøgle.

Kommer der stadigvæk ikke noget svar fra node 2, er det et problem som ikke kan relateres til selve nøgleudvekslingen.

2. Da node 1 ikke får noget svar tilbage fra node 2 (tre forsøg), kan node 1 antage at node 2 ikke bruger den nye nøgle, men er ikke sikker. Derfor kontaktes kontrolleren med en forespørgelse om, hvilken nøgle node 2 bruger. Svarer kontrolleren tilbage at node 2 stadigvæk bruger den gamle nøgle, kan node 1 kommunikere med node 2 med den gamle nøgle. Hvis kontrolleren svarer tilbage, at den har skiftet over til den nye nøgle, kan dette enten lige være sket eller node 2 er bare ikke til at få fat i. For at finde ud af dette, prøver node 1 igen med den nye nøgle.

3.3.3.2 Nøgleudskiftning med aktivering

Kontrolleren sender en ny nøgle til de individuelle noder, én ad gangen (singlecast). Noderne bruger ikke den nye nøgle med det samme, men venter på besked fra kontroller. Når alle har fået den nye nøgle, sender kontrolleren en aktiveringsbesked til noderne om, at de skal bruge den nye nøgle. Dette gøres først med en broadcast, og bliver derefter fulgt op med singlecasts, så det er sikkert at alle skifter.

Fordelen ved denne løsning frem for den ovenfor beskrevet er, at tidsrummet hvor nogle noder bruger den gamle nøgle, mens andre bruger den nye, er mindre. Grunden til dette er, at en aktiveringspakken ikke fylder lige så meget som selve pakken med den nye nøgle, så derfor sendes den hurtigere frem og tilbage i netværket. Vil man yderligere mindske dette tidsrum, kan man overveje om nogle af noderne skal videresende broadcast pakken med aktiveringen.

Dette kan gøres ved at i selve broadcast pakken også findes en sandsynlighedsværdi, som noderne bruger til at finde ud af om de skal sende pakken videre eller ej. Har noderne først en gang taget stilling til om videresendelsen skal finde sted eller ej, må den ikke tage stilling til dette igen ved denne nøgleudveksling, selvom den en gang til skulle modtage en aktiveringsbroadcast. Sandsynlighedsværdien kan kontrolleren f.eks. fastsætte ud fra størrelsen af netværket, men det skal advares mod at bruge en for høj værdi, så netværket ikke blokeres fordi alle noder sender broadcast pakker.

Aktivering ved først at bruge broadcast bevirker, at mange noder skifter over til den nye nøgle med det samme. Dog kan vi ikke garantere at alle noder har hørt denne broadcast, så kontrolleren må følge op med singlecast til hver enkelt node for, at sikre at alle noder har skiftet over.

Ser vi på belastningen af netværket, når denne procedure for nøgleudveksling benyttes, må det konstateres at den er større end forrige løsning, da alle noderne først får tilsendt nøglen (via singlecast) og derefter får en aktiveringsbesked (broadcast og singlecast). Belastningen forøges yderligere, hvis noderne også skal sende broadcast pakker ud. De to forslag gennemgået i forrige afsnit omhandlende, hvordan noder med forskellige nøgler kan kommunikere, kan også bruges her uden ændringer.

3.3.3.3 Tidsstyret nøgleudskiftning

En anden fremgangsmåde, er at kontrolleren sender en ny nøgle til de individuelle noder, én ad gangen (singlecast). Sammen med nøglen sendes også et aktiveringstidspunkt. Noderne skifter nøgle på det angivne tidspunkt, eller næste gang de vågner op efter dette tidspunkt.

Fordelen ved denne løsning, er at nøgleudskiftningen kan planlægges lang tid i forvejen, så kontrolleren i løbet af en dag får kontaktet alle noder om nøgleskiftet. Derved får vi spredt belastningen af netværket ud på et stort tidsrum, som (belastningen) desuden må betegnes at være minimal, da noderne kun skal kontaktes én gang.

Bruges denne fremgangsmåde kræves det, at urene på noderne går ens/har samme tid. Jo mindre forskel, des mindre risiko er der for, at noderne bruger forskellige nøgler. En protokol som bliver brugt på Internettet er NTP[99], hvor der sendes adskillige pakker frem og tilbage for at synkronisere tiden. Nøjagtigheden er for 90% på under 100ms[100]. En sådan nøjagtighed i et netværk er altid godt at have, men absolut ikke et krav. For at kompensere for unøjagtigheden af urene, kunne man evt. indføre en ”stillezone” efter skiftet. I dette tidsrum, er det ikke tilladt for noderne at kommunikere. Skiftet af nøgler sker på et tidspunkt, som nævnt er bestemt af kontrolleren. Det anbefales, at kontrolleren overvåger netværket for at finde et tidspunkt, hvor der statistisk set, ikke er meget trafik. Det er med denne løsning ikke nødvendigt for noderne, at gemme den gamle nøgle som ved de tidligere løsninger, da skiftet sker relativt hurtigt og samtidigt.

[99] Website: http://www.ntp.org/
[100] Kilde: http://www.ntp.org/ntpfaq/NTP-s-algo.htm#Q-ACCURATE-CLOCK

3.3.4 Opsummering af replay attacks

Ovenfor blev der gennemgået løsningsforslag til at undgå replay attacks mod Z-Wave netværket. Et replay attack mod Z-Wave kan undgås ved at benytte en af de løsninger vi foreslog, enten Challenge/Response teknikken, eller ved brug af Sekvensnummer, eller ved brug af et Time stamp. Man kan også benytte at bruge to af de løsninger på samtidig, f.eks. man kan benytte Time Stamps metoden først ved starten af det accepterede tidsvindue og ved resterende tid kan man benytte Challenge/Response teknikken. Denne løsningsmetode er også blevet beskrevet i design delen.

Challenge/Response teknikken er en god og sikker metode til at undgå replay attacks. Men i Z-wave netværk er Challenge/Response metoden ikke realistisk, dels pga. at den kræver for meget overheads for meddelelsen bliver accepteret og dels pga. udveksling af beskeder mellem noderne kræver ekstra kommunikations- og behandlingstid.

Som sagt kan man også benytte at bruge Sekvensnummer til at undgå replay attacks. Det sker ved, at man addere et Sekvensnummer i sendte meddelelser vha. kryptering, som dermed kan bruges til at afgøre om de modtagne meddelelser er replay meddelelser eller ej. Her skal der huskes at det brugte sekvensnummer skal stige med hver sendte meddelelse. Denne løsning kræver et højt hukommelsesforbrug, da hver node i netværket skal gemme en tabel af sekvensnumre for alle de andre noder, som den kommunikerer med.

Time Stamps kan også benyttes til at undgå replay attacks. Time Stamp bruges ved at man vha. kryptering addere et Time stamp i sendte meddelelser, så modtageren kan afgøre om de modtagne meddelelser er replay meddelelser eller ej. Ved brug af Time Stamp løsningen skal man være opmærksom på synkronisering af tiden og bestemmelse af transport tiden. Hvis der er god synkronisering af urene og hvis man kan forudsige transporttiden af meddelelserne, så er denne løsning god til at undgå replay attacks.

Der er også gennemgået et løsningsforslag, hvor man bruger Time Stamps løsningen og Challenge/Response teknikken samtidig. Dette forslag går ud på, at man først bruger Time Stamps løsningen i starten af et tidsvindue, og ved resterende tid af tidsvindue kan man bruge Challenge/Response løsningen. Man forventer at 80% af de sendte meddelelser med Time Stamp bliver accepteret, mens de resterende 20% accepteres efter brug af Challenge/Response. Denne

løsning er også god til at undgå replay attacks, da man bruger Time Stamps løsningen i første omgang og begrænser tidsvinduet helt ned til X sekunder (kan være ¼ af det totale tidsvindue). Men hvis meddelelserne ikke accepteres ved brug af Time Stamps løsningen, så skal man benytte Challenge/Response løsningen.
Denne løsning anbefaler vi til forhindring af Replay attacks mod Z-Wave. Denne løsninger begrænser tidsvinduet for en sendt meddelelse og er ret sikker til forhindring af Replay attacks. Da 80% af de sendte meddelelser med denne løsning accepteres vha. Time Stamps og de resterende 20% accepteres med Challenge/Response, så er der ikke behov for, for meget overhead.

Med hensyn til hvor tit nøglerne i Z-Wave netværket skal skiftes, er der kigget på Unicity distance, så der kunne bestemmes hvor lang tid det tager for at knække DES og 3DES. Vi har fundet frem til, at man ved at investere millioner dollars, kan knække DES på nogle sekunder. Men da her er tale om et almindelig Home Automation netværk, så er det usandsynligt at nogen kan finde på at bruge penge på at knække netværksnøglen. Hvis der bliver brugt almindelig værktøj til at knække DES, så tager det et par år. Men for 3DES kommer dette op på flere år. Derfor kan vi nemt konkludere at netværket er ret sikker og man behøver ikke at skifte netværksnøglerne tit. Man kan selv bestemme, hvor tit man skifter nøglerne i netværket, men vi anbefaler at man gør det en gang om måneden for sikkerhedsskyld, så man dermed begrænser angribere til at knække ens netværksnøgle. Man skal være opmærksom på at det ikke koster noget at skifte netværksnøglerne, ulemperne ved det er, at det kræver tid og man risikere at overbelaste netværket hvis man skifter nøglerne tit.

4.0 Evaluering

I kapitel 3 blev bla. beskrevet, hvordan den initiale nøgleudveksling (20 grupper af løsningsforslag) mellem noderne i et Z-Wave netværk kan foregå. Løsningsforslagene kan opdeles i to grupper; den ene hvor nøgleudvekslingen finder sted ved, at personen skal udføre en aktiv handling (SIL/DIL kontakter, trykknapper, tastatur, drejeomskifter, trimmer, lysdioder, 7-segment) og den anden, hvor personen stort set ikke skal foretage sig noget aktivt (RF, IR, Ledning, Forudprogrameret, Smart Card, Fingeraftryk, Stregkode, Magnetkort).

Ser vi på den første gruppe, så er der i afsnit 3.2.22 ”Opsummering af initial nøgleudveksling” vist hvilken løsning, der er bedst når man ser på forholdet mellem kostprisen og mapningen. Efter at have fundet nogle passende billige løsninger, vil vi nu se nærmere på hvor brugervenlige (ease-of-use) løsningerne er samt, hvor lang tid inkluderingen tager. Grunden til dette er, at installationsomkostninger er interessante og afhænger af hvor hurtig nøgleoverførelsen kan foretages, samt sandsynligheden for at foretage fejl. Vi går ud fra, at jo længere tid det tager at udføre en handling, des sværere er det at holde koncentrationen og risikoen for fejl stiger derfor. Endvidere tager vi udgangspunkt i, at det er nemmere at foretage få handlinger hvor man har overblikket end, at foretage mange handlinger hvor man kan miste overblikket.

Starter vi med SIL kontakterne, så er de meget små og de enkelte kontakter skal ikke flyttes meget for at skifte stilling, hvilket kan være svært at se. Der skal enten bruges en lille skrutrækker eller kuglepen for at skifte stilling og selv med dem, er det let at ramme ved siden af. Det er lettes med mange kontakter på række, så man ikke skal skifte dem så ofte som ved få, og ser vi på den enkelte kontakt, så skal stillingen skiftes 50 % af gangene. Tidsmæssigt er dette en fordel, men det kan hurtigt komme ud på ét, da brugeren skal bruge ekstra tid for at sikre sig, at kontakten har skiftet stilling. Pga. den meget lille fysiske udformning af kontakterne og deres skrøbelige udformning, frarådes denne løsningsgruppe.

DIL kontakterne er også små, men trods alt større end SIL kontakterne. DIL kontakterne er meget mere robuste og klikker tydeligt, når der skiftes stilling. Kontakterne er udformet, så skruetrækkeren/kuglepennen ikke så nemt glider væk. I en snæver vending er det muligt, at bruge neglene til at skifte stilling. Ligesom SIL kontakterne skal den enkelte kontakt skiftes 50 % af

gangene, men her går det hurtigere da man ikke er i tvivl om hvilken stilling kontakten står i. Løsningerne hvor der er 10/12 kontakter er værd at overveje, da der kun skal skiftes stilling 6 gange.

I næste løsningsgruppe bruges der ekstra trykknapper, for at foretage inkluderingen. Hvordan trykknapperne skal udformes og dermed hvor store/små de skal være, er op til producenten. Med få trykknapper skal brugeren holde knapperne nede mange gange og det er ikke så svært, men kan tage lang tid. Derimod hvis der bruges flere trykknapper kan det være svært, at finde ud af om hvorvidt den enkelte knap er trykket ned eller ej. Det er også lettere "at miste grebet", så man tror at en knap er trykket ned, selvom den ikke er det. Løsningsgruppen kan være svær at vurdere, da det ikke er til at vide, hvordan knapperne udformes. Det er svært at anbefale denne løsningsgruppe, da det er meget afhængigt af implementeringen. Er knapperne "store", nemme at trykke ned, klikker og sidder med en hvis afstand, kan løsningsgruppen overvejes, men da der ikke kan stilles krav til ovenstående, må vi generelt fraråde løsningsgruppen.

Tastaturløsningerne er store og meget dyre, så det frarådes at bruge dem.

Drejeomskifterne har den fordel, at de klikker når der skiftes stilling. Vi har set på versioner med forskelligt antal positioner og må konkludere, at dem med få indstillinger er nemme at bruge, tager tid men til gengæld er der større chance for, at positionen ikke skal skiftes. De versioner med mange indstillinger kan være svære at bestemme hvilken position omskifteren står i, så de er ikke særlig brugervenlige. Drejeomskifter til og med 10 positioner er det nemt og hurtigt at finde positionen, men den med 16 kræver lidt mere tid. Drejeomskifterne er dyre i forhold til funktionaliteten, så denne løsningsgruppe frarådes det at bruge.

Den næste løsningsgruppe gøre brug af to forskellige versioner af trimmere. Den ene type er meget billig, men har en lav drejningsvinkel og stor usikkerhed på det mekaniske og elektriske. Den anden type har stor drejningsvinkel, lave usikkerhedsmargener men er dyrere. Hvis man vælger et lavt antal inddelinger kan den billige version bruges, mens den dyre version må anbefales hvis der skal være mange inddelinger. Trimmerne har den ulempe, at skift mellem de forskellige positioner foregår flydende, dvs. der er ikke nogen klar indikationen af at stillingsskift er foretaget. Der skal bruges en lille skrutrækker for at skifte stilling. Løsninger med få positioner tager lang tid da mange

indstillinger skal foretages, men det gør løsninger med mange positioner også, da man her skal bruge mere tid på at ramme helt rigtigt. Der findes versioner med knap på, men de er så små, at man hurtigt bliver træt i fingrene. Man kan evt. overveje at sætte en stor knap på, da man nemmere vil kunne sætte positionen præcist. Alt efter implementeringen (antal inddelinger/størrelse på knap) kan disse løsninger anbefales.

Farvet lysdioder kan bruges til at overføre den initiale nøgle mellem noder. Idéen med at bruge farvet lysdioder er oplagt, da mange applikationer i dag bruger en lysdiode til at vise status. Vi har set på forskellige metoder og må konkludere at de er meget brugervenlige. Jo flere farver der bruges, des færre gange skal brugeren taste på kontrolleren hvilke(n) farve(r) der blev vist. Blinkesekvenserne må ikke være for lange, da det så er svært/umuligt for brugeren at huske denne. Dette gælder især ved flere farver. De lysdioder der er blevet brugt, spænder vidt i pris. Vi fraråder at bruge full-colour lysdioden, da den er meget dyr i forhold til, hvad man kan få ud af den. Det er vigtigt, at der er stor kontrastforskel mellem de forskellige farver, så brugeren ikke er i tvivl, hvorfor diodens teoretiske potentiale ikke kan udnyttes. Det anbefales enten at bruge én-, to- eller tre-farvet lysdiode pga. den meget lave pris og store brugervenlighed.

Bruges 7-segmenter til inkluderingsproceduren, må det betegnes at være en brugervenlig løsning. Alle er vant til at aflæse 7-segmenter fra hverdagen, og holder man sig til kun at vise tal mellem 0 og 9, så er ingen i tvivl om, hvilken tast der skal trykkes på kontrolleren. Rent fysisk fylder et 7-segment lidt plads, hvorfor de fleste producenter nok vil foretrække at gemme det/dem væk. Betjeningen er nem og der vindes intet ved, at bruge flere 7-segmenter andet end tid, som vi ikke mener kan opveje meromkostningen. Den højere kostpris gør, at de ikke er lige så effektive (mapning/pris) i forhold lysdioder men må betegnes som anbefalelsesværdige.

Løsningsgrupperne ovenfor har som før skrevet det tilfælles, at brugeren skal deltage mere aktivt i inkluderingsproceduren. Ved de resterende løsninger skal brugeren kun foretage sig noget én(to) gang(e), og vi vil her se nærmere på brugervenligheden af disse.

Ved brug af ledning mellem enhederne foregår overførelsen hurtigt. Vi har set på forskellige stiktyper og anbefaler at man bruger mini jack typen, da det er hurtigt at sætte ind/tage ud og fylder fysisk set ikke meget.

En anden måde at overføre nøglen på er via Smart Card. Løsningen kræver en kortenhed både i kontroller og node og har en fysisk størrelse som gør, at den vil blive gemt væk bag en klap. Brugervenligheden er i top, da kortet bare skal tages ind og ud af kontroller og node.

Brug af magnetkort- eller stregkodelæser må konkluderes at være ikke eksisterende alene pga. prisen. Laser frarådes pga. mulighed for skadeliggørelse af mennesker og ultra lyd frarådes, da det meget nemt vil være muligt at "lytte med". Løsningen som indebærer brug af PAN, må vi også fraråde, da det ikke er muligt at få funktionsdygtige implementeringer.

Sidste løsning hvor brugeren skal foretage sig minimalt er at gøre brug af IR. Alle løsninger er baseret på, at den mindste omkostning placeres på noderne, mens den største på kontrollerne, men her vil det være en klar fordel at bytte om på denne politik. Brug af IR gør, at brugeren bare skal pege kontrolleren mod noden, der skal inkluderes og så overføres den initiale nøgle. Løsningen anbefales kraftigt, da den er billig, trådløs og hurtig. Vi vil anbefale, at bruge løsningen hvor emitteren er meget retningsbestemt (16°) og modtageren har en stor åbningsvinkel (150°). Dette bevirker nemlig, at brugeren ikke skal stå lige foran noden samt, at lysbølgerne rammer inden for et lille område, dog skal enhederne være i line-of-sight. En anden positiv konsekvens er, at det ikke er strengt nødvendigt at stå med noden i hånden, når den skal inkluderes. Modtageren er ikke udstyret med IR filter, så anden stærk lyskilde kan virke støjende og derved genere overførelsen. I det tilfælde skal kontroller og node placeres tæt på hinanden for at undgå lys fra andre kilder. Løsningen betegnes at være meget fleksibel og anbefalelsesværdig.

Evalueringerne af de forskellige løsninger er samlet i bilag F. Her ses en oversigt, der vha. farvekoder indikerer hvor billig løsningen er, hvor nem den er at bruge, hvor lang tid det tager at inkludere noder samt hvor mange I/O forbindelser det kræver på ASIC'en.

Udfra de ovenstående gennemgang anser vi for løsningen, der gør brug af IR som den bedste, derefter løsninger der bruger lysdiode eller trykknap.

Til forhindring af replay attacks er der kigget på løsningsforslag som Challenge/Response, Sequence Numbers, Time Stamps og en kombination af Time Stamps og Challenge/Response.

Sidste nævnte løsning ser ud til at være bedste og mest realistiske løsning for forhindring af replay attacks mod Z-Wave. Denne løsning er meget sikker til at forhindre replay attacks i Z-Wave, hvis der er god synkronisering af urene på enhederne og hvis man kan bestemme transporttiden af meddelelser.

5.0 Konklusion

I rapporten er to vigtige sikkerhedselementer (initial nøgleudveksling og replay attacks) undersøgt for det trådløse netværk Z-Wave. Der blev analyseret en del løsningsforslag til de to sikkerhedsbegreber, så vi tilsidst kunne bestemme de bedste løsninger til initial nøgleudveksling mellem enhederne i et Z-Wave netværk, og forhindring af replay attacks mod enhederne i Z-Wave netværk.

Med udgangspunkt i at enhederne i netværket har begrænset ressourcer, dvs. lille regnekraft, lille lager samt en lav båndbrede er i alt gennemgået 20 løsningsforslag til, hvordan initial nøgleudveksling mellem enhederne i et Z-Wave netværk kan finde sted. Til løsningerne er kun medtaget de komponenter som er absolut nødvendige. Dette er gjort for, at holde kostprisen nede samt ikke at bruge for meget fysisk plads. Fordele og ulemper af designet af de forskellige løsninger er fundet efter analyse. En detaljeret beskrivelse og analyse af disse løsningsforslag findes i afsnittet 3.2 her i rapporten.

I opsummeringen i afsnit 3.2.21 er løsningsforslagene som kræver at brugeren deltager aktivt i inkluderingsproceduren kort opridset, og der er set nærmere på, hvilke løsninger der er mest effektive. Dette er gjort ved at se på, hvor mange handlinger brugeren skal foretage sig for at inkludere en enhed til netværket. Vi har fundet frem til, at den løsning der er billigst og mest effektiv er ved brug af fler-farvet lysdiode, derefter kommer brug af trimmer og så DIL-kontakter.

I kapitel 4 er løsninger blevet evalueret gennem eksperimentelle forsøg, som vi har udført. Vi har fundet frem til, at selvom en løsning synes effektiv, kan den være svær for brugeren. Dette kan f.eks. være pga. små dimensioner eller mange gentagelser skal udføres, hvormed koncentrationen mistes med fejl til følge. Vi har fundet frem til, at den mest optimale løsning er ved brug af IR, da det ikke kræver meget af brugeren. Ved inkluderingen skal enhederne kunne se hinanden og det er ikke streng nødvendigt, at stå med enhederne i hånden. Det gør løsningen meget fleksibel i forhold til de andre foreslag, da installationen kan effektiviseres og opsætningstiden holdes på en minimum. Det har ikke været muligt, at foretage forsøg med IR, så det må anbefales.

For forhindring af replay attacks er der gennemgået fire løsningsforslag. En detaljeret beskrivelse og analyse af disse løsningsforslag findes i afsnittet 3.3 her i rapporten og en evaluering/opsummering findes i slutningen af afsnittet. Efter analyse af de fire løsningsforslag har vi bestemt den bedste og mest realistiske løsning mod forhindring af replay attacks. Det er løsningsforslaget, som er en kombination af Time Stamps og Challenge/Response. De andre løsninger: Challenge/Response, Sequence numbers og Time Stamps er også sikre løsninger mod forhindring af replay attacks, men disse kræver enten for meget kapacitet, har for stort overhead eller for stort tidsvindue skal bruges. For meget kapacitet/overheads forbrug er ikke særlig optimalt i Z-Wave netværket, da noderne i netværket er begrænset med kapacitet. Time Stamps løsningen kræver ikke ekstra kapacitet eller overhead, men tidsvinduet kan nemt blive for stort hvilket gør, at angribere har tid nok til at replay de opsnappede meddelelser.

En kombination af Time Stamps og Challenge/Response løsningen er den mest realistiske og bedste løsning til forhindring af replay attacks i et Z-Wave netværk. Man begrænser nemlig tidsvinduet (helt ned til ca. ¼), hvilket gør, at angribere får svært ved at opsnappe en meddelelse og replay den igen på en kort tid. 80 % af de sendte meddelelser accepteres ved brug af Time Stamps og de resterende 20 % (hvor meddelelser ikke accepteres med Time Stamps) accepteres med Challenge/Response. Da kun 20 % af meddelelserne accepteres med Challenge/Response løsning, er overheadet begrænset. For at Time Stamps løsningen fungerer optimalt, kræver det, at der er god synkronisering af urene i enhederne i netværket og at transporttiden af meddelelser kan forudbestemmes. I afsnit 3.3 er der gennemgået, hvordan synkronisering af urene kan finde sted. Bestemmelse af transporttiden i netværket er et svært emne. I bedste tilfælde kan man bestemme transporttiden (se afsnit 3.3) men i værste tilfælde er det svært, da man ikke kan vide hvor stort og belastet netværket er samt, hvor meget information der skal transporteres. Alt i alt er løsningsforslaget, kombination af Time Stamps og Challenge/Response, en god løsning for forhindring af replay attacks i et Z-Wave netværk.

Vi har også kigget på, hvor tit der skal fornyes nøgler i et Z-Wave netværk, for at sikre at angribere ikke knækker nøglen i netværket. For at løse dette problem har vi kigget på Unicity distance og de eksisterende informationer om hvor lang tid det tager at knække DES, da der i nuværende tidspunkt ikke findes nogen viden om hvor lang tid det tager at knække 3DES, som Z-Wave bruger. Ved at kigge på Unicity distance og information om knæktiderne for DES, har det vist sig at det vil være

en god idé at forny nøgler i netværket en gang om måneden. Man kan skifte nøglerne i netværket hyppigere i kortere perioder, hvis man har tid, men hér skal man være opmærksom på at dette vil belaste netværket. Alt efter brugernes egne interesser og ønsker om, hvor tit han vil skifte nøglerne i netværket. Vi anbefaler at man i ”værste” tilfælde skifter nøglerne mindst én gang om året.

Rasmus Christiansen	***Yavuz Seker***

6.0 Bilag

Bilag A – US Patent – Security apparatus and method during Bluetooth pairing

US 20030050009A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2003/0050009 A1**

Kurisko et al. (43) **Pub. Date: Mar. 13, 2003**

(54) **SECURITY APPARATUS AND METHOD DURING BLUETOOTH PAIRING**

(76) Inventors: **Mark A. Kurisko**, Orefield, PA (US); **Philip D. Mooney**, Sellersville, PA (US)

Correspondence Address:
MANELLI DENISON & SELTER PLLC
7th Floor
2000 M Street, N.W.
Washington, DC 20036-3307 (US)

(21) Appl. No.: **09/949,673**

(22) Filed: **Sep. 12, 2001**

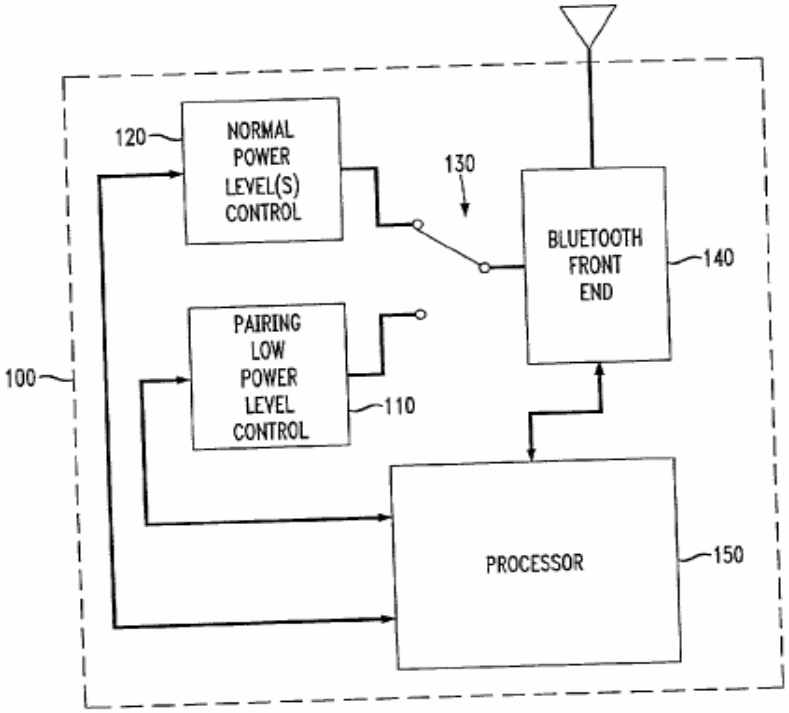
Publication Classification

(51) **Int. Cl.**[7] **H04B 5/00**

(52) **U.S. Cl.** **455/41**; 455/522; 455/67.1

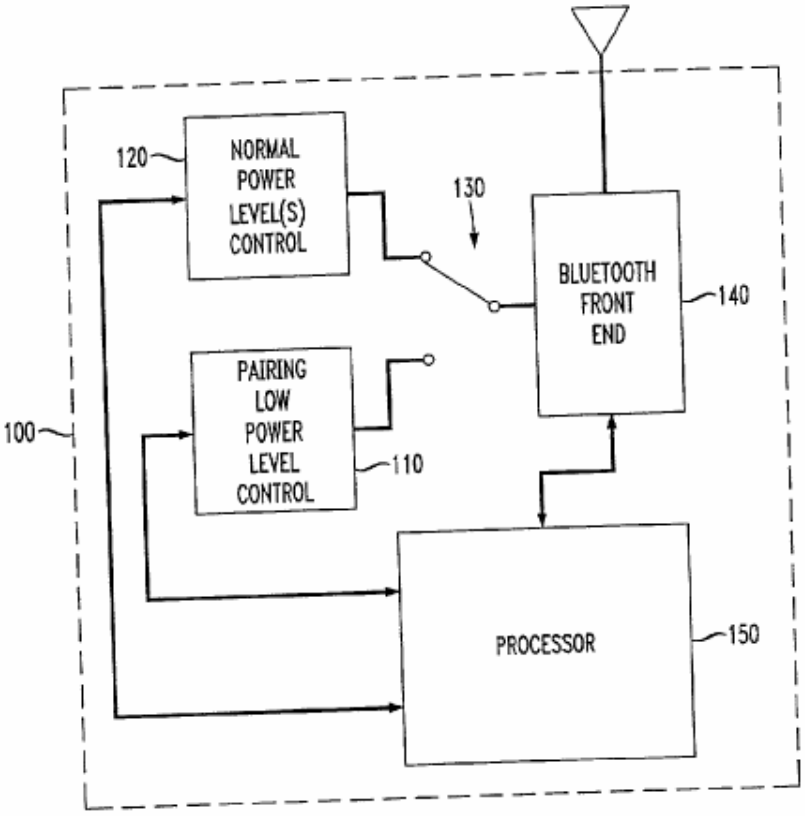
(57) **ABSTRACT**

A BLUETOOTH device is provided wherein the output RF transmission power level during pairing is purposefully reduced from otherwise conventional or normal communication levels to a low power level, greatly reducing the range of possible interception. Security can be improved even more by further reducing the transmit power even below that defined for a class 2 radio to an extremely low power level. After the link keys have been passed and/or other pairing processes, the BLUETOOTH devices may safely return to normal power levels to continue communications. Thus, a BLUETOOTH device is forced to radiate in low power when pairing is performed. The user(s) may be directed to co-locate the pairing BLUETOOTH devices in any appropriate manner, e.g., through a display prompt on the BLUETOOTH device. In an alternative embodiment, a BLUETOOTH device may be required to transmit data keys (e.g., a link key) and/or other pairing operations over a temporary wired connection (or temporary line-of-sight or near line-of-sight connection such as infrared) to another BLUETOOTH device.



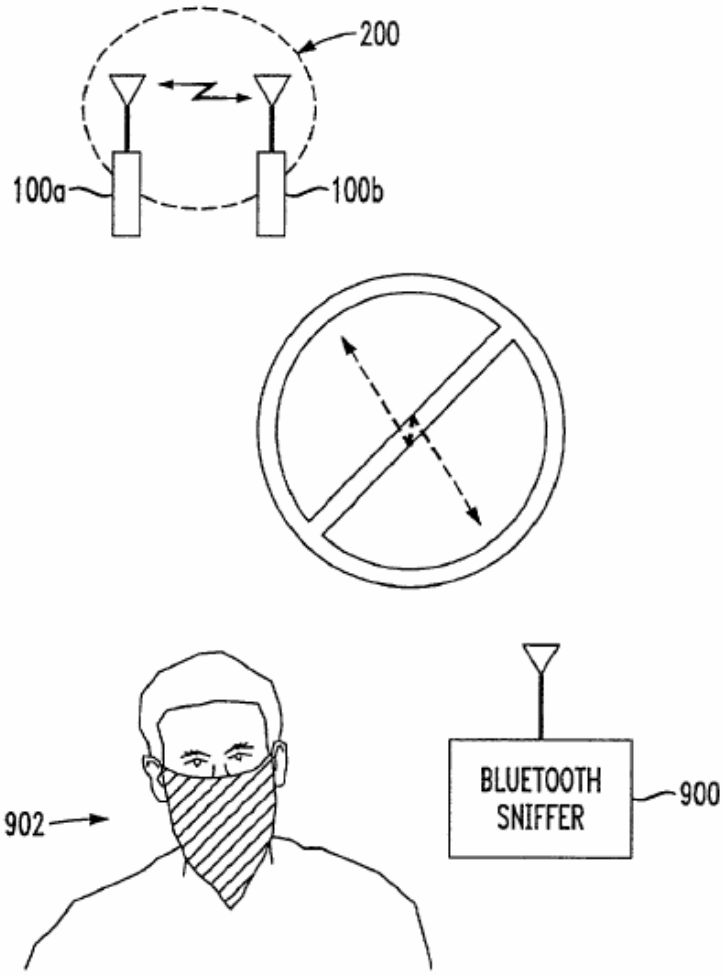
Patent Application Publication Mar. 13, 2003 Sheet 1 of 7 US 2003/0050009 A1

FIG. 1



Patent Application Publication Mar. 13, 2003 Sheet 2 of 7 US 2003/0050009 A1

FIG. 2



Patent Application Publication Mar. 13, 2003 Sheet 3 of 7 US 2003/0050009 A1

FIG. 3

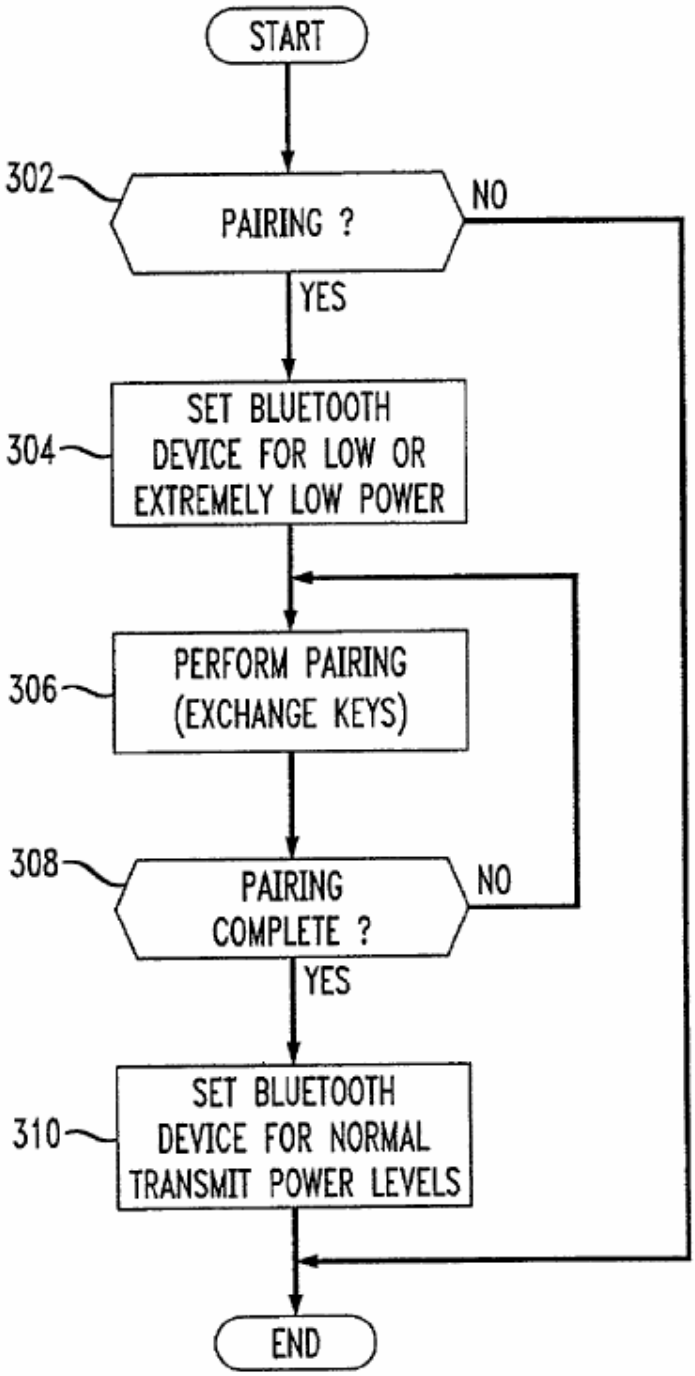


FIG. 4

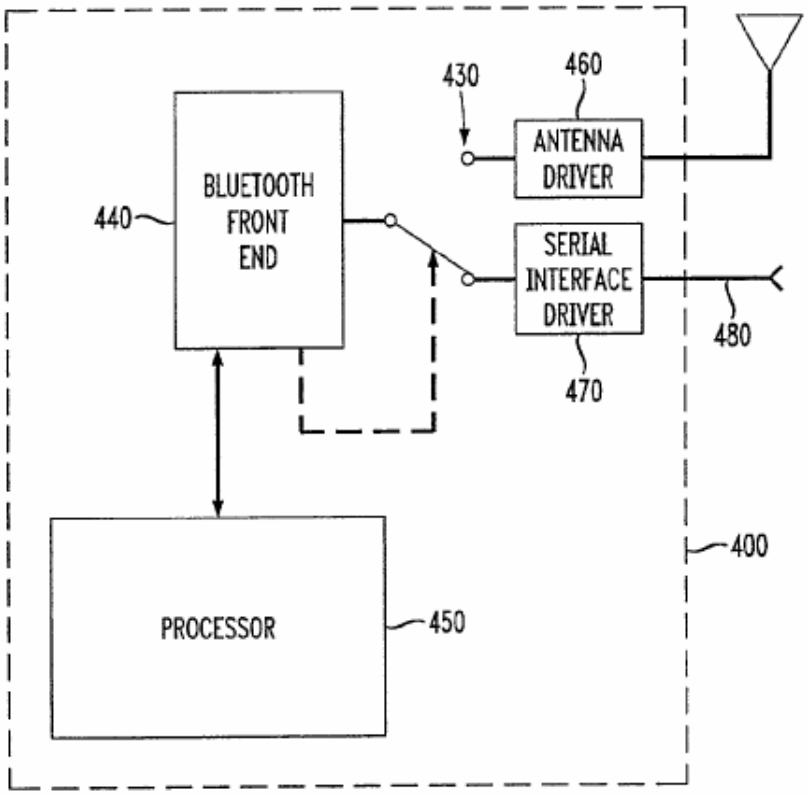
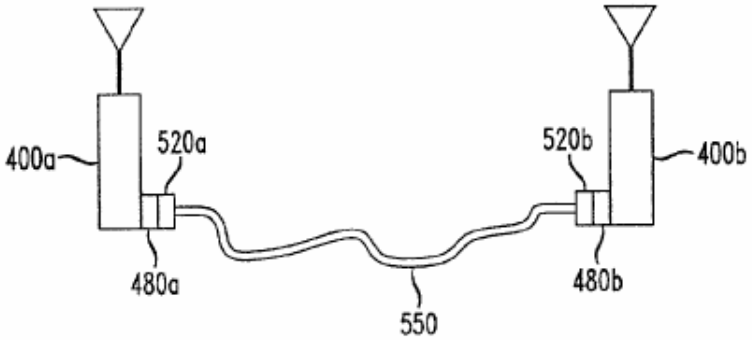

430
460
ANTENNA DRIVER
BLUETOOTH FRONT END
440
SERIAL INTERFACE DRIVER
480
470
400
PROCESSOR
450

FIG. 5


400a
520a
520b
400b
480a
480b
550

Patent Application Publication Mar. 13, 2003 Sheet 5 of 7 US 2003/0050009 A1

FIG. 6

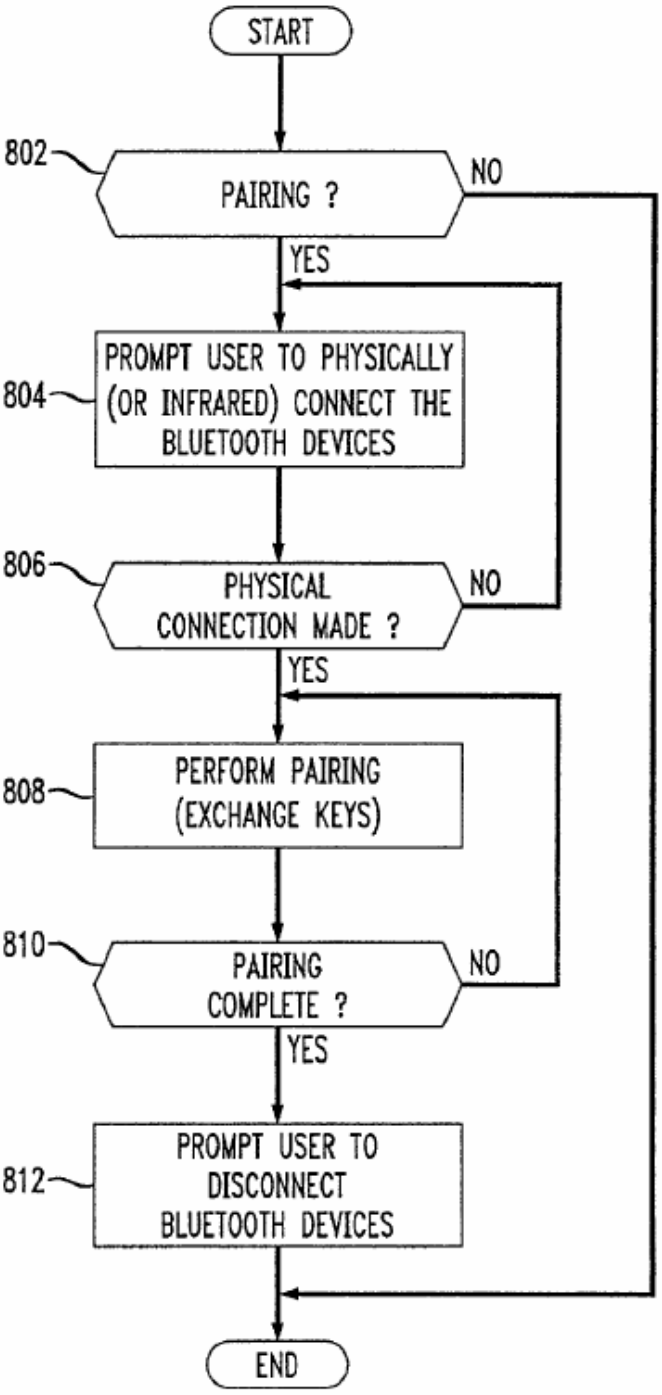


FIG. 7
PRIOR ART

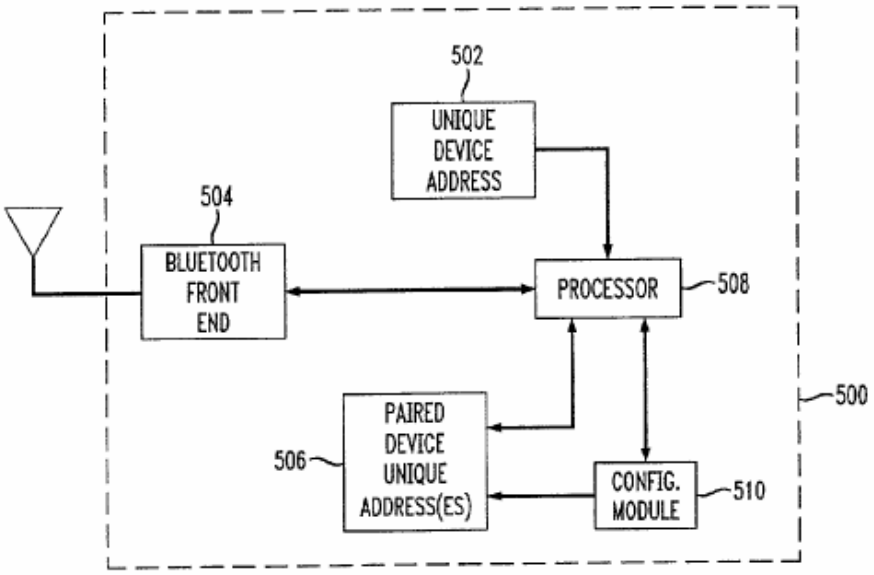
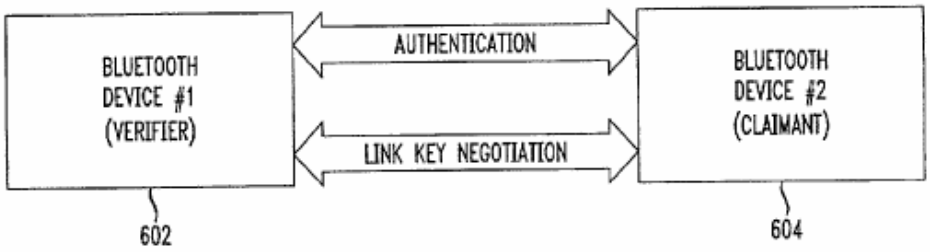
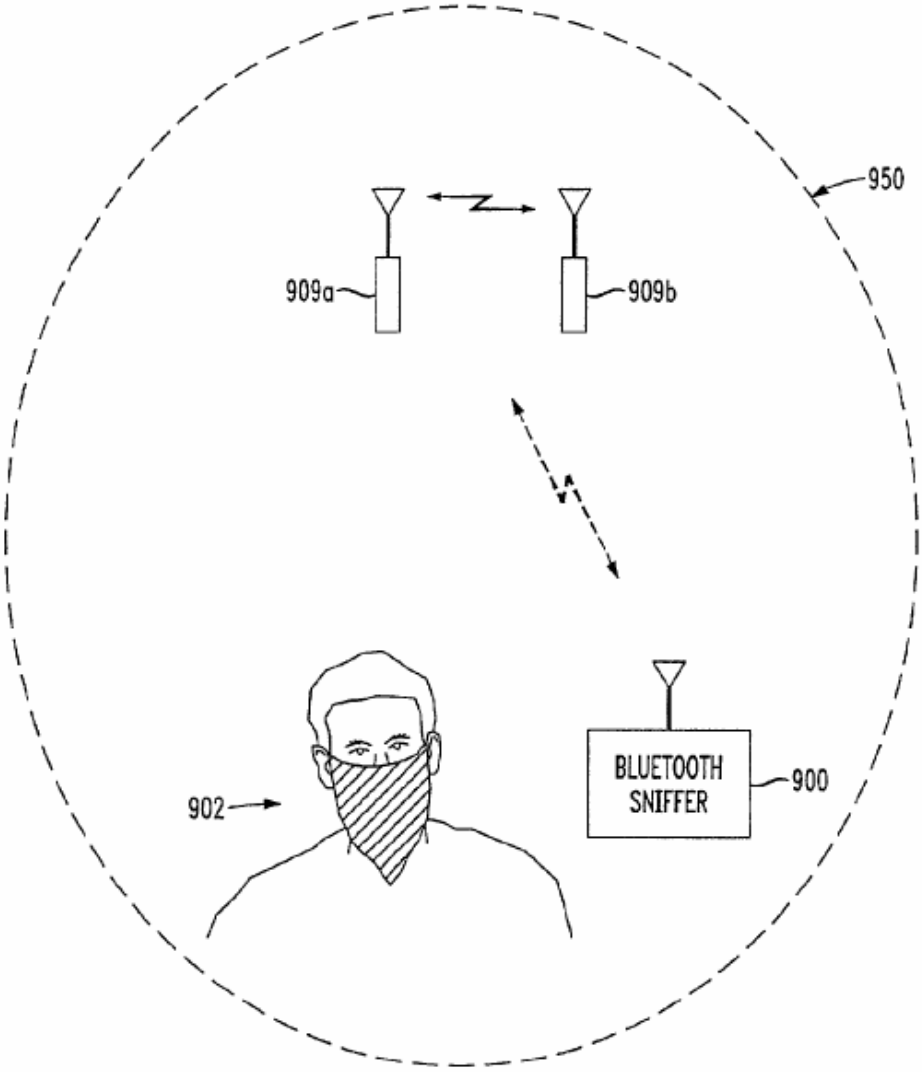


FIG. 8
PRIOR ART



Patent Application Publication Mar. 13, 2003 Sheet 7 of 7 US 2003/0050009 A1

FIG. 9
PRIOR ART



US 2003/0050009 A1 Mar. 13, 2003

SECURITY APPARATUS AND METHOD DURING BLUETOOTH PAIRING

BACKGROUND OF THE INVENTION

[0001] 1. Field of the Invention

[0002] This invention relates generally to piconet wireless networks. More particularly, it relates to a more secure pairing process in a piconet network such as a BLUETOOTH™ type piconet network.

[0003] 2. Background

[0004] Piconets, or small wireless networks, are being formed by more and more devices in many homes and offices. In particular, a popular piconet standard is commonly referred to as a BLUETOOTH™ piconet. Piconet technology in general, and BLUETOOTH technology in particular, provides peer-to-peer communications over short distances.

[0005] The wireless frequency of the piconets may be 2.4 GHz as per BLUETOOTH standards, and/or typically have a 20 to 1000 foot range. The piconet RF transmitter may operate in common frequencies which do not necessarily require a license from the regulating government authorities, e.g., the Federal Communications Commission (FCC) in the United States. Alternatively, the wireless communication can be accomplished with infrared (IR) transmitters and receivers, but this is less preferable because of the directional and visual problems often associated with IR systems.

[0006] A plurality of piconet networks may be interconnected through a scatternet connection, in accordance with BLUETOOTH™ protocols. BLUETOOTH network technology may be utilized to implement a wireless piconet network connection (including scatternet). The BLUETOOTH standard for wireless piconet networks is well known, and is available from many sources, e.g., from the web site www.bluetooth.com.

[0007] As part of an initial communication between BLUETOOTH devices, the BLUETOOTH devices within range of one another perform what is known in the art as "pairing".

[0008] **FIG. 7** depicts a conventional BLUETOOTH device **500**.

[0009] In particular, as shown in **FIG. 7, a** conventional BLUETOOTH device **500** includes a processor or logic device **508** (e.g., a microprocessor, a microcontroller, or a digital signal processor (DSP)), and a BLUETOOTH front end **504**. Moreover, the BLUETOOTH device **500** includes a unique 48-bit BD_ADDR **502**, and a table **506** containing a list of paired BLUETOOTH devices in the particular piconet. The paired device unique address table **506** may be pre-configured at the factory, or written to by a suitable user interface such as a software-based configuration module **510** allowing entry of the 48-bit address of paired devices for storage in the paired device unique address table **506**.

[0010] When configuring a BLUETOOTH device in a BLUETOOTH piconet, the devices communicating on the piconet must know the specific unique 48-bit address of matching devices on the piconet. For instance, it may be desirable for entertainment devices (e.g., TV, radio, CD player, DVD player, MP3 player, etc.) having BLUETOOTH communication capabilities to communicate with one another, but it may not be desirable (nor make sense) for appliances such as a stove or refrigerator, toaster, blender, etc. having BLUETOOTH communication capabilities talk with entertainment devices.

[0011] This is particularly true since the maximum number of BLUETOOTH devices in a piconet is somewhat restricted. For instance, current BLUETOOTH standards permit one (1) master and seven (7) slaves to be active in the piconet at any one time (plus a number of BLUETOOTH devices being capable of being 'parked').

[0012] According to the standard, all BLUETOOTH devices are assigned a unique 48-bit BLUETOOTH device address (BD_ADDR). This address is derived from the IEEE802 standard, and is divided into three fields: a lower address part (LAP) comprising 24 bits; an upper address part comprising 8 bits; and a non-significant address part (NAP) comprising 16 bits. The LAP and UAP form the significant part of the 48-bit BLUETOOTH device address (BD_ADDR). The total address space obtained is 2^{32}.

[0013] The BLUETOOTH device address (BD_ADDR) is unique for each BLUETOOTH device. The BLUETOOTH addresses are publicly known, and can be obtained by a manufacturer via MMI interactions, or, automatically, via an inquiry routine by a BLUETOOTH device. Blocks of 48-bit addresses may be assigned to various manufacturers, who in turn factory pre-configure each BLUETOOTH device to include a unique 48-bit address (BD_ADDR) as well as a table of unique 48-bit addresses of 'paired' devices which will all communicate over a common piconet.

[0014] When a user buys or replaces a BLUETOOTH equipped electronic device, the user must configure the new BLUETOOTH device for communication with relevant and desired devices in the relevant piconet. Moreover, to provide a certain level of security, the BLUETOOTH protocol provides for encryption of data passed therebetween. To this end, there are a number of different link and encryption keys currently used in BLUETOOTH, all of which are collectively referred to herein as 'data keys'.

[0015] For instance, link keys are used as authentication keys between BLUETOOTH devices, and to generate encryption keys.

[0016] A master key is used for point to multi-point communications, and may replace for a time the current link key.

[0017] A unit key is a semi-permanent, often ROM-based key generated in every single unit often only once during factory setup. Though unlikely, the unit key might be exchanged at any time.

[0018] A combination key is dependent on two BLUETOOTH devices. Each device produces and sends a random number to the other, and a new 128 bit combination key is derived using a SAFER+ algorithm. A combination key is often created toward the end of unit pairing.

[0019] A 128 bit initialization key is a link key used for a single session, and is created each time the BLUETOOTH device is initialized. An initialization key is used only when no combination keys or unit keys have been exchanged yet. An initialization key is often created toward the beginning of unit pairing.

[0020] An encryption key is derived from the current link key, and is used by an encryption engine to produce encrypted data.

[0021] **FIG. 8** depicts the authentication process and subsequent link key process between two BLUETOOTH devices.

[0022] To communicate, both BLUETOOTH devices **602**, **604** must share the same secret key. The secret key can be built in by manufacturers (a fixed key), or could be derived from a Personal Identification Number (PIN) or BLUETOOTH passkey.

[0023] To begin communicating with one another, the BLUETOOTH devices **602**, **604** bond by having link managers in the respective devices **602**, **604** verify with one another that they share a secret key through a process called authentication. While often time authentication takes place at link setup, it need not. After authentication, the link managers of the respective devices **602**, **604** create and exchange a link key. The process of authentication and link key generation are collectively called BLUETOOTH bonding or pairing.

[0024] If the BLUETOOTH devices **602**, **604** determine that they share the same secret key, then they go on to use their shared secret key to generate a link key and ultimately to encrypting traffic on the link.

[0025] The present inventors have appreciated that there is a weakness in the BLUETOOTH specification that might allow an adversary to steal the keys used for authentication and encryption that are intended to keep BLUETOOTH communications secure.

[0026] **FIG. 9** depicts the range of wireless communications between two BLUETOOTH devices during conventional pairing operations.

[0027] In particular, **FIG. 9** depicts two conventional BLUETOOTH devices **909***a*, **909***b* communicating using conventional BLUETOOTH RF messages during pairing, including the transmission of link keys. However, it is contemplated that a BLUETOOTH identity thief **902** might have a BLUETOOTH sniffer **900** be within range **950** of the BLUETOOTH devices **909***a*, **909***b* during their pairing process. The information gained by the BLUETOOTH sniffer **900** can prove disastrous to the users of the BLUETOOTH devices **909***a*, **909***b*.

[0028] For instance, an attack might be made during the initial pairing of two BLUETOOTH devices **602**, **604** that enables the adversary to intercept keys over the air and thereafter eavesdrop on future connections. Though BLUETOOTH transactions used for mobile commerce (m-commerce) that require a high level of security would most assuredly have greater security imposed by a higher layer (i.e. application layer using SSL, RSA, etc.) this security weakness in BLUETOOTH makes the user vulnerable to attack in two ways. First he or she could be impersonated by one who has intercepted the device addresses and keys. Possible examples would be impersonating a person's headset and stealing cellular air time or impersonating a person's laptop and stealing dial-up network access from the cell phone or stealing address book information.

[0029] Moreover, it is possible for an unauthorized receiver to eavesdrop on information passed between two (or more) BLUETOOTH devices **602**, **604**. Examples of the type of information would be non-encrypted e-mail, web sites being accessed, or even which stock quotes were being requested. Though some of this may not seem very important to some, it has the potential of providing an unfair and generally illegal advantage, particularly in the corporate or business world.

[0030] One possible way around the vulnerability of BLUETOOTH devices during pairing might be for a manufacturer to provide previously and permanently paired devices, paired in the secrecy and security of the manufacturing facility. However, such predetermined and/or dedicated pairing would tend to restrict use of the BLUETOOTH devices such that they would work only with other devices sold by the same manufacturer.

[0031] There is a need for a more secure pairing technology and apparatus with respect to piconet devices in general, and BLUETOOTH™ piconet devices in particular.

SUMMARY OF THE INVENTION

[0032] In accordance with the principles of the present invention, an apparatus and method of providing security during a network establishment operation (e.g., a pairing operation of a piconet network device) comprises receiving a link key during pairing, and rejecting the received link key if transmitted at a power level above a low power threshold specifically intended for very close range communications.

[0033] A method for securely transmitting a data key from a piconet device in accordance with another aspect of the present invention comprises forcing an RF front end of the piconet device to transmit at a low power level to transmit the data key to another piconet device at the low power level. After the data key has been transmitted, the RF front end of the piconet device is reset to transmit at a normal power level for ordinary communications.

BRIEF DESCRIPTION OF THE DRAWINGS

[0034] Features and advantages of the present invention will become apparent to those skilled in the art from the following description with reference to the drawings, in which:

[0035] **FIG. 1** shows a BLUETOOTH piconet device having a low power control for use with pairing operations including the transfer of keys, in accordance with the principles of the present invention.

[0036] **FIG. 2** depicts the very close range limit of BLUETOOTH devices pairing at low or extremely low powers, in accordance with the principles of the present invention.

[0037] **FIG. 3** is an exemplary process by which the BLUETOOTH device of **FIGS. 1 and 2** is set to low or extremely low transmit power for pairing operations.

[0038] **FIG. 4** shows another embodiment of the present invention wherein a BLUETOOTH device includes a physical or line-of-sight connector intended to provide temporary wired pairing operations with another BLUETOOTH device, in accordance with the principles of the present invention.

[0039] **FIG. 5** depicts two BLUETOOTH devices as shown in **FIG. 4** temporarily wired together to allow secure wired pairing therebetween.

[0040] **FIG. 6** shows an exemplary process by which the user of a BLUETOOTH device shown in **FIG. 5** is prompted to provide the temporary physical or line-of-sight connection to allow pairing operations, in accordance with the principles of the present invention.

[0041] **FIG. 7** depicts relevant features of a conventional BLUETOOTH device.

[0042] **FIG. 8** depicts the authentication process and subsequent link key process between two conventional BLUETOOTH devices.

[0043] **FIG. 9** depicts the range of wireless communications between two BLUETOOTH devices during conventional pairing operations.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

[0044] If one considers the range of a class I BLUETOOTH radio, the area where one could eavesdrop on the connection is over 30,000 square meters. If the three-dimensional aspects of the radio transmission are considered (e.g., an eavesdropper on another floor of a building) the number of places an adversary could be to intercept data packets is even greater than the two dimensional consideration.

[0045] As described above, the period of time in which the BLUETOOTH device users are most particularly vulnerable is during the pairing of two BLUETOOTH devices. In accordance with the principles of the present invention, RF transmissions are seriously reduced and/or replaced during the pairing of BLUETOOTH devices, to provide added security.

[0046] The present inventors have appreciated that during pairing, it is likely that the BLUETOOTH devices are closely located anyway, or at least that a temporary requirement to bring the devices close together will probably not be a large burden. By limiting the power level of RF transmissions in a BLUETOOTH device during pairing, the range of possible interception is greatly reduced from that otherwise allowed by the current BLUETOOTH standard.

[0047] For instance, if the radio power is limited to standard low power during pairing, the possible intercepting area would be just over 300 square meters. This is only one percent (1%) of the area in which the BLUETOOTH devices are vulnerable with the conventional use of the high powered radio during pairing, and far less if considering the three dimensional aspects such as in a large building.

[0048] While the use of standard low power specifically during pairing is within the scope of the present invention, security can be improved even more by further reducing the transmit power even below that defined for a class 2 radio. For instance, by adding an additional requirement that the two devices be only a few inches apart during pairing to allowing successful pairing at very low transmit levels, risk of eaves-dropping is extremely reduced from that otherwise allowed under the current BLUETOOTH standards.

[0049] In accordance with the principles of the present invention, the vulnerability is substantial during pairing operations only. Thus, after the link keys have been passed and/or other pairing processes, the BLUETOOTH devices may safely return to normal power levels to continue communications.

[0050] Thus, in accordance with the principles of the present invention, a BLUETOOTH device is forced to radiate in low power when pairing is performed.

[0051] It is also preferred that as pairing is being initiated, the BLUETOOTH device not accept temporary link keys from another BLUETOOTH device since it would be unsure what power that other device was transmitting at. Rather, it is preferred that the receiving BLUETOOTH device simply reject that pairing request, making a record of it, and then itself initiate key transfer back with that same other device in a low power mode in accordance with the principles of the present invention.

[0052] The user(s) may be directed to co-locate the pairing BLUETOOTH devices in any appropriate manner, e.g., through a display prompt on the BLUETOOTH device, through an audible instruction, through a written instruction sheet included with the BLUETOOTH device, etc.

[0053] **FIG. 1** shows the relevant elements of an exemplary BLUETOOTH piconet device having a low power control for use with pairing operations including the transfer of keys, in accordance with the principles of the present invention.

[0054] In particular, as shown in **FIG. 1, a** BLUETOOTH device **100** includes a BLUETOOTH front end **140** and a processor **150**. The processor **150** may be any suitable processing device, e.g., a microcontroller, microprocessor, digital signal processor (DSP), ASIC, etc.

[0055] Importantly, in accordance with the principles of the present invention, the BLUETOOTH device **100** includes a low RF power capability **110**, in addition to its normal operating level RF level(s) **120**. This is depicted in **FIG. 1** by the selection of either the normal power level(s) control module **120** or the pairing low power level control module **110** to control the RF output of the BLUETOOTH front end **140**. While this selection is shown in **FIG. 1** by way of a switching function **130**, this selection of course may be performed without the need for a physical switch, e.g., by software adjustment of a power level control register to the BLUETOOTH front end **140**.

[0056] In the given embodiment, the pairing low power level control **110** directs the BLUETOOTH front end **140** to transmit at a power not exceeding that which provides a nominal range of no more than about, e.g., preferably 10 meters. Of course, much smaller ranges are possible and preferable, within the principles of the present invention.

[0057] **FIG. 2** depicts the very close range limit of BLUETOOTH devices pairing at low or extremely low powers, in accordance with the principles of the present invention.

[0058] In particular, in **FIG. 2**, the thief **902** who was otherwise able to intercept BLUETOOTH pairing transmissions of the conventional BLUETOOTH devices **909***a*, **909***b* shown in **FIG. 9** is now foiled, because the much, much smaller range **200** of low power RF transmissions from the BLUETOOTH devices **100***a*, **100***b* during pairing operations (particularly when exchanging a key) in accordance with the principles of the present invention does not reach the BLUETOOTH sniffer **900** machine.

[0059] **FIG. 3** is an exemplary process by which the BLUETOOTH device of **FIGS. 1 and 2** is set to low or extremely low transmit power for pairing operations.

[0060] In particular, in step **302** of **FIG. 3**, it is determined whether or not a pairing operation is to be performed.

[0061] In step **304**, the BLUETOOTH device is set for low transmission power. Alternatively, if provided, the BLUETOOTH device may be set to an 'extremely low' power, i.e., to a power below the lowest used for ordinary communications (and/or below those in the current BLUETOOTH specification).

[0062] In step **306**, pairing operations are performed, including the exchange of data keys. In an alternative embodiment, the low power mode of the BLUETOOTH device may be used only to transmit a data key.

[0063] In step **308**, it is determined whether or not the pairing operation has been completed. If not, the pairing process continues in step **306**.

[0064] In step **310**, once the pairing process using a low transmission power has been completed, the BLUETOOTH device is reset for normal communication activities at a normal power level.

[0065] In an alternative embodiment, a BLUETOOTH device may be required to transmit data keys (e.g., a link key) and/or other pairing operations over a temporary wired connection to another BLUETOOTH device.

[0066] In particular, a temporary physical or line-of-sight (e.g., Infrared) communication path (collectively referred to herein as 'physical' connection) may be required for pairing operations. The requirement for a temporary physical connection provides a secure connection between two (or more) BLUETOOTH devices during pairing operations, removing the need to transmit initial link keys over the air. This eliminates the possibility of interception of pairing signals by an unauthorized device, e.g., by a BLUETOOTH Sniffer.

[0067] **FIG. 4** shows another embodiment of the present invention wherein a BLUETOOTH device includes a physical or line-or-sight connector intended to provide temporary wired pairing operations with another BLUETOOTH device, in accordance with the principles of the present invention.

[0068] In particular, in **FIG. 4**, a BLUETOOTH device **400** includes a BLUETOOTH front end **440** which has the option of outputting data in RF form through an antenna driver **460**, or through a serial interface driver **470** and associated serial interface **480**. The selection of RF or digital data transmission is depicted by a selectable switching function **430**, though the invention does not limit the selectability operation to a switch. For instance, a software setting such as in a register is suitable to change output options from the BLUETOOTH front end **440**, in accordance with the principles of the present invention.

[0069] The physical connection can be a short electrical or optical cable, e.g., provided by the manufacturer of the BLUETOOTH device.

[0070] **FIG. 5** depicts two BLUETOOTH devices as shown in **FIG. 4** temporarily wired together to allow secure wired pairing therebetween.

[0071] In particular, as depicted in **FIG. 5**, an electrical or optical cable **550** with connectors **520***a*, **520***b* may be connected between suitable and matching serial (or parallel) connectors **480***a*, **480***b* of the BLUETOOTH devices **400***a*, **400***b*, respectively.

[0072] Alternatively, the physical connection can be provided by a set of electrical contacts on one BLUETOOTH device that connects or meets with matching contacts on the other BLUETOOTH device. For instance, one BLUETOOTH device could include a pop-out connector that would not be visible under normal use, but which would extend to mate with the other BLUETOOTH device during the initial pairing operations.

[0073] The user can be prompted of the need to perform pairing operations, wait for confirmation of completion of the temporary physical connection (or monitor for it), and then perform the pairing operations only when the devices are physically connected.

[0074] In addition to providing extremely high security for the passage of data keys, the use of a physical connection between BLUETOOTH devices in a piconet to perform pairing operations reduces or eliminates the otherwise conventional hindrance associated with the higher level of technical expertise required by a user to properly initiate BLUETOOTH pairing.

[0075] **FIG. 6** shows an exemplary process by which the user of a BLUETOOTH device shown in **FIG. 5** is prompted to provide the temporary physical or line-of-sight connection to allow pairing operations, in accordance with the principles of the present invention.

[0076] In particular, in step **802** of **FIG. 6**, it is determined whether or not a pairing operation is to be performed.

[0077] In step **804**, the BLUETOOTH device prompts the user to physically connect the two (or more) BLUETOOTH devices to allow pairing operations to continue.

[0078] In step **806**, a physical connection is made by the user between the two BLUETOOTH devices, either using a cable, direct connector-to-connector contact between the two BLUETOOTH devices, using a line-of-sight infrared connection, etc.

[0079] In step **808**, the pairing operations are performed, including the exchange of data keys. In an alternative embodiment, the low power mode of the BLUETOOTH device may be used only to transmit a data key.

[0080] In step **810**, it is determined whether or not the pairing operation has been completed. If not, the pairing process continues in step **808**.

[0081] In step **812**, once the pairing process using a low transmission power has been completed, the user is prompted to disconnect the physical connection or otherwise is instructed that normal wireless range operations may commence or continue.

[0082] Provision of a physical connection also opens up the possibility for an improved user experience while pairing. For instance, standard over-the-air pairing requires multiple steps by the user. One device needs to be placed in a pairable mode, while the other must be told to initiate pairing. It then scans for all devices and asks the user to select which one is the desired one. The user then must enter a PIN (this step may or may not still be desirable).

[0083] By supplying a physical connection, other user interaction can be removed. The user plugs in the cable into both devices. They communicate over the cable so that RF eavesdroppers are thwarted. They exchange device addresses, names, class, etc. They negotiate who generates the initial temporary link key. The link key is generated. The PINs are used to generate semikeys and finally, the link keys are verified by successfully linking over the air (but not exposing any keys to those near-by).

[0084] Of course, a cable is not the only implementation of this invention. For instance, suitable line-of-sight or near line-of sight devices could be used. An IrDA infrared link is an example of an alternative communication mechanism.

[0085] In another aspect of the invention, security in a BLUETOOTH device is enhanced by causing the RF front end of a first piconet device to transmit the data key along a directed path towards a second piconet device. Typically, the directed path is a straight line connecting the first piconet device to the second piconet device.

[0086] In known telecommunication systems, an RF front end transmits a signal to a receiving system. The RF front end transmits the signal in an outward radiating pattern from the RF front end. In comparison, under this aspect of the invention, the RF front end does not transmit the signal in an outward radiating pattern. Rather, the signal is transmitted along a linear path towards a receiver.

[0087] By directing the data key along a directed path from the first piconet device towards the second piconet device, the overall security of the system is increased. In order to effectively snoop the present embodiment, a BLUETOOTH snooper would be required to be positioned in a linear path directly between the first and second piconet devices. A BLUETOOTH snooper positioned merely in the vicinity of the first or second piconet devices would not effectively capture the transmitted data (e.g., the data key), unless the snooper is positioned along the directed path between the first and second piconet devices.

[0088] One of ordinary skill in the art will recognize that there are numerous ways to direct a transmission along a directed path. For instance, two or more antennas can be used to form a beam forming network that can direct a transmitted signal. Such a beam forming network would allow a first piconet device to securely connect to a second piconet device by pointing the first piconet device towards the second piconet device.

[0089] The present invention improves BLUETOOTH security during pairing, and reduces the possibility of identity impersonation and/or eavesdropping. Pairing operations become simpler because there need be less interaction by the user, leading to a reduced risk of error in the pairing.

[0090] The disclosed embodiments reduce or eliminate the risk of someone intercepting the RF data sent during pairing between two devices. Once pairing has finished, further risk of eavesdropping has been virtually eliminated. A first embodiment restricts the transmit power while link keys are being passed. A second embodiment replaces over the air link exchange with a physical electrical connection. This also removes the need of a user interface to initiate pairing. In any event, these two embodiments in particular solve problems associated with BLUETOOTH eavesdropping.

[0091] The disclosed embodiments reduce security weaknesses found in BLUETOOTH by reducing or removing the risk of eavesdropping during the insecure period of pairing. Invention 2 also removes the requirement of the user initiating the pairing process through buttons or menus on one or both devices, making it much easier to take advantage of the BLUETOOTH connection, yet it does not force a permanent pairing as fixed link keys would.

[0092] While the invention has been described with reference to the exemplary embodiments thereof, those skilled in the art will be able to make various modifications to the described embodiments of the invention without departing from the true spirit and scope of the invention.

What is claimed is:

1. A method of providing security during a pairing operation of a wireless network device, comprising:

receiving data associated with network establishment during network establishment; and

rejecting said received data if transmitted at a power level above a low power threshold specifically intended for very close range communications.

2. The method of providing security during a pairing operation of a wireless network device according to claim 1, wherein:

said network establishment is a pairing operation.

3. The method of providing security during a pairing operation of a wireless network device according to claim 1, wherein:

said data includes link key information.

4. The method of providing security during a pairing operation of a wireless network device according to claim 1, wherein:

said wireless network device is a BLUETOOTH piconet device.

5. A method for securely transmitting a data key from a wireless network device, comprising:

forcing an RF front end of said wireless network device to transmit at a low power level to transmit said data key to another network device at said low power level; and

after said data key has been transmitted, resetting said RF front end of said wireless network device to transmit at a normal power level for ordinary communications.

6. The method for securely transmitting a data key from a wireless network device according to claim 5, wherein:

said wireless network device is a piconet network device.

7. The method for securely transmitting a data key from a wireless network device according to claim 5, wherein:

said low power level is lower than a lowest normal communication power level.

8. The method for securely transmitting a data key from a wireless network device according to claim 5, wherein:

said wireless network device is a BLUETOOTH piconet device.

9. The method for securely transmitting a data key from a wireless network device according to claim 5, wherein:

said RF front end is forced to transmit at said low power level during pairing operations between said wireless network device and said another wireless network device.

10. The method for securely transmitting a data key from a wireless network device according to claim 5, wherein:

said RF front end of said wireless network device transmits said data key along a directed path towards another wireless network device.

11. Apparatus for providing security during a pairing operation of a wireless network device, comprising:

means for receiving data during network establishment; and

means for rejecting said received data if transmitted at a power level above a low power threshold specifically intended for very close range communications.

12. The apparatus for providing security during a pairing operation of a wireless network device according to claim 11, wherein:

said network establishment is a pairing operation.

13. The apparatus for providing security during a pairing operation of a wireless network device according to claim 11, wherein:

said data is a link key.

14. The apparatus for providing security during a pairing operation of a wireless network device according to claim 11, wherein:

said wireless network device is a BLUETOOTH piconet device.

15. Apparatus for securely transmitting a data key from a wireless network device, comprising:

means for forcing an RF front end of said wireless network device to transmit at a low power level to transmit said data key to another wireless network device at said low power level; and

means for resetting said RF front end of said wireless network device to transmit at a normal power level for ordinary communications after said data key has been transmitted.

16. The apparatus for securely transmitting a data key from a device according to claim 15, wherein:

said low power level is lower than a lowest normal communication power level.

17. The apparatus for securely transmitting a data key from a device according to claim 15, wherein:

said wireless network device is a BLUETOOTH piconet device.

18. The apparatus for securely transmitting a data key from a wireless network device according to claim 15, wherein:

said means for forcing said RF front end sets said RF front end to transmit at said low power level during pairing operations between said wireless network device and said another wireless network device.

* * * * *

Bilag B – DES tabeller og S-bokse

DES-tabellerne

Den initiale permutation ***IP***;

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Den iverse permutation IP^{-1};

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Ekspansionsfunktionen ***E***;

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

De otte *S-bokse*

S1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

***P-box* permutation:**

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

***PC-1* permutation:**

C_i

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36

D_i

63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

***PC-2* permutation:**

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Bilag C – Tal Teori

Talteori beskæftiger sig med egenskaber ved de heltal $\mathbb{Z}$,som er (....,-3,-2,-1,0,1,2,3,...). Nedenunder gennemgåes (kort) de vigtige begreber inden for tal teori[101].

Primtal

For at forstå hvad *primtal* er fastlægges følgende definition;

Definition: Et helttal ***p>1*** kaldes et ***primtal***, hvis ***p*** kun har ***1*** og ***p*** som divisorer.

Et ***sammensatte tal*** et et ***tal>1***, som ikke er et primtal. Tallet 1 er altså "hverken eller".

Division og rester

Definition: Et heltal $a \neq 0$ siges at være divisor i et heltal b, hvis der findes et heltal q sådan at $b = qa$, dvs a går op i b, altså $a|b$.

Der gælder også følgende; For alle $m \in \mathbb{Z}$ og $n \in \mathbb{N}$ findes entydigt bestemte hele tal q og r så

$$m = qn + r, \qquad 0 \leq r < n$$

Modulo

Definition: For vilkårlige hele tal m og n, hvor $n>0$ defineres $m \pmod n$ som;

$m \pmod n$ = *den principale rest af* m *ved division med* n

Hvis $a, b,$ og n er heltal og $n>0$ siges a at være kongruent med b ***modulo*** n, hvis n går op i $(a - b)$.

Dette skrives $a \equiv b \pmod n$.

Neden under kan vises nogle regneregler for modulær aritmetik.

Kongruenser:	$a \equiv b \pmod n \Leftrightarrow a \pmod n = b \pmod n$
Addition:	$a + b \pmod n = a \pmod n + b \pmod n$
Subtraktion:	$a - b \pmod n = a \pmod n - b \pmod n$
Multiplikation:	$a * b \pmod n = a \pmod n * b \pmod n$

[101] For mere information om tal teori, se Handbook of Applied Cryptography by A. Menezes, P. van Oorschot and S. Vanstone.

Multiplikation kan retultere i $a * b = 0$, selv om hverken a eller b er 0, f.eks. *3 * 2 mod 6 = 0*

Division: $a / b \pmod n = a * b^{-1} \pmod n$

Der findes altså et $b^{-1} \pmod n$ sådan at $b * b^{-1} = 1 \pmod n$. F.eks. *2 * 3 = 1 (mod 5)* således at *4 / 2 = 4 * 3 ≡ 2 (mod 5)*

Grupper

En gruppe $G = <G,*,e,()^{-1}>$ er en mængde G af elementer med en associativ *multiplikation* (*), et *identitetselement* e hvor $e * x = x * e = x \ (x \in G)$ og et *inverselement* $()^{-1}$ med $x * x^{-1} = x^{-1} * x = e$.

En *cyklisk gruppe* er en gruppe hvor alle elementerne i gruppen kan frembringes ved successiv potensopløftning af et bestemt element g i gruppen: $G = \{g^0, g^1, g^2, g^3, ..., g^k = g^0\}$. Elementet g kaldes en ***generator*** og antallet af forskellige elementer k kaldes gruppens ***orden***.

Største fælles divisor, gcd()

Definition: Den største fælles divisor d = gcd (a, n) for a og n er det største tal d som går op i både a og n.

Man bruger generelt Euklids algoritme[102] til at finde største fælles divisor ***gcd(a,n)*** for to tal a og n, hvor $a < n$. Man bruger den egenskab at hvis d går op i a ($d|a$) og d går op i n ($d|n$) så går d og så op i $n-a$ ($d|(n-a)$), der gælder altså $gcd\ (a,n) = gcd\ (n-a,n)$, hvilket er sand pga enhver divisor e den ene også er en divisor i den anden.

Eulers φ-funktion (Totient-funktionen)

To heltal a og n siges at være indbyrdes primiske, hvis de har største fælles divisor gcd(a, n) = 1. Man siger også at a er ***primisk relativt*** til n. Ved regning (***mod n***) er der en delmængde af den fuldstændige mængde af rester bestående af de rester, der er relativt primiske til n, dvs de har ikke andre fælles divisorer med n end 1. F.eks. for $n = 10$ bliver den fuldstændige mængde af rester *{0,1,2,3,4,5,6,7,8,9}* mens den reducerede delmængde af rester relativt primiske til $n = 10$ er mængden *{1,3,7,9}*. Alle elementerne i denne mængde har hver et inverst element, hvor mængden af *invertible elementer* betegnes Z_n^* .

Eulers Totient fuktionen af et tal er defineret som mængden af heltal mindre end det tal der er relativ primtal til tallet. Eulers φ-funktion er et eksempel på en aritmetisk eller talteoretisk funktion,

[102] Kapitel 2 i Handbook of Applied Cryptography by A. Menezes, P. van Oorschot and S. Vanstone

altså en funktion der er defineret på heltal $\mathbb{Z}$. En aritmetisk funktion ***f*** siges at være *multiplikativ*, hvis ***f (mn) = f (n) * f(m),*** når ***gcd(m,n) = 1***.

Eulers φ-funktion defineres således;

$$\varphi(1) = 1$$

$$\varphi(n) = \text{antallet af elementer i for } Z_n^* \quad n = 2,3,4,5,...$$

Hvis ***m = p*** og ***n = q*** er primtal så gælder der;

$$\varphi(p) = p - 1$$

$$\varphi(p * q) = (p - 1) * (q - 1)$$

Eulers sætning: Hvis ***gcd (a,n) = 1*** så gælder der $a^{\varphi(n)} \equiv 1 \pmod n$.

Fermats lille sætning: Hvis ***p*** er et primtal og ***gcd (a, p) = 1***, så gælder der $a^{p-1} \equiv 1 \pmod n$.

Både *Eulers* og *Fermats lille sætninger* er vigtige sætninger ved brug af **RSA**.

Bilag D – Mapning mellem bits i nøgle og overført bits

Dette bilag viser og gennemgår sammenhængen i mapning mellem bits'ne i den initielle nøgle, som noderne skal bruge ved inkluderingsproceduren til netværket, og antallet af handlinger som brugeren skal foretage for at overføre disse.

Mapningstabellen er opbygget ved i første søjle (# bits) at vise antallet af bits i nøglen, gående fra 1 til 56. Vi bruger 56 bits i den initielle nøgle, så det er i realiteten kun den sidste række vi bruger, men for komplethedens skyld er oversigten lavet for alle nøglelængder til og med 56. Dette gør det nemt at finde mapningsværdier samt, at finde ud af hvor mange handlinger der skal foretages, hvis en anden nøglelængde skal bruges i stedet for de 56.

Anden søjle (# kom.) i tabellen viser antallet af kombinationer, som nøglen har ud fra længden af nøglen. Værdierne er udregnet således: $2^{\#\ \text{bits i nøgle}}$. For kontakter m.m. som kun har 2 poler/stillinger, er antallet af handlinger lig med nøglelængden, da mapningen er 1. Det er først ved kontakter m.m., som har mere end to stillinger, at tabellen kommer til sin ret, hvilket vil fremgå af det følgende.

De efterfølgende søjler i tabellen hører sammen parvis. Værdierne i første søjle i parret (x polet) er udregnet ligesom anden søjle i tabellen (# kom.) ved, at udregne $x^{\#\ \text{handlinger brugeren skal foretage}}$, mens anden søjle (#) angiver antallet af handlinger. Antallet af kombinationer (x polet) er opstillet ud fra søjlen med antallet af kombinationer i nøglen (# kom.), således at antallet af kombinationer for x pol er lig med eller større end antallet af kombinationer i nøglen (# kom.). Hvis ikke der er nogen værdi, så skyldes det at antallet af kombinationer for den x polet komponent, bliver mere end fordoblet og derfor kan dække over en større nøgle, hvorfor værdien længere nede i samme søjle skal bruges. Et eksempel vil belyse dette nærmere.

# bits	# kom.	3 polet #	4 polet #	5 polet #	6 polet #	7 polet #
1	2	3 (1)				
2	4		4 (1)	5 (1)	6 (1)	7 (1)
3	8	9 (2)				
4	16	27 (3)	16 (2)	25 (2)		
5	32				36 (2)	49 (2)
6	64	8,10E+01 (4)	6,40E+01 (3)	1,25E+02 (3)		
7	128	2,43E+02 (5)			2,16E+02 (3)	
8	256		2,56E+02 (4)			3,43E+02 (3)
9	512	7,29E+02 (6)		6,25E+02 (4)		
10	1024		1,02E+03 (5)		1,30E+03 (4)	
11	2048	2,19E+03 (7)		3,13E+03 (5)		2,40E+03 (4)
12	4096	6,56E+03 (8)	4,10E+03 (6)		7,78E+03 (5)	
13	8192			1,56E+04 (6)		
14	1,64E+04	1,97E+04 (9)	1,64E+04 (7)			1,68E+04 (5)
15	3,28E+04	5,90E+04 (10)			4,67E+04 (6)	
16	6,55E+04		6,55E+04 (8)	7,81E+04 (7)		1,18E+05 (6)
17	1,31E+05	1,77E+05 (11)				
18	2,62E+05		2,62E+05 (9)	3,91E+05 (8)	2,80E+05 (7)	
19	5,24E+05	5,31E+05 (12)				8,24E+05 (7)
20	1,05E+06	1,59E+06 (13)	1,05E+06 (10)	1,95E+06 (9)	1,68E+06 (8)	
21	2,10E+06					
22	4,19E+06	4,78E+06 (14)	4,19E+06 (11)			5,76E+06 (8)
23	8,39E+06	1,43E+07 (15)		9,77E+06 (10)	1,01E+07 (9)	
24	1,68E+07		1,68E+07 (12)			
25	3,36E+07	4,30E+07 (16)		4,88E+07 (11)	6,05E+07 (10)	4,04E+07 (9)
26	6,71E+07	1,29E+08 (17)	6,71E+07 (13)			
27	1,34E+08			2,44E+08 (12)		
28	2,68E+08	3,87E+08 (18)	2,68E+08 (14)		3,63E+08 (11)	2,82E+08 (10)
29	5,37E+08					
30	1,07E+09	1,16E+09 (19)	1,07E+09 (15)	1,22E+09 (13)		1,98E+09 (11)
31	2,15E+09	3,49E+09 (20)			2,18E+09 (12)	
32	4,29E+09		4,29E+09 (16)	6,10E+09 (14)		
33	8,59E+09	1,05E+10 (21)			1,31E+10 (13)	1,38E+10 (12)
34	1,72E+10	3,14E+10 (22)	1,72E+10 (17)	3,05E+10 (15)		
35	3,44E+10					
36	6,87E+10	9,41E+10 (23)	6,87E+10 (18)		7,84E+10 (14)	9,69E+10 (13)
37	1,37E+11			1,53E+11 (16)		
38	2,75E+11	2,82E+11 (24)	2,75E+11 (19)		4,70E+11 (15)	
39	5,50E+11	8,47E+11 (25)		7,63E+11 (17)		6,78E+11 (14)
40	1,10E+12		1,10E+12 (20)			
41	2,20E+12	2,54E+12 (26)		3,81E+12 (18)	2,82E+12 (16)	
42	4,40E+12	7,63E+12 (27)	4,40E+12 (21)			4,75E+12 (15)
43	8,80E+12				1,69E+13 (17)	
44	1,76E+13	2,29E+13 (28)	1,76E+13 (22)	1,91E+13 (19)		3,32E+13 (16)
45	3,52E+13	6,86E+13 (29)				
46	7,04E+13		7,04E+13 (23)	9,54E+13 (20)	1,02E+14 (18)	
47	1,41E+14	2,06E+14 (30)				2,33E+14 (17)
48	2,81E+14		2,81E+14 (24)	4,77E+14 (21)		
49	5,63E+14	6,18E+14 (31)			6,09E+14 (19)	
50	1,13E+15	1,85E+15 (32)	1,13E+15 (25)			1,63E+15 (18)
51	2,25E+15			2,38E+15 (22)	3,66E+15 (20)	
52	4,50E+15	5,56E+15 (33)	4,50E+15 (26)			
53	9,01E+15	1,67E+16 (34)		1,19E+16 (23)		1,14E+16 (19)
54	1,80E+16		1,80E+16 (27)		2,19E+16 (21)	
55	3,60E+16	5,00E+16 (35)		5,96E+16 (24)		
56	7,21E+16	1,50E+17 (36)	7,21E+16 (28)	2,98E+17 (25)	1,32E+17 (22)	7,98E+16 (20)
Mapning (56):		(56/36) 1,56	(56/28) 2	(56/25) 2,24	(56/22) 2,55	(56/20) 2,8

# bits	# kom.	8 polet #	9 polet #	10 polet #	11 polet #	12 polet #
1	2					
2	4					
3	8	8 (1)	9 (1)	10 (1)	11 (1)	12 (1)
4	16					
5	32					
6	64	6,40E+01 (2)	8,10E+01 (2)	1,00E+02 (2)		
7	128				1,21E+02 (2)	1,44E+02 (2)
8	256					
9	512	5,12E+02 (3)	7,29E+02 (3)	1,00E+03 (3)		
10	1024				1,33E+03 (3)	1,73E+03 (3)
11	2048					
12	4096	4,10E+03 (4)	6,56E+03 (4)			
13	8192			1,00E+04 (4)	1,46E+04 (4)	
14	1,64E+04					2,07E+04 (4)
15	3,28E+04	3,28E+04 (5)	5,90E+04 (5)			
16	6,55E+04			1,00E+05 (5)		
17	1,31E+05				1,61E+05 (5)	2,49E+05 (5)
18	2,62E+05	2,62E+05 (6)				
19	5,24E+05		5,31E+05 (6)	1,00E+06		
20	1,05E+06			(6)	1,77E+06 (6)	
21	2,10E+06	2,10E+06 (7)				2,99E+06 (6)
22	4,19E+06		4,78E+06 (7)			
23	8,39E+06			1,00E+07 (7)		
24	1,68E+07	1,68E+07 (8)			1,95E+07 (7)	
25	3,36E+07		4,30E+07 (8)			3,58E+07 (7)
26	6,71E+07			1,00E+08 (8)		
27	1,34E+08	1,34E+08 (9)			2,14E+08 (8)	
28	2,68E+08		3,87E+08 (9)			4,30E+08 (8)
29	5,37E+08			1,00E+09 (9)		
30	1,07E+09	1,07E+09 (10)				
31	2,15E+09		3,49E+09 (10)		2,36E+09 (9)	
32	4,29E+09					5,16E+09 (9)
33	8,59E+09	8,59E+09 (11)		1,00E+10 (10)		
34	1,72E+10		3,14E+10 (11)		2,59E+10 (10)	
35	3,44E+10					6,19E+10 (10)
36	6,87E+10	6,87E+10 (12)		1,00E+11 (11)		
37	1,37E+11				2,85E+11 (11)	
38	2,75E+11		2,82E+11 (12)			
39	5,50E+11	5,50E+11 (13)		1,00E+12 (12)		7,43E+11 (11)
40	1,10E+12					
41	2,20E+12		2,54E+12 (13)		3,14E+12 (12)	
42	4,40E+12	4,40E+12 (14)				
43	8,80E+12			1,00E+13 (13)		8,92E+12 (12)
44	1,76E+13		2,29E+13 (14)		3,45E+13 (13)	
45	3,52E+13	3,52E+13 (15)				
46	7,04E+13			1,00E+14 (14)		1,07E+14 (13)
47	1,41E+14		2,06E+14 (15)			
48	2,81E+14	2,81E+14 (16)			3,80E+14 (14)	
49	5,63E+14			1,00E+15 (15)		
50	1,13E+15		1,85E+15 (16)			1,28E+15 (14)
51	2,25E+15	2,25E+15 (17)			4,18E+15 (15)	
52	4,50E+15					
53	9,01E+15		1,67E+16 (17)	1,00E+16 (16)		1,54E+16 (15)
54	1,80E+16	1,80E+16 (18)				
55	3,60E+16				4,59E+16 (16)	
56	7,21E+16	1,44E+17 (19)	1,50E+17 (18)	1,00E+17 (17)	5,05E+17 (17)	1,85E+17 (16)
Mapning (56):		(56/19) 2,95	(56/18) 3,11	(56/17) 3,29	(56/17) 3,29	(56/16) 3,5

# bits	# kom.	13 polet #	14 polet #	15 polet #	16 polet #
1	2				
2	4				
3	8	13 (1)	14 (1)	15 (1)	
4	16				16 (1)
5	32				
6	64				
7	128	1,69E+02 (2)	1,96E+02 (2)	2,25E+02 (2)	
8	256				2,56E+02 (2)
9	512				
10	1024				
11	2048	2,20E+03 (3)	2,74E+03 (3)	3,38E+03 (3)	
12	4096				4,10E+03 (3)
13	8192				
14	1,64E+04	2,86E+04 (4)			
15	3,28E+04		3,84E+04 (4)	5,06E+04 (4)	
16	6,55E+04				6,55E+04 (4)
17	1,31E+05				
18	2,62E+05	3,71E+05 (5)			
19	5,24E+05		5,38E+05 (5)	7,59E+05 (5)	
20	1,05E+06				1,05E+06 (5)
21	2,10E+06				
22	4,19E+06	4,83E+06 (6)	7,53E+06 (6)		
23	8,39E+06			1,14E+07 (6)	
24	1,68E+07				1,68E+07 (6)
25	3,36E+07	6,27E+07 (7)			
26	6,71E+07		1,05E+08 (7)		
27	1,34E+08			1,71E+08 (7)	
28	2,68E+08				2,68E+08 (7)
29	5,37E+08	8,16E+08 (8)			
30	1,07E+09		1,48E+09 (8)		
31	2,15E+09			2,56E+09 (8)	
32	4,29E+09				4,29E+09 (8)
33	8,59E+09	1,06E+10 (9)			
34	1,72E+10		2,07E+10 (9)		
35	3,44E+10			3,84E+10 (9)	
36	6,87E+10				6,87E+10 (9)
37	1,37E+11	1,38E+11 (10)			
38	2,75E+11		2,89E+11 (10)		
39	5,50E+11			5,77E+11 (10)	
40	1,10E+12	1,79E+12 (11)			1,10E+12 (10)
41	2,20E+12		4,05E+12 (11)		
42	4,40E+12			8,65E+12 (11)	
43	8,80E+12				
44	1,76E+13	2,33E+13 (12)			1,76E+13 (11)
45	3,52E+13		5,67E+13 (12)		
46	7,04E+13			1,30E+14 (12)	
47	1,41E+14				
48	2,81E+14	3,03E+14 (13)			2,81E+14 (12)
49	5,63E+14		7,94E+14 (13)		
50	1,13E+15			1,95E+15 (13)	
51	2,25E+15	3,94E+15 (14)			
52	4,50E+15				4,50E+15 (13)
53	9,01E+15		1,11E+16 (14)		
54	1,80E+16			2,92E+16 (14)	
55	3,60E+16	5,12E+16 (15)			
56	7,21E+16	6,65E+17 (16)	1,56E+17 (15)	4,38E+17 (15)	7,21E+16 (14)
Mapning (56):		(56/16) 3,5	(56/15) 3,73	(56/15) 3,73	(56/36) 4

Eksempel på anvendelsen af mapningstabellen:

Vi antager, at have en nøglelængde på 34 bits, at vi bruger en komponent som kan sættes i 6 stillinger, og vil finde ud af om brugeren skal stille komponent end med en komponent som sættes i 5 stillinger. Vi finder 34 i den første søjle og går ud til søjleparret som dækker over 6 poler. Her finder vi ingen værdi og må derfor gå nedad, indtil vi finder en værdi. Værdien vi finder er på $7{,}84*10^{10}$ kombinationer, når der foretages 14 handlinger. Ved 13 handlinger ville vi kun kunne dække over $1{,}31*10^{10}$ nøgler, hvilket er en restriktion på nøglerummet og det ønsker vi ikke. Så svaret er ja, brugeren skal foretage færre handlinger, hvis vi udskifter komponenten med en type som kan sættes i 6 stillinger. Hvis vi går tilbage igen fra den fundne værdi til første søjle i tabellen, så ses det, at brugeren ikke skal foretage flere handlinger, hvis nøglelængden forøges fra 34 til 36 bits.

# bits	# kom.	3 polet	#	4 polet	#	5 polet	#	6 polet	#
...	...	...	...	...	...	...	...	...	...
32	4,29e9			4,29e9	(16)	6,10e9	(14)		
33	8,59e9	1,05e10	(21)					1,31e10	(13)
34	1,72e10	3,14e10	(22)	1,72e10	(17)	3,05e10	(15)		
35	3,44e10								
36	6,87e10	9,41e10	(23)	6,87e10	(18)			7,84e10	(14)
37	1,37e11					1,53e11	(16)		
...	...	...	...	...	...	...	...	...	...

Bilag E – Easy-of-use test med nogle af løsningsforslagene til initial nøgleudveksling

Farvet Lysdioder:

Vi har lavet forsøg med farvet lysdioder. Det vi gjorde var, at en af os spillede kontroller mens den anden spillede node. Når noden viser en farvet blinkesekvens, så viser personen den farve vha. et farvepapir til den anden person, som indtastes dette på kontrolleren og næste blinkesekvens fås ved at trykke på knappen på noden. Det er altså personen som spiller node, der bestemmer de forskellige blinkesekvenser (Dette skal bestemmes før man går i gang med forsøget). Den anden taster ind på computeren hvad han registrerer, som gøres ved at sætte farver på nogle af tasterne det på tastaturet. Vi har åbnet et tekst dokument og givet farven rød til tast 1, farven grøn til tast 2 osv.). Bagefter kan man sammenligne om de tastede resultater passer med de forudbestemt resultater, og dermed afgøre hvor mange fejl der blev lavet.

Vi har først prøvet med to forskellige farver (Rød = 1 og Grøn = 2), dvs. 56 tastetryk. Hvor en af os viste farve kombinationerne til den anden, som indtaster.

Forsøg1:

Rigtige værdier =
11221212111221212222211111222212121212111111222222121 11
Tastede værdier =
112212121112212122222211111222212121212111111222222221 11
Tid: 1m14s
Resultat: 1 FEJL

Forsøg2:

Rigtige værdier =
1112122222211111121212121222211111222222121221112121221 1
Tastede værdier =
1112122222211111121212121222211111222222121221112121221 1
Tid: 1m10s
Resultat: Ingen FEJL

Forsøg3:

Rigtige værdier =

122122111222112121221111212212211221212112211121221221 12

Tastede værdier =

122122111222112121221111212212211211212111211121221221 12

Tid: 1m1s

Resultat: 2 FEJL

Forsøg4:

Rigtige værdier =

211221221211122112121221122122121111221212112221112212 21

Tastede værdier =

211221221211121112121211221221212112212121122211122122 1

Tid: 51s

Resultat: 3 FEJL

Forsøg5:

Rigtige værdier:

122122111222112121221111212212211221212112211121221221 12

Tastede værdier:

122122111222112121221111212212211221212112211121221221 12

Tid: 2m34s

Resultat: Ingen FEJL

Så prøvede vi med tre forskellige farver (Rød = 1, Grøn = 2, Gul = 3).

Forsøg1:

Rigtige værdier =

13221133222132213313222223131221232 1

Tastede værdier =

13221133222132213313222223131221332 1

Tid: 56s

Resultat: 1 FEJL

Forsøg2:

Rigtige værdier =

12321221313222223133122312223311223 1

Tastede værdier =

123212213132222231311221122233112231

Tid: 53s

Resultat: 2 FEJL

Forsøg3:

Rigtige værdier =

123113223312111233112313223332231 32

Tastede værdier =

132113223312111233113313223332231 32

Tid: 1m5s

Resultat: 3 FEJL

Forsøg4:

Rigtige værdier =

231322333223132113321111213322311321

Tastede værdier =

321322333223132113321111233322311321

Tid: 1m1s

Resultat: 3 FEJL

Til sidst prøvede vi med fire forskellige farver (Rød = 1, Grøn = 2, Gul = 3, og Blå = 4).

Forsøg1:

Rigtige værdier =

41323314321442211324144423 23

Tastede værdier =

41323314321442211324144423 23

Tid: 56s

Resultat: Ingen FEJL

Forsøg2:

Rigtige værdier =

32324441423112244123413323l4

Tastede værdier =

32324441423112244123413323l4

Tid: 48s

Resultat: Ingen FEJL

Forsøg3:

Rigtige værdier =

14322311242423114243442333l2

Tastede værdier =

12322311242423114243442333l2

Tid: 55s

Resultat: 1 FEJL

Forsøg4:

Rigtige værdier =

21333244342411324242113223l1

Tastede værdier =

21333244342411324342113223l1

Tid: 52s

Resultat: 1 FEJL

7-segment:

Her bruges samme procedure. Det viste tal på 7-segmentet på noden vises til personen, som holder øje med kontrolleren så han hurtig kan taste dette nummer ind på kontrolleren. Så snart personen har indtastet tallet på kontrolleren, skal han trykke på knappen på noden så der bliver vist et nyt tal. Dette kan udføres ved at åbne et tekst dokument, og tallene indtastes der, så man bagefter kan kontrollere om der er blevet lavet fejl undervejs. (Dette forsøg er udført vha. et tekst dokument ligesom foregående forsøg.

Forsøg med et 7-segment (17 tastetryk):

Forsøg1:

Tastede værdier: 25790157324881764

Rigtige værdier: 25790157324881764

Tid: 52s

Resultat: Ingen FEJL

Forsøg2:

Tastede værdier: 38925147268091350

Rigtige værdier: 38925147268091350

Tid: 40s

Resultat: Ingen FEJL

Forsøg3:

Tastede værdier: 19782315408761532

Rigtige værdier: 19782315408761532

Tid: 36s

Resultat: Ingen FEJL

Forsøg med to 7-segment (9 tastetryk):

Forsøg1:

Tastede værdier: 14 28 35 54 85 92 01 51 67

Rigtige værdier: 14 28 35 54 85 92 01 51 67

Tid: 40s

Resultat: Ingen FEJL

Forsøg2:

Tastede værdier: 90 80 57 61 15 23 32 40 41

Rigtige værdier: 90 80 57 61 15 23 32 40 41

Tid: 32s

Resultat: Ingen FEJL

Forsøg3:

Tastede værdier: 01 73 48 41 50 24 10 09 07

Rigtige værdier: 01 73 48 41 50 24 10 09 07

Tid: 35s

Resultat: Ingen FEJL

Tastatur:

En af os spiller kontroller, som bestemmer nøglen og viser den som tal/bogstav. Det viste tal på displayet vises videre til personen, som tasterdette nummer på tastaturen på noden. Så snart personen har indtastet tallet på kontrolleren, skal han trykke på knappen på kontrolleren, så der bliver vist et nyt tal. Dette kan også udføres vha. computer, hvor man åbner et tekst dokument og taster de forskellige karakter. Derefter kan man samenligne de indtastede værdier med de rigtige værdier.

Der er lavet forsøg med 4 forskellig karakter (tal, 28 tastetryk):

Forsøg1:

Tastede værdier: 134121442131113321321441422 3

Rigtige værdier: 134121442131113321321441422 3

Tid: 48s

Resultat: Ingen FEJL

Forsøg2:

Tastede værdier: 322414412312331113124412143 1

Rigtige værdier: 322414412312331113124412143 1

Tid: 49s

Resultat: Ingen FEJL

Der er lavet forsøg med 12 forskellig karakter (Bogstaver, 16 tastetryk):

Forsøg1:

Tastede værdier: ACDLBFJGHLKIJLEB

Rigtige værdier: ACDLBFJGHLKIJLEB

Tid: 45s

Resultat: Ingen FEJL

Forsøg2:

Tastede værdier: CEJBKIGHLFJEDAGF

Rigtige værdier: CEJBKIGHLFJEDAGF

Tid: 41s

Resultat: Ingen FEJL

Der er lavet forsøg med 16 forskellig karakter (både tal og bogstaver, 14 tastetryk): Her er det svært at gennemskue de viste karakter, om det er tal eller bogstaver, da 0, O, B, 8, G og 6 kan forstås som et tal eller bogstav (nul eller o, B eller 8, G eller 6).

Forsøg1:

Tastede værdier: A1C7KE0E849FJL

Rigtige værdier: A1C7KE0E849FJL

Tid: 30s

Resultat: Ingen FEJL

Forsøg2:

Tastede værdier: 50E1IALFK091JJ

Rigtige værdier: 50E1IALFK091JJ

Tid: 28s

Resultat: Ingen FEJL

Bilag F – Oversigt over løsningsforslagene til initial nøgleudveksling

Oversigten viser en indikation af løsningens pris, ease-of-use, tidsforbrug der bruges til inkludering samt hvor mange I/O det kræver af ASIC' en. Farvekodning er som følger:

- Pris – prisen for hele løsningen ligger i intervallet:
  - : 1,00 – 5,00 kr.
  - : 5,01 – 10,00 kr.
  - : 10,01 – 15,00 kr.
  - : Over 15,00 kr.
- Ease-of-use – hvor nemt er det at inkludere en ny node til netværket:
  - : Meget nemt
  - : Nemt
  - : Svært
  - : Meget svært
- Tidsforbrug – hvor lang tid tager selve inkluderingen:
  - : Meget stort
  - : Stort
  - : Lavt
  - : Meget lavt
- Hardware – hvor stort et antal I/O skal der bruges på ASIC'en:
  - : Meget stort
  - : Stort
  - : Lavt
  - : Meget lavt

Prisen er hentet ud fra forhandlernes hjemmeside og hvad ang. harware "forbruget" så stammer det fra datablade. Ease-of-use samt tidsforbrug er opgjort på baggrund af eksperimentelle forsøg.

Løsning:	Pris	Ease-of-use	Tidsforbrug	Hardware
SIL 1				
SIL 2				
SIL 3				
SIL 4				
DIL 5				
DIL 6				
DIL 7				
DIL 8				
DIL 9				
DIL 10				
Trykknap 1				
Trykknap 2				
Trykknap 3				
Trykknap 4				
Drejeomskifter 1				
Drejeomskifter 2				
Drejeomskifter 3				
Drejeomskifter 4				
Trimmer 1				
Trimmer 3				
Trimmer 5				
Trimmer 7				
Trimmer 9				
Trimmer 11				
Trimmer 13				
Trimmer 2				
Trimmer 4				
Trimmer 6				
Trimmer 8				
Trimmer 10				
Trimmer 12				
Trimmer 14				
Farvet lysdioder 1				
Farvet lysdioder 2				
Farvet lysdioder 3				
Farvet lysdioder 4				
Farvet lysdioder 5				
Farvet lysdioder 6				
Farvet lysdioder 7				
Farvet lysdioder 8				
Farvet lysdioder 9				
7-segment 1				
7-segment 2				
7-segment 3				
7-segment 4				
Tastatur 1				
Tastatur 2				
Tastatur 3				
Smart Card				
Magnet kort				
IR				
Fingeraftryk				
Stregkode				

Table 33: Oversigt over løsninger til initial nøgleudveksling med indikation af pris, ease-of-use, tidsforbrug og hardware omkostninger på ASIC

Nedenstående tabel er bla. brugt til at lave ovenstående oversigt:

#	Løsningsforslag	Mapning	Ease-of-use	Pris	Mapning/Pris	EoU/pris	Tidsforbrug	Hardware krav
1	SIL 1	2	1	8,4	0,24	0,12	1	4
2	SIL 2	4	1	10	0,40	0,10	1	3
3	SIL 3	5,6	1	8,97	0,62	0,11	2	2
4	SIL 4	8	1	9,38	0,85	0,11	2	1
5	DIL 5	2	1	4,55	0,44	0,22	2	4
6	DIL 6	4	2	3,59	1,11	0,56	2	3
7	DIL 7	6	2	4,28	1,40	0,47	3	2
8	DIL 8	8	3	6,76	1,18	0,44	3	2
9	DIL 9	9,3	3	5,52	1,68	0,54	3	1
10	DIL 10	11,2	4	6,49	1,73	0,62	3	1
11	Trykknap 1	1	2	2,48	0,40	0,81	2	4
12	Trykknap 2	2	2	4,96	0,40	0,40	2	4
13	Trykknap 3	3	1	7,44	0,40	0,13	3	3
14	Trykknap 4	4	1	9,92	0,40	0,10	3	3
15	Drejeomskifter 1	1	3	8,97	0,11	0,33	2	4
16	Drejeomskifter 2	1,56	3	8,97	0,17	0,33	3	4
17	Drejeomskifter 3	3,29	1	14,63	0,22	0,07	3	4
18	Drejeomskifter 4	4	1	14,9	0,27	0,07	2	4
19	Trimmer 1	1	3	1,7	0,59	1,76	1	4
20	Trimmer 3	1,56	3	1,7	0,92	1,76	1	4
21	Trimmer 5	2	3	1,7	1,18	1,76	2	4
22	Trimmer 7	2,24	2	1,7	1,32	1,18	3	4
23	Trimmer 9	2,55	1	1,7	1,50	0,59	2	4
24	Trimmer 11	2,8	1	1,7	1,65	0,59	1	4
25	Trimmer 13	2,95	1	1,7	1,74	0,59	1	4
26	Trimmer 2	1	3	4,8	0,21	0,63	1	4
27	Trimmer 4	1,56	3	4,8	0,33	0,63	1	4
28	Trimmer 6	2	3	4,8	0,42	0,63	2	4
29	Trimmer 8	2,24	3	4,8	0,47	0,63	3	4
30	Trimmer 10	2,55	3	4,8	0,53	0,63	3	4
31	Trimmer 12	2,8	2	4,8	0,58	0,42	4	4
32	Trimmer 14	2,95	2	4,8	0,61	0,42	3	4
33	Farvet lysdioder 1	1	4	0,39	2,56	10,26	2	4
34	Farvet lysdioder 2	1	4	0,966	1,04	4,14	2	4
35	Farvet lysdioder 3	1,33	3	0,828	1,61	3,62	3	3
36	Farvet lysdioder 4	1,47	3	0,828	1,78	3,62	3	3
37	Farvet lysdioder 5	1,33	3	0,828	1,61	3,62	3	3
38	Farvet lysdioder 6	1,4	2	0,828	1,69	2,42	3	3
39	Farvet lysdioder 7	2	3	20,56	0,10	0,15	3	1
40	Farvet lysdioder 8	2	2	20,56	0,10	0,10	3	1
41	Farvet lysdioder 9	2,5	2	20,56	0,12	0,10	3	1
42	7-segment 1	3,29	4	8,83	0,37	0,45	3	2
43	7-segment 2	4	2	8,83	0,45	0,23	3	1
44	7-segment 3	6,22	4	17,66	0,35	0,23	3	2
45	7-segment 4	8	2	17,66	0,45	0,11	3	1
46	Tastatur 1	2	4	118	0,02	0,03	3	2
47	Tastatur 2	3	3	45,95	0,07	0,07	3	1
48	Tastatur 3	4	3	57,55	0,07	0,05	3	1
49	Samrt Card	x	4	24,56	x	x	4	3
50	Magnetkort	x	4	70	x	x	4	3
51	IR	x	4	6,69	x	x	4	4
52	Fingeraftryk	x	4	x	x	x	4	3
53	Stregkode	x	4	x	x	x	4	3

7.0 Figuroversigt

8.0 Tabeloversigt

9.0 Litteraturliste

1. Artikel/Homepage: http://www.lk.dk/public/s_gen1.asp?what=tekst&sideid=192 Intelligente Huse
2. Artikel/Homepage: http://www.smarthome.com/secvehicle.html Smart Home
3. Artikel/Homepage: http://www.computerworld.dk/default.asp?Mode=2&ArticleID=23946 Intelligent Huse
4. Artikel/Homepage: http://libra.unitbv.ro/internet/OSI%20model.htm ISO
5. Artikel/Homepage: http://www.iso.org ISO: International Organization for Standardization
7. Artikel/Homepage: http://www.netgear.com/docs/refdocs/Wireless/wirelessBasics.htm Wireless Networking Basics
8. BOG: *Wireless Communications and Networks* by William Stallings, Prentice Hall, October 2001, ISBN 0-13-040864-6
9. Artikel/Homepage: http://www.hw.cz/english/docs/irda/irda.html IRDA
10. Artikel/Homepage: http://www.ieee.org/portal/index.jsp 802.11x
11. BOG: *802.11 Wireless* Networks, The Definitive Guide, Matthew S. Gast, ISBN: 0-596-00183-5
12. Artikel/Homepage: http://grouper.ieee.org/groups/802/11/ 802.11x
13. Artikel/Homepage: http://www.wifialliance.com/OpenSection/protected_access.asp WiFi/WPA
14. Artikel/Homepage: http://en.wikipedia.org/wiki/ISM_band ISM
15. Artikel/Homepage: http://www.webopedia.com/TERM/p/piconet.html Bluetooth
16. Artikel/Homepage: http://www.webopedia.com/TERM/S/scatternet.html Bluetooth
17. Artikel/Homepage: http://www.aimglobal.org/technologies/rfid/what_is_rfid.asp RFID
18. Artikel/Homepage: http://www.webopedia.com/TERM/m/mesh.html Mesh netværk
19. Artikel/Homepage: http://www2.rad.com/networks/1994/err_con/crc.htm CRM
20. Artikel/Homepage: http://ftp.arl.mil/~mike/ping.html PING
21. Artikel/Homepage: http://www.kingfisher.com.au/appnotes/A01.htm Decibel
22. BOG: *Kryptologi – fra viden til videnskab* by Peter Landrock og Knud Nissen. ABACUS, 1997.
23. BOG: *Cryptography, theory and practice* by Dougles R. Stinson, CRC Press 2002

24. BOG: *Applied Cryptography Protocols, Algorithms, and Source Code in C* by Bruce SCHNEIER, second edition, ISBN 0-471-12845-7

25. Artikel/Homepage: http://www.iscit.surfnet.nl/team/Erik/masterth/authenti.htm Autentifikation

26. Artikel/Homepage: http://www.cs.wm.edu/~hnw/courses/cs420/slides/ch13.pdf Autentifikation

27. BOG: *Handbook of Applied Cryptography, by* Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone. CRC Press, ISBN: 0-8493-8523-7, October 1996

28. Artikel/Homepage: http://www.rsasecurity.com/ RSA

29. Artikel/Homepage: http://catb.org/~esr/jargon/html/C/cracker.html Angreb

30. Artikel/Homepage: http://www.rsonline.dk

31. Artikel/Homepage: http://www.farnellinone.dk

32. Artikel/Homepage: http://www.cryptico.com/Files/filer/produktoversigt.pdf

33. Artikel/Homepage: http://www.visonic.com/

34. Artikel/Homepage: http://www.quard.dk/

35. Artikel/Homepage: http://ask.yahoo.com/ask/20030630.html Bar Code

36. Artikel/Homepage: http://www.barcode-1.com/pub/russadam/upccode.html Universal Product Code

37. Artikel/Homepage: http://www.barcodehq.com/primer.html Bar Code

38. Artikel/Homepage: http://www.symbol.com/products/oem/cse600.html Bar Code

39. Artikel/Homepage: http://www.magtek.com/documentation/public/99800004-1.pdf Magnetkort

40. Artikel/Homepage: http://www.chez.com/mosfet/cread2.txt Magnetkort

41. Artikel/Homepage: http://www.adcomdata.dk Magnetkort

42. Artikel/Homepage: http://www.encyclopedia.com/html/u1/ultrason.asp Ultralyd

43. Artikel/Homepage: http://ej.rsna.org/ej3/0079-98.fin/doppler.htm Ultralyd

44. Artikel/Homepage: http://www.almaden.ibm.com/cs/user/pan/pan.html PAN

45. Artikel/Homepage: *Personal Area Networks: Near-field intrabody communication,* Thomas G. Zimmerman, IBM Systems Journal, Vol 35, NOS 3&4, 1996

46. Artikel/Homepage: http://www.packetstormsecurity.org/mag/crypto-gram/crypto-gram-9812.html DES

47. Artikel/Homepage: www.sans.org/rr/papers/20/726.pdf DES

48. Artikel/Homepage: http://www.ntp.org/ NTP

49. Artikel/Homepage: http://www.ntp.org/ntpfaq/NTP-s-algo.htm#Q-ACCURATE-CLOCK

50. Artikel/Homepage: http://www.computerworld.com/mobiletopics/mobile/story/0,10801,89495,00.html?nas=MW-89495 Bluetooth

51. Artikel/Homepage: http://www.aimglobal.org/technologies/rfid/resources/RFIDCharacteristics.pdf RFID

52. Artikel/Homepage: http://www.ieee.org/portal/index.jsp?pageID=corp_level1&path=about/802std&file=index.xml&xsl=generic.xsl#802_11gen 802.11x

53. Artikel/Homepage: Z-Wave Protocol Overview, May 5, 2003

54. Artikel: *Bluetooth and its inherent security issues* by Tu C. Niem, SANS GIAC Security Essentials Certification (GSEC) v1.4b, 11/04/2002

55. Artikel: *Bluetooth Security* by Juha T. Vainio, Department of Computer Science and Engineering, Helsinki University of Technology

56.Artikel: Tea extensions by Roger M. Needham and David J. Wheeler, October 1996

57. Artikel: PC IR Remote Control Hardware, http://www.ee.washington.edu/circuit_archive/circuits/PCIR/pcirhw.html

58. Artikel: RS-232 Laser Transciever http://www.geocities.com/SiliconValley/Lakes/7156/laser.htm

59. Artikel: Secure Hash Standard http://www.itl.nist.gov/fipspubs/fip180-1.htm#FIPS_TOP

60. Artikel: Simple PC smartcard reader http://www.epanorama.net/documents/smartcard/smartcard_reader.html

61. Artikel: *The Tiny Encryption Algorithm (TEA)* by Simon Shepherd, Bradford University, England

62. Artikel: *Universal IR Controller for a PC*, Electronics Australia http://www.geocities.com/SiliconValley/Lakes/7156/

63. Artikel: *Security Services Specification* Release 0.76 by ZigBee www.zigbee.org

64. Artikel: *Personal Area Network* by Nitesh Ambastha, Mumbai, April 2000 http://www.acm.org/chapters/bombay/news/archives/200004.html#01

65. Artikel: *A Broad-band Intrabody Communication System with Electro-Optic Probe* by Masaaki Fukomoto, Mitsuru Shinagawa, Toshiaki Sugimura

66. Artikel: *Body Coupled Fingering: Wireless Wearable Keyboard* http://www.acm.org/sigchi/chi97/proceedings/paper/fkm.htm#U25

67. Artikel: *A Survey on Sensor Networks* by Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubrmaniam and Erdal Cayirci, Georgia Institute of Technology

68. Artikel: On Communication Security in Wireless Ad-Hoc Sensor Networks by Sasha Slijepcevic, Miodrag Potkonjak, Vlasios Tsiatsis, Scott Zimbeck, Mani B. Srivastava, UCLA. www.ucla.edu

69. Artikel: On Preventing Replay Attacks on Security Protocols www.cs.uidaho.edu/~jimaf/docs/**replay**02.pdf

70. Artikel: A Taxonomy of Replay Attacks http://chacs.nrl.navy.mil/publications/CHACS/1994/1994syverson-foundations.pdf