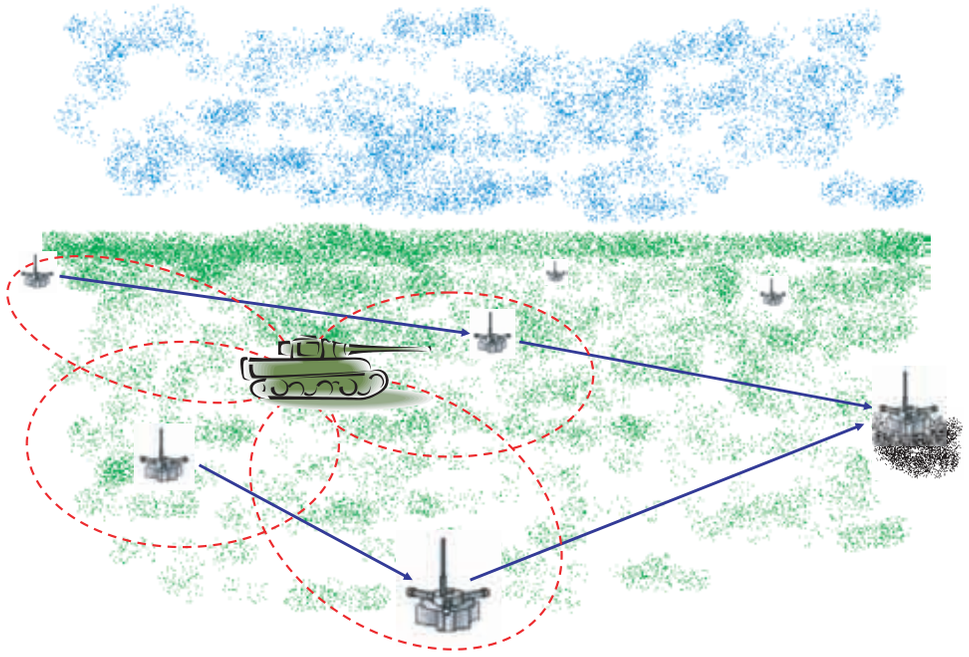
Säkerhet i trådlösa energibegränsade marksensornät

LINDA FARMAN, ULF STERNER, LARS WESTERDAHL
OCH PELLE ZEIJLON



FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1350 anställda varav ungefär 950 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömning av olika typer av hot, system för ledning och hantering av kriser, skydd mot och hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.

FOI
Totalförsvarets forskningsinstitut
Ledningssystem
Box 1165
581 11 Linköping

Tel: 013-37 80 00
Fax: 013-37 81 00

www.foi.se

FOI-R--1912--SE
ISSN 1650-1942

Teknisk rapport
Januari 2006

Ledningssystem

Säkerhet i trådlösa marksensornät med fokus på energi

FOI-R–1912–SE
ISSN 1650-1942

Ledningssystem
Teknisk rapport

Januari 2006

Utgivare Totalförsvarets Forskningsinstitut Ledningssystem Box 1165 SE-581 11 LINKÖPING	**Rapportnummer, ISRN** FOI-R–1912–SE	**Klassificering** Teknisk rapport
	Forskningsområde 4. Ledning, informationsteknik och sensorer	
	Månad, år Januari 2006	**Projektnummer** E793055
	Delområde 41. Ledning med samband och telekom och IT-system	
	Delområde 2	
Författare Linda Farman, Ulf Sterner, Lars Westerdahl och Pelle Zeijlon	**Projektledare** Linda Farman	
	Godkänd av Sören Eriksson	
	Uppdragsgivare/kundbeteckning FMV- Försvarets materielverk	
	Teknisk och/eller vetenskapligt ansvarig Peter Stenumgaard	

Rapportens titel

Säkerhet i trådlösa marksensornät med fokus på energi

Sammanfattning

En viktig utmaning för säkerheten i sensornät är nodernas begränsade energi- och beräkningskapacitet. En annan viktig utmaning för säkerheten är att routingen är distribuerad. Det är viktigt att beakta dessa aspekter vid utformningen av säkerheten i sensornätet och att de kan utnyttjas av illasinnade och användas för att förstöra sensornätets funktioner.

Autentisering, dvs. en känd säkerhetslösning, av noderna eller att hantera säkerheten på rotuingnivå med hjälp av multipla vägar studeras som två metoder för att förbättra säkerhetsegenskaperna hos sensornätet. En viktig del är även att undersöka vad respektive metod innebär för resursförbrukningen. Den attack som studeras är en routingattack av typen ett svart hål. En utvärderingsparameter är bl a hur datafusionen i nätet påverkas av ett svart hål.

Resultaten visar på att autentisering är ett effektivt skydd mot ett svart hål, men ger en ökad resursförbrukning. Att skydda sig mot ett svart hål med hjälp av multipla vägar däremot visar sig inte vara effektivt.

Nyckelord

trådlösa sensornät, IT-säkerhet, autentisering, svart hål, AODV med multipla vägar, energi, datafusion

Övriga bibliografiska uppgifter	**Språk** Svenska
ISSN	**Antal sidor:** 52 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Issuing organization Swedish Defence Research Agency Command and Control Systems P.O. Box 1165 SE-581 11 LINKÖPING SWEDEN	**Report number, ISRN** FOI-R–1912–SE	**Report type** Technical report
	Programme areas 4. C^4ISTAR	
	Month year 2006-01-31	**Project No.** E793055
	Subcategories 41. C^4I	
	Subcategories 2	

Author/s Linda Farman, Ulf Sterner, Lars Westerdahl and Pelle Zeijlon	**Project manager** Linda Farman
	Approved by Sören Eriksson
	Sponsoring agency FMV- Defence Material Administration
	Scientifically and technically responsible Peter Stenumgaard

Report title

Security in wireless sensor networks with focus on energy

Abstract

An important challenge for wireless sensor networks are the nodes limited energy- and calculation resources. Further, another important challenge for security is the distributed routing. When designing the security for the sensor network it is therefore important to consider these aspects and that someone may use them to destroy the functions in the sensor network.

Authentication, i.e. a well known security solution, or handle the security at the routing layer with multiple routes (multipath) are the two methods that are studied to improve the security in the sensor network. Further, the energy consumption for respectively method is also considered. The studied routing attack is called a black hole. One evaluation parameter is for example the effect on the datafusion in the network when a black hole is introduced.

The results show that authentication is an effective protection against a black hole, but gives an increased energy consumption. To use multipath on the other hand is not effective.

Keywords

wireless sensor networks, security, autenthication, routing attack, AODV-multipath, energy, datafusion

Further bibliographic information	**Language** Swedish
ISSN	**Pages** 52 p.
	Price acc. to pricelist

Innehåll

Kapitel 1

Introduktion

1.1 Bakgrund

Detta arbete är gjort inom ramen för samverkan LedSystT inom IPT NETS under perioden $2005 - 2006$ [10]. Arbetet utfört i denna studie ligger förankrat i en förstudie utförd under 2004 inom projektet [5].

Det finns ett stort behov inom försvaret att ha en korrekt och relevant lägesbild i olika sammanhang. Lägesbilden underlättar för användaren att ta beslut och utföra riktiga handlingar. För att uppnå detta är sensornät en mycket viktig komponent som informationskälla. Ett sensornät förmedlar en lägesbild till en användare i form av t ex information om vad som händer i ett geografiskt område. En stor fördel med sensornät är att användaren kan få ut informationen från området som bevakas utan att själv behöva befinna sig där. Detta kan t ex vara aktuellt ifall området är ett riskområde i något avseende.

Valet av typ av sensor beror på användningsområdet för sensornätet. Sensorerna kan t ex mäta ljud, temperatur, gas, vibrationer eller tryck. I vissa fall kan dessa olika typer av sensorer kombineras i ett sensornät. Forskningen inom sensornät motiverades från början av olika militära applikationer. Idag har dock sensornät många andra användningsområden. Ett är att upprätthålla säkerheten hos olika infrastrukturer som t ex att skydda kritiska byggnader mot eventuella terroristattacker. Andra områden där sensornät visat sig användbara är i industrin, i trafiken både i och utanför bilen, och för övervakning av naturområden.

1.2 Motivering

En grundläggande del i ett sensornät är fungerande, tillförlitlig och säker kommunikation för att sprida informationen mellan sensorerna och ut till andra nät. Eftersom kommunikationen i nätet bör vara trådlös, radiokommunikation, ökar risken för attacker. Risken för en fysisk attack är stor då sensorerna kan placeras ut helt öppet i terrängen och i militära sammanhang ofta i en fientlig miljö. Användaren måste kunna lita på den lägesbild som skapats från informationen som sensorerna har samlat in och fusionerat. För att möjliggöra detta måste därför sensordata skyddas så att data inte förvrängs, förstörs, förfalskas, avlyssnas, försvinner eller genereras av en fientlig sensor. Nätet måste skyddas så att inte kommunikationen till eller i nätet påverkas. Dessa krav innebär att IT-säkerhet måste integreras i nätet på olika nivåer och olika sätt.

En viktig utmaning för säkerheten i sensornät är nodernas begränsade energi- och beräkningskapacitet. En annan viktig utmaning för säkerheten är att routingen är distribuerad, vilket innebär att noderna måste samarbetar med varandra. Dessa faktorer är viktiga att beakta vid utformningen av säkerheten i sensornätet och kan utnyttjas av illasinnade och användas för att förstöra sensornätets funktioner. Det finns en rad möjliga attacker som ett sensornät kan råka ut för, t ex routingattacker så som ett svart hål och attacker på länklagret i form av att noder medvetet sänder när de inte ska eller när de hör att någon annan sänder [5].

Utmaningen för sensornät ligger i att tillhandahålla säkerhet i nätet samtidigt som energiförbrukningen måste minimeras i en värld där routingen bygger på att noder litar på varandra och kan samarbeta.

1.3 Syfte

Syftet med studien är att dels fördjupa kunskaperna om de behov som finns gällande säkerhet i trådlösa marksensornät och dels studera och jämföra metoder för att förbättra säkerhetsegenskaperna hos ett sensornät. En viktig del är att undersöka vad olika säkerhetslösningar innebär för resursförbrukningen i sensorerna.

Den attack som studeras är en routingattack av typen ett svart hål. Attacken fungerar så att en nod alltid utger sig för att ha en mycket låg kostnad till den önskade destinationsnoden, t ex mätt i antal hopp. Detta innebär att andra noder kommer inkludera denna nod i sin väg till destinationsnoden. I denna

studie är kostnaden satt till noll, vilket innebär att noden utger sig för att vara destinationsnoden. Det svarta hålet kommer sedan helt enkelt kasta all data den får från sensorerna. Data kommer därmed att försvinna på vägen från källan till destinationen.

Syftet är att studera hur två olika principer för säkerhetslösningar hanterar ett svart hål i sensornätet. Den ena principen är att använda autentisering i nätet. Noden som agerar som ett svart hål kan inte autentisera sig mot de andra noderna, dvs. verifiera att den är den nod som den utger sig för att vara, och därmed kommer data från denna nod att ignoreras och på vis kan sensornätet skydda sig mot det svarta hålet. Den andra principen är att använda multipla vägar i nätet som inte får gå över några gemensamma noder mellan käll- och destinationsnoden. På så vis kan det finnas en eller flera vägar som inte innehåller det svarta hålet och data kan komma fram till destinationen trots förekomsten av ett svart hål.

Frågan är bl a hur väl dessa två metoder skyddar mot det svarta hålet samt om de ger ett likvärdigt skydd. För att kunna avgöra detta undersöks hur väl datafusionen fungerar i nätet trots närvaron av ett svart hål. Eftersom energiförbrukningen i ett sensornät är en mycket viktig fråga undersöks även den ökade mängden data för respektive metod.

1.4 Disposition

Kapitel 2 beskriver autentisering av datameddelanden och hantering av nycklar som används för autentiseringsmekanismen. I kapitel 3 ges en beskrivning av routingprotokollet AODV samt utökningen av detta för att hitta multipla vägar. Kapitel 4 resonerar kring var energiförbrukningen i nätet sker och kapitel 5 redogör för trafikmodellen i sensornätet, inmätning av ett mål och dess tillförlitlighet. Kapitel 6 beskriver de scenarier som studeras. Kapitel 7 tillhandahåller resultat och kapitel 8 slutsatser från studien. Det sista kapitlet, kapitel 9, diskuterar kring några allmänna slutsatser.

Kapitel 2

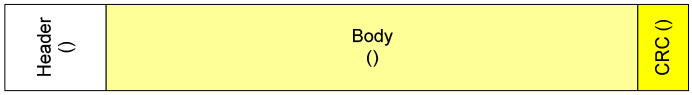
IT-säkerhet

Trovärdighet i ett meddelande kommer ofta ifrån att avsändaren är känd och accepterad samt att innehållet i meddelandet kan anses rimligt. Även om en avsändare uppger sig vara någon som är känd för mottagaren kan det vara svårt att avgöra om det faktiskt är den påstådda avsändaren som skickat ett meddelande. Tekniker för att avgöra vem som har skickat ett meddelande kallas autentisering. Gemensamt för alla autentiseringsmetoder är att de bygger på kryptografiska funktioner. Det innebär att tekniker från kryptografi utnyttjas för att bevisa äkthet i ett meddelande men meddelandet i sig är inte krypterat. Autentisering är således inte en metod för att uppnå sekretess, men autentisering är nödvändigt för att kunna avgöra t ex ursprung och accessrättigheter.

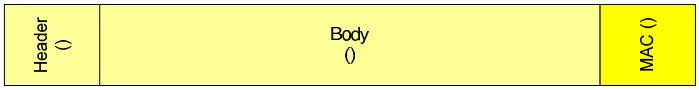
Autentisering kan göras på flera nivåer. Vanligtvis skiljer man på användarautentisering och meddelande autentisering. En mer teknisk korrekt distinktion är att skilja på symmetrisk och asymmetrisk autentisering. I symmetrisk autentisering har både sändare och mottagare samma nyckel och det är kännedom om denna nyckel som ger förtroende och tillhörighet. Vid asymmetrisk autentisering används två nycklar; en för att signera ett meddelande och en för att verifiera det signerade meddelandet.

2.1 Autentisering

Symmetriska autentiseringsmetoder är mer lämpliga i ett sensornät eftersom de är snabbare och mer ekonomiska ur ett minnes- och beräkningsperspektiv jäm-



Figur 2.1: Ett meddelandeblock med CRC.



Figur 2.2: Ett meddelandeblock med MAC.

fört med asymmetriska metoder.

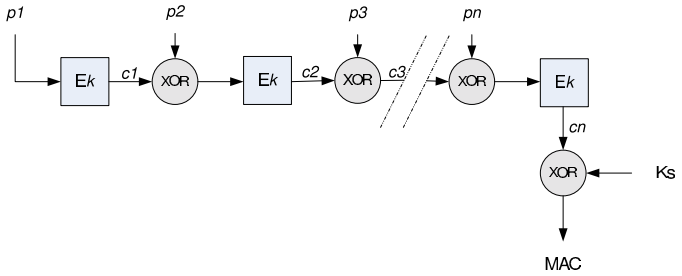
2.1.1 Meddelandeautentisering

En ekonomisk lösning för autentisering är att använda sig av en meddelandeautentiseringskod MAC (*Message Authentication Code*). Ett traditionellt meddelande utan autentisering har ett huvud (eng. *head*) och kropp (eng. *body*) samt oftast någon form av kontroll av att meddelandet är korrekt överfört. Kontrollen består av någon form av checksumma, t ex *Cycle Check Sum* (*CRC*). Figur 2.1 visar ett vanligt meddelande.

Checksumman är till för att kunna verifiera att bitarna i meddelandekroppen (body) är korrekt överförda avseende antal och inbördes ordning. Det ligger inget som helst säkerhetstänkande bakom detta då vem som helst kan räkna ut CRC:n. Ändrar en angripare meddelandet kan denne dessutom räkna ut en ny CRC och skicka med denna utan att mottagaren märker av något.

Genom att använda en MAC istället tillförs ett kryptografiskt element i form av en nyckel. En avsändare skapar sin meddelandeautentisering genom räkna ut en MAC med hjälp av en nyckel. På så sätt blir en MAC en kryptografisk checksumma som både ger högre säkerhet och ersätter den vanliga checksumman. Figur 2.2 visar ett meddelande där MAC används som meddelandeautentisering och kontrollsumma. Här har man dessutom tagit med headern i meddelandet för att öka integriteten i sändningen.

En legitim mottagaren av meddelandet äger en likadan nyckel som avsän-



Figur 2.3: CBC-MAC. Ett meddelande delas upp i block och krypteras sedan med hjälp av föregående krypterade block.

daren och kan därmed själv räkna ut värdet på MAC:en och jämföra med den medskickade. Stämmer MAC:arna överens är det rimligt att anta att meddelandet är skickat av en nod som tillhör nätet.

2.1.2 CBC-MAC

En MAC kan skapas på flera olika sätt. MAC återfinns dessutom i litteraturen under flera olika namn som t ex *kryptografisk hash* och *message digest*. Skillnaden mellan ett hashvärde och en MAC är att när man skapar ett hashvärde använder man en känd nyckel så att även mottagaren kan skapa motsvarande hash och verifiera korrektheten i överföringen. Det går att göra om ett hashvärde till en MAC genom att använda en nyckel som är hemlig för alla utom sändaren och mottagaren. Det går även att ta en MAC och göra om till en hashfunktion genom att offentliggöra nyckeln.

Ett snabbt och effektivt sätt att skapa en MAC är att använda sig av *Cipher Block Chaining* (CBC). Metoden CBC är en kryptografisk funktion baserat på blockchiffer. Ett meddelande delas upp i block och krypteras sedan med hjälp av föregående krypterade block. Figur 2.3 visar CBC.

Ett meddelande (M) delas upp i block (p) av förutbestämd storlek sådant att M = p1 + p2 + . . . pn. Det första blocket körs genom ett vald symmetrisk blockchifferfunktion vilket resulterar i ett krypterat block (c1). c1 samkörs med p2 genom en XOR-funktion. Resultatet körs därefter igenom blockchifferfunktionen och genererar ett andra krypterat block c2. Samkörning och kryptering

fortsätter så länge det finns block av meddelandet kvar. Det sista krypterade blocket, cn, samkörs i en XOR-funktion med autentiseringsnyckeln, Ks, vilket resulterar i MAC:en. En MAC är således beroende av alla block i meddelandet samt den nyckel som används för autentisering. Alla krypterade block slängs efter det att MAC:en har tagits fram, därmed skickas alltså inte data krypterat.

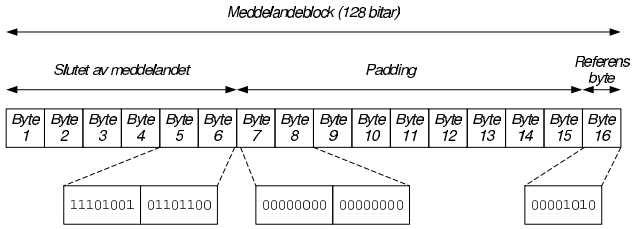
Vilken annan nod som helst i system som innehar samma nyckel (Ks) kan beräkna en MAC med samma resultat och därmed även avgöra om meddelandet kommer från legitim, dvs. autentiserad, nod.

2.1.3 Variabel meddelandelängd

Storleken på blocken (pn) är beroende av den blockchiffermetod som används. Flertalet blockchiffer på marknaden har fram till nyligen använt en nyckellängd på 56 bitar. Det är i kortaste laget för att bibehålla en basal säkerhetsnivå. Säkerheten i en MAC är i första hand beroende av två saker; säkerheten i det underliggande blockchiffret samt längden av MAC:en. Nyare blockchiffer ligger på 128 bitar medan t ex Advanced Encryption Standard (AES) använder sig av nyckellängder på upp till 256 bitar. En nyckel på 56 bitar kan idag inte räknas som tillförlitligt då det är rimligt att knäcka den med *brute force*, dvs. det är möjligt att testa alla möjliga kombinationer av nycklar inom en rimlig tid. Dessa siffror gäller dock för MAC i traditionella nätverk, dvs. med normalt sett kraftfulla datorer och utan större energirestriktioner, och är inte det samma som i sensornät.

Ett meddelande kan variera i längd. Det innebär att ett uppdelat meddelande kanske inte blir en multipel av nyckellängden. För att MAC-metoden ska kunna användas måste därmed meddelandet fyllas ut, så kallad padding. Antag att vi har en nyckellängd på 128 bitar och en rest på 48 bitar (6 byte), dvs. den sista delen av meddelandet är mindre än 128 bitar. För att komma upp i ett fullt block på 128 bitar fylls därför blocket på med t ex nollor ($'0'$). I exemplet blir det 72 bitar (9 byte) vilket lämnar oss en sista byte som kontrollbyte. Kontrollbyten får värdet på de antal $'0'-$bytes inklusive kontrollbyten som sätts in. I vårt exempel får kontrollbyten värdet 10, se Figur 2.4.

Även om ett meddelande är en jämn multipel av nyckellängden används padding för att markera avslutning på ett meddelande. I det här fallet blir hela sista blocket fyllt med $'0'$ och kontrollbyten får värdet 16.



Figur 2.4: Padding av ett 128 bitars block.

2.2 CBC-MAC i sensornät

Det här avsnittet tittar närmare på lämplighet och konsekvenser av att använda CBC-MAC i ett trådlöst marksensornät.

2.2.1 Säkerheten i en CBC-MAC

Det grundläggande säkerhetsantagandet i CBC-MAC är att den underliggande krypteringsfunktionen är säker. Genom att använda ett beprövat blockchiffer, t ex AES eller Skipjack, kan man visa att de kryptografiska beräkningarna är korrekta och robusta.

Längden på själva MAC:en, dvs. de antal bitar som MAC:en utgör har också betydelse för hur robust den är. I de flesta fall rekommenderas en längd av 8 eller 16 byte. Det motsvarar nyckellängder på 64 eller 128 bitar. DES är ett blockchiffer som tidigare var mycket populärt i CBC-MAC-sammanhang som genererar en MAC på 64 bitar. I kryptosammanhang är det dock känt sedan länge att en symmetrisk nyckel på 64 bitar kan knäckas genom *brute force*. Det gör att en MAC på 64 bitar (8 byte) inte heller kan räknas som tillförlitlig utan man bör använda sig av nycklar med minst 128 bitars längd i det underliggande blockchiffret.

Ett trådlöst marksensornät med batteridrift har dock andra förutsättningar jämfört med ett traditionellt datornät. Rekommendationen på 16 byte MAC baserar sig på att det är möjligt att testa sig fram till en korrekt nyckel och sedan kunna använda denna. I [11] hävdas att en MAC på 4 byte (32 bitar) är fullt tillräckligt i ett sensornät. Påståendet bygger på att en angripare inte kan verifi-

era om de har kommit på rätt nyckel utan att skicka ett meddelande till någon i nätverket. På så sätt kan en angripare bli tvungen att skicka upp till 2^{31} meddelanden över en begränsad kanal. I [11] uppges en tid på 20 månader för ett framgångsrikt för sök. Ett batteridrivet sensornät skulle vara uttömt lång tid innan den riktiga nyckeln framkom. Ur detta perspektiv verkar en MAC på 4 byte fullt tillräckligt. För studien väljer vi därmed en nyckellängd på 4 byte (32 bitar).

2.2.2 Effektkostnader med CBC-MAC

När två noder kommunicerar med varandra förbrukar de mer energi än när de utför beräkningar [9]. Att noden förbrukar energi när den sänder och tar emot data är inget som autentiseringen kan påverka. Däremot påverkar autentiseringen energiförbrukningen indirekt genom att den lägger på extra bitar i varje meddelande. I studien ligger därför fokus på hur mycket extra bitar som skickas och tas emot när noderna använder autentisering i form av CBC-MAC och vad det betyder för energiförbrukningen.

TinyOS [7] är ett operativsystem för sensornät utvecklat på UC Berkeley, USA. Paritetskontrollen (CRC) i TinyOS utgör 2 byte av den maximala 36 byte långa meddelandelängden. En MAC har, utöver meddelandeautentisering, samma effekt som en paritetskontroll vilket gör att den kan ersätta denna. Det gör att en MAC får CRC:n 2 byte ”gratis”. Väljer man att använda en MAC på 4 byte enligt [11] blir den totala ökningen av ett meddelandeblock 2 byte. Oavsett vilket format man väljer på meddelandeblock så kan MAC:en ersätta paritetskontrollen vilket sänker det totala antalet bitar som behöver transporteras.

2.3 Nyckeldistribution

Nyckelhantering är ett kapitel inom IT säkerhet som blir mer komplext i takt med att användandet av kryptografiska applikationer ökar. Allmänt när det gäller utnyttjandet av nycklar kan man säga:

- En nyckel har en uppgift.
  - T ex en nyckel (k1) för autentisering, en nyckel (k2) för kryptering, ect.
- Nycklar skall skyddas

- Säkerheten i större kända kryptoalgoritmer bygger på att man har en nyckel som man håller hemlig. Algoritmen i sig själv är publik. Det är särskilt viktigt att hålla en symmetrisk nyckel hemlig då det är samma nyckel som används för kryptering som dekryptering.

• Nycklar skall bytas ut efter hand

- En nyckel ”förbrukas” i det avseende att risken att en angripare lyckas samla in tillräckligt med material för att knäcka en nyckel med *brute force* ökar ju längre nyckeln används. Därav bör en nyckel bytas ut då och då. Hur ofta beror på den hot och riskkalkyl som gjorts.

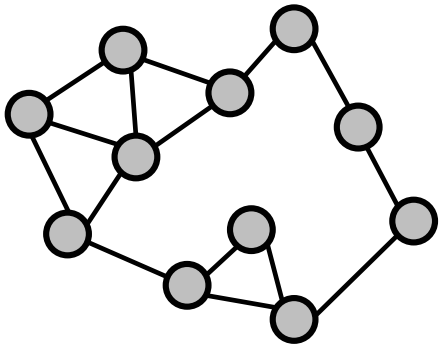
Initialt är det relativt enkelt att förse sitt system med en grunduppsättning av nycklar. Den här fasen kan kallas för före uppdrag. När väl systemet är på plats och i funktion har vi en underhållsfas där enstaka nycklar (förutsatt att alla inte har samma) bör kunna bytas ut med relativt kort varsel. Under underhållsfasen går det inte att samla in alla sensorer och byta ut nycklar utan det får ske genom de normala kommunikationskanalerna. Ett specialfall av underhåll är om man skulle behöva byta ut samtliga nycklar vid ett tillfälle. Använder systemet en gruppnyckel är detta alltid fallet vid nyckelutbyte.

2.3.1 Gruppnycklar

Den enklaste formen av nyckelinfrastruktur är en *global grupp nyckel*. En global gruppnyckel delas av alla som ingår i nätverket. Fördelen är att alla har tillgång till samma nyckel och därmed kan kommunicera med varandra så länge som sändningsförhållandena tillåter detta. Nackdelen är lika uppenbar, dvs. om en nyckel avslöjas så sätts hela systemet i ett osäkert tillstånd då en angripare kan både sända och ta emot autentiserade meddelanden med den avslöjade nyckeln.

En gruppnyckel behöver inte alltid vara global. Det finns tillfällen då man vill skapa undergrupper inom nätverket. Genom att skapa kluster av sensorer isolerar man informationsflödet och har därmed möjligheten att påverka vad som sprids och hur. Kluster kan skapas med en *klusternyckel* som delas av en mindre och intilliggande grupp av sensorer.

Nycklar med speciella syften kan också vara gruppnycklar, till och med globala gruppnycklar. Ett sådant exempel kan vara en nyckel för massutskick – en *broadcast nyckel*. Det kan vara fördelaktigt att ha en global nyckel som används



Figur 2.5: Sensornät med gruppnyckelsystem.

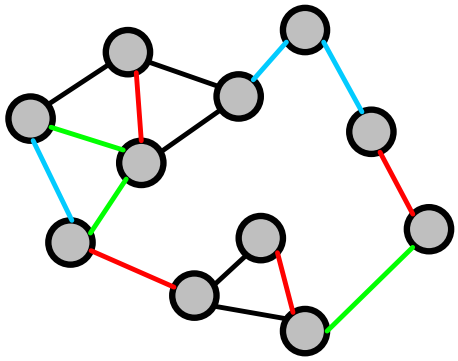
mycket restriktivt för att kunna nå samtliga noder i sensornätverket, även om man i övrigt använder sig av andra lösningar.

Figur 2.5 visar ett system där alla använder samma nyckel för att autentisera sig, vilket visas genom de svarta förbindelserna.

2.3.2 Parnycklar

Ett alternativ till gruppnycklar är att alla nodpar delar en nyckel. Tanken är att varje nod skall ha en unik nyckel att autentisera sig och sin granne med. En nod behöver således flera nycklar för att kunna autentisera sig inför sina grannar. I praktiken används en uppsättning nycklar som är mindre i antal jämfört med det totala antalet noder i nätet. Unika nycklar mellan varje nod är sällan nödvändigt och dessutom mycket svårt att genomföra. I [2] kom de fram till att i en pool av 10 000 nycklar behövs det 75 nycklar för att med 50 procent sannolikhet kunna kommunicera med alla noder i nätet. Ökar man antalet nycklar till 100 000 behövs 250 nycklar för en kontaktbarhet på 50 procent. Behovet växer således inte linjärt med utbudet.

Parnycklar minskar drastiskt effekten av att en nod blir övertagen och nycklarna i noden avslöjas. Om en nod blir övertagen kommer endast de noder som använder samma nycklar som finns i den övertagna noden att bli avslöjande. Dessa har dock möjlighet att ersätta sina nycklar med nya. I ett nät där noderna använder sig av parnycklar finns dock risken att två intilliggande noder (geo-



Figur 2.6: Sensornät med parnycklar.

grafiskt närliggande) inte kan kommunicera om de inte har någon gemensam nyckel. Oftast är detta ett hanterbart problem [2], men det kan även ge konsekvenser för kommunikationsmöjligheterna i nätet.

Figur 2.6 visar ett sensornät med parvisa autentiseringsnycklar, där respektive färg motsvarar en nyckel. Det finns alltså fyra olika nycklar i nätet.

Val av nyckelstruktur påverkar inte nödvändigtvis vilken autentiseringsmetod som används. CBC-MAC fungerar t ex både med grupp- och med parnycklar. Skillnaden ligger i hur säker man vill vara att alla sensorer kan aktiveras och sprida sin kunskap inom nätet.

Att ha i minnet är också att alla noder förstår varandra oavsett om MAC-autentiseringen veriferas korrekt eller inte. Det noden bör misstänka om MAC:en inte stämmer är att den avsändande noden kanske blivet angripen och övertagen av en fiende.

Kapitel 3

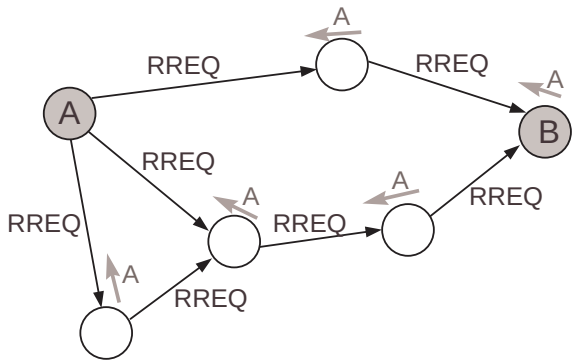
Routing

Den attack som studeras är en routingattack av typen ett svart hål. Attacken fungerar så att en nod alltid utger sig för att ha en mycket låg kostnad till den önskade destinationsnoden, t ex mätt i antal hopp. Detta innebär att andra noder kommer inkludera denna nod i sin väg till destinationsnoden. I denna studie är kostnaden satt till noll, vilket innebär att noden utger sig för att vara destinationsnoden. Det svarta hålet kommer sedan helt enkelt kasta all data den får från sensorerna.

Det routingprotokoll, eller trafikstyrningsprotokoll, som valts är *Ad Hoc On-Demand Distance Vector,* AODV [13]. Motiveringen till detta är framförallt för att det går att utvidga protokollet relativt enkelt så att det kan hitta multipla vägar i nätet till en destination. AODV är även ett standardiserat routingprotokoll för ad hoc-nät. Ytterligare en fördel med AODV är att det går att minska overheadtrafiken i nätet genom att det går att begränsa sökningen efter en väg lokalt i nätet, vilket är att föredra om en nod söker efter en väg till en nod som ligger nära.

3.1 AODV

AODV är ett reaktiv routingprotokoll som också ofta kallas *on-demand* (routing). Detta innebär att vägarna endast skapas då de behövs, dvs. på begäran från den sändande noden. En förfrågan skickas ut bland de andra noderna för att hitta rätt väg. När en väg hittas skickas en bekräftelse tillbaka till sända-

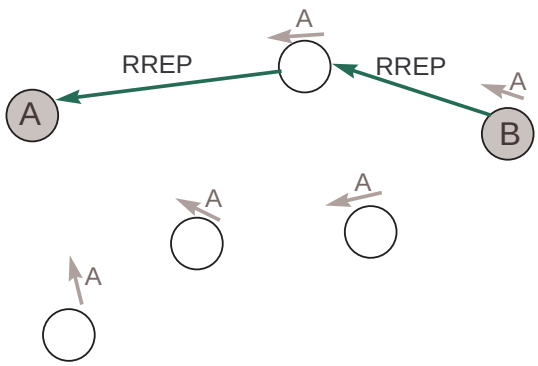


Figur 3.1: Sändaren A känner inte till någon väg till mottagare B, och sänder då en förfrågan, RREQ, genom nätet. I varje nod som passeras sätts en ”pekare” mot nod A.

ren. Vägen behålls därefter uppe tills den inte behövs mer eller tills ett avbrott sker. Algoritmen kan alltså delas upp i två delar, vägsökning samt underhåll av existerande väg.

Då en sändare A ska skicka trafik till en mottagare B börjar AODV med att söka efter en väg som paketet kan sändas längs. Finns det ingen sedan tidigare använd väg skickar sändarnoden ut en förfrågan (*route request* - RREQ) om närmaste väg till sina grannar, se figur 3.1 [14].

Genom att broadcasta, dvs. skicka förfrågan till alla grannar samtidigt sprids förfrågningen snabbt ut i nätet. Nästa nod tar emot förfrågan och skickar ut den till alla sina grannar. Noder som tar emot en förfrågan uppdaterar också sin information om sändaren A och sätter upp en *pekare* tillbaka till sändaren i sin routingtabell. Det kan liknas med hur ett system av vägar byggs samman och vägskyltar sätts upp i varje korsning, se figur 3.1. När sedan förfrågan når fram till mottagaren B eller en nod som känner till vägen till nod B skickas det en bekräftelse (*route reply* - RREP) på att en väg hittats. Denna bekräftelse tar då samma väg tillbaka som förfrågan som nådde fram först kom, se figur 3.2, och man kan därmed utöka tabellen av pekare med information om åt vilket håll mottagaren B finns. När sändare A till sist tar emot bekräftelsen kan trafik börja sändas.



Figur 3.2: Den närmaste vägen till nod B har hittats och en bekräftelse, RREP, sänds tillbaka till nod A. Pekare mot nod B sätts i noderna som passeras på vägen tillbaka och därmed kan trafiken börja flyta.

Det vanliga är att en väg upprätthålls så länge den används och för att inte för gammal och därmed felaktig information om vägar ska finnas i nätet tas information bort efter en viss tid om länken inte använts. I ett sensornät där noderna är fixa sker inte förändringar lika ofta som när noderna är mobila. Därmed är det inte lika ofta som vägen är för gammal eller felaktig i ett nät med fixa noder. I denna studie upprätthålls därför en väg under hela simuleringstiden. Om nätet är i drift under en längre tid är det dock möjligt att noder slås ut p g a att batterierna tar slut.

Varje nod i nätet har ett så kallat sekvensnummer. Detta sekvensnummer räknas upp så fort en nod skickar ett RREQ. Noden har även, i det fall att den har en väg till en nod, sparat respektive noders sekvensnummer. Detta sekvensnummer används för att noderna ska kunna bestämma vilken uppgift om en väg som är den senaste av flera möjliga. Om t ex nod A ska hitta en väg till nod C kommer den skicka ut ett RREQ. I detta RREQ sätter den bl a sitt egna sekvensnummer och det senaste sekvensnumret den hade för nod C. Om den inte har något väg till nod C sedan tidigare kommer den att sätta sekvensnumret till -1. En mellanliggande nod som har en väg till nod C med ett högre sekvensnummer vet då att denna väg är mer aktuell och kan då att svara med ett RREP tillbaka. Om den mellanliggande noden istället har ett lägre sekvensnummer till nod C

kommer den inte svara med ett RREP. När slutligen nod A får ett RREP för väg till nod C kommer den att sätta om sitt sekvensnummer för nod C till det värde som skickades med i RREP.

I studien används ”utökande ringsökning” *eng. expanded ring search*. Detta innebär att ett RREQ kan begränsas i hur långt det ska spridas i nätet. Första gången skickar t ex noden endast RREQ:et ett hopp bort och om den inte får något RREP tillbaks kommer noden att sända om RREQ:et, denna gång t ex tre hopp bort och så vidare. På så vis kommer RREQ:et att återsändas längre och längre bort i nätet för att till slut gå till hela nätet. I denna studie där sensornätet består av ett stort antal noder, men datafusionen sker relativt lokalt i nätet är det önskvärt att kunna begränsa skickandet av RREQ förfrågan till och i närheten av det område där datafusionen sker.

3.2 AODV med multipla vägar

Om trafiken bara skickas en väg i nätet och det finns ett svart hål är risken stor att trafiken skickas till det svarta hålet. Om routingen istället använder sig av multipla vägar, så som AODV med multipla vägar, som inte har några gemensamma noder på vägen kan det finnas en chans att trafiken kommer fram till destinationsnoden ändå. Noden skickar nämligen samma paket över flera vägar i nätet och därmed kan något av de multipla paketen komma fram till destinationen. På så vis skulle effekten av ett svart hål kunna minskas. I [16] ges ett förslag på hur AODV kan modifieras så att det går att hitta flera vägar till samma destination från en sändare, dvs. att hitta multipla vägar i nätet. De vägar som skapas går inte över några gemensamman noder mellan start- och destinationsnod (eng. node-disjoint).

3.2.1 Att hitta multipla vägar

AODV med multipla vägar fungerar så att mellanliggande noder sparar information från alla RREQ i en RREQ tabell. Informationen som sparas i RREQ tabellen är vilken nod som har sänt ut RREQ från början, destinationsnoden, d v s den nod som noden vill hitta en väg till, vilken granne RREQ kom ifrån och slutligen antalet hopp som RREQ har gått. Mellanliggande noder får inte svara direkt med att skicka RREP tillbaka till den nod som frågar om vägen, även om

de redan känner till en väg till den destinationen. När destinationsnoden slutligen får RREQ så kommer den att skicka ut ett RREP. Antalet RREP som noden svarar med är dock begränsat. För varje RREQ som destinationen svarar på den att svara med ett RREP till den nod som den fick ett RREQ av. Ett RREP innehåller ett unikt väg-id som skiljer sig åt för varje RREP. Samma RREQ kommer alltså att nå destinationsnoden från flera olika noder och besvaras av destinationsnoden med ett RREP innehållande bl a ett unikt väg-id.

När en mellanliggande nod får ett RREP från en granne kommer den att gå in i sin RREQ tabell och ta bort den noden ur tabellen och istället sätta den som nästa hopp i sin routingtabell. Genom routingtabellen vet noden därmed till vilken nod den ska sända ett paket som ska till en viss destination. Noden kommer därefter att gå in i sin RREQ tabell för att hitta den kortaste vägen till den nod som den tidigare fick RREQ av, dvs. vilken grannod den ska sända till. Noden kommer därefter att skicka RREP till denna granne och plocka bort den ur RREQ tabellen. Om en mellanliggande nod skulle få ett RREP som den inte kan skicka vidare p g a att dess RREQ tabell är tom eller att den redan sänt ett RREP med annat väg-id kommer den att skicka tillbaka ett Route Discovery ERror, RDER, till den nod som skickade RREP. Noden som tar emot RDER kommer då att försöka hitta en annan väg med hjälp av sin RREQ tabell och skicka RREP till en annan nod som den hittar i tabellen. Antalet RREP som en nod skickar ut är begränsat. Noder som överhör ett RREP plockar bort den nod som RREP kom ifrån i sin RREQ tabell. På så vis undviks att en nod skickar onödiga RREP till noder som ändå kommer besvara detta med ett RDER eftersom de redan har skickat och tagit emot RREP sedan tidigare.

Den nod som från början skickade ut förfrågan om en väg i form av ett RREQ kommer att skicka ett Route Reply ConfirMation paket, RRCM till destinationsnoden. Detta paket innehåller bl a information om väg-id och antal hopp till destinationen. Detta görs för att destinationsnoden ska få veta hur många RREP som verkligen kom fram av de som den skickade ut och så att den kan sätta upp dessa vägar i sin routingtabell. I studien är det maximala antalet vägar som används vid AODV med multipla vägar begränsat till två.

3.3 Routingattack i form av ett svart hål i AODV

En övertagen nod kan utnyttja AODVs funktioner och börja agera som ett svart hål [6]. När en nod skickar ut ett RREQ kommer det svarta hålet, en nod i nätet, att svara med ett RREP att den har en väg till destinationen. Den kommer att i detta RREP sätta antalet hopp till noll och sekvensnumret till högre än det som skickades ut i RREQ:et. Mellanliggande noder som får RREP:et och den nod som skickade ut RREQ:et kommer troligen att spara denna väg i respektive routingtabell. Konsekvensen av detta blir att noden ser ut att vara destinationen. Svarta hålet kommer i ett senare skede när den tar emot nyttotrafik från andra noder att slänga dessa paket, dvs. den kommer inte att vidarbefodra paketen. Detta leder till att nyttotrafiken aldrig kommer fram till den tänkta destinationen utan försvinner i det svarta hålet.

Kapitel 4

Energimodell

En sensorplattform kan delas in i ett antal huvudkomponenter: sensor, radio och CPU (Central Processing Unit). Beskrivningen nedan är en möjlig lösning i en sensorplattform. I realiteten byggs detta “komplexare” än beskrivet nedan.

4.1 Komponenter i sensorplattformen

Sensorenheten är den del som känner av omgivningen på något sätt (ljud, markvibrationer, gas etc.) På en sensorplattform kan denna del bestå av multipla sensorer. Sensorernas signaler skickas till CPU-delen för vidare behandling. Radioenheten sköter den externa kommunikationen med andra sensorer och till en eventuell användare utanför nätet.

CPU är den del som handhar filtrering av sensorinformation för att hantera t ex falskalarm eller allmänt sköta tröskling av sensordata så att inte information i onödan skickas vidare i nätet. CPUn handhar också radioalgoritmerna för bl a kanaltilldelning och routing.

CPU lasten är oftast inte försumbar. CPUn kan stå för merparten av energikonsumtionen. En enda skickad bit kan i vissa fall jämföras med 3000 CPU-instruktioner [1]. Mycket beror dock på den utsända effekten. P_{ut} = 1W gör kanske att CPU-energin är försumbar, men P_{ut} = 1mW gör det inte.

4.2 Energikonsumtion i sensorplattformen

Protokollval

Ur ett energiperspektiv konsumerar huvudkomponenterna olika mängder energi beroende på sensornodens aktuella status. En stor påverkan på energiåtgången i CPUn är valet av signalbehandling för sensordata innan den skickas vidare till andra sensorer.

För radiodelen gäller det motsvarande vid val av protokoll för kommunikationen. Ett protokoll som skickar mycket information över tiden kommer att belasta batteriet i sensornoden mer, än ett protokoll som så långt det är möjligt minimerar informationsutbytet. Andra faktorer som påverkar är noddensiteten, dvs. hur tätt sensorerna ligger, vilket avgör hur mycket effekt noderna måste sända med för att informationen från en sensor ska nå fram till de andra.

Värt att notera är att för korta sändningsavstånd, då vi inte behöver sända med speciellt stor effekt, kostar det lika mycket energimässigt att sända data som att ta emot data [3]. I studien beaktar vi därmed både antalet bitar som sänts och tagits emot.

Kretsval

Stor betydelse för energikonsumtionen i sensorplattformen har valet av kretsar. Genom att välja en strömsnål CPU är mycket vunnet. Som jämförelse [3] kan tas WINS som med alla komponenter aktiva och vid sändning konsumerar 1W medan MEDUSA II i samma läge konsumerar ca 25 mW, dvs. en faktor 40 mindre.

Spara energi

Genom att inte aktivera fler delar i sensornoden än de som behövs för en aktuell uppgift finns möjlighet att spara energi. Sensorn kanske inte behöver vara igång mer än någon minut per timme. Vad gäller radiodelarna är det därmed lämpligt med ett tidluckebaserat kanaltilldelningsprotokoll, vilket gör det möjligt för radiodelen i en sensornod att helt stänga av sig tills det är dess tur att sända eller ta emot information från någon annan sensornod. På detta sätt kan energiåtgången begränsas med heltalsmultipler. Viktigt att notera är dock att gå från ett läge till ett annat t ex sovande till aktiv innebär även det en kostnad. Det kan vara så

att sensornoden behöver sova en längre stund för att energibesparingen inte helt ska ätas upp av extrakostnaden som lägesförändringen innebär [12].

Avgränsningar

Vi kommer inte att utgå från en verklig sensorplattform då vi inom denna studie bara är intresserade av en jämförande mätning mellan två olika säkerhetsmetoder. Vi mäter endast på antalet bitar som har sänts och tagits emot. Vi lämnar alltså diskussionen om energiåtgång i absoluta termer därhän. Vi har i studien begränsat oss till att undersöka energiförbrukningen kopplat till antal bitar som sänts respektive tagits emot. Därmed studeras inte energiförbrukningen för olika algoritmer som t ex signalbehandlingen av sensordata eller valda protokoll för autentisering och trafikstyrning.

Kapitel 5

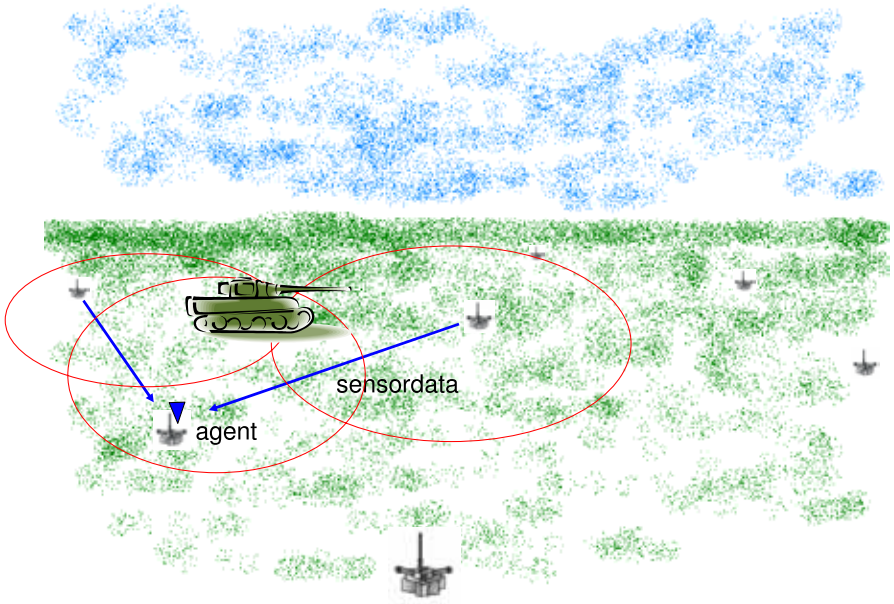
Sensornätmodell

5.1 Datafusion

För att få ut relevant information och försöka minimera eventuella felkällor så fusioneras data från olika sensorer. För denna studie har vi valt att studera den typ av datafusion som användes i FoT-projektet Interaktiva Adaptiva Marksensornät (IAM). Datafusionen är en distribuerad fusionsarkitektur [8]. En generell trafikmodell för denna typ av datafusion finns beskriven i [4]. Den studerade datafusionen i sensornätet är modellerad utifrån denna.

Varje objekt eller mål som sensorerna detekterar motsvaras av en så kallad agent. Agenter är autonoma program som utför en specifik uppgift och kan flytta sig i nätet. Agentens uppgift är att fusionera data från de sensorer som kan detektera målet, för att t ex kunna följa och klassificera ett mål, och den är unik för respektive mål som registrerats i nätet, se Figur 5.1. Innan en sensor skickar vidare data till agenten så sker en lokal signalbehandling i varje sensornod. I agenten sker sedan ytterligare beräkningar utifrån samlad data från sensorerna. För denna studie har vi avgränsat oss och behandlar endast skickande av sensordata till agenten och flytt av agent i nätet och utför inte signalbehandling av data.

Vilken nod som ska äga agenten avgörs av avståndet från respektive sensor till målet, där den nod som befinner sig närmast målet ska äga agenten. På så vis kommer agenten att följa målet i nätet och flyttas mellan sensorerna så att agenten befinner sig så nära målet som möjligt under hela tiden sensornätet



Figur 5.1: Datafusion i form av distribuerad fusionsarkitektur. Den blå triangeln motsvarar agenten i nätet för fordonet. Agenten flyttas i nätet beroende på hur målet, objektet, rör sig.

detekterar objektet. Syftet är att hålla nere trafiken i nätet så mycket som möjligt. Hur noden vet att den ska flytta agenten och vart den ska flytta den ligger utanför denna studie.

Noder i nätets som detekterar målet kommer att skicka sensordata till den nod som äger agenten för tillfället. Den datamängd som skickas från en nod till agenten är totalt 668 bitar [4]. Antalet sensorer som skickar sensordata till agenten varierar och beror på hur målet rör sig i nätet, positionerna på sensorerna, hur tätt de ligger utplacerade samt vilken detektionsradie sensorerna har. Hur mycket data som skickas över vid flytt av agenten varierar i denna studie beroende på hur många sensorer som varit involverade i fusionen under den senaste sekunden. Den information som skickas över vid flytt av agenten är hela agentens tillstånd och varierar med antalet sensorer som är involverade i att skicka data till agenten. Om till exempel tre sensorer bidrar med sensordata till agenten kommer agenten bestå av cirka 1800 bitar.

Om agenten försvinner på vägen p g a att ett svart hål inte skickar vidare agenten kommer en ny agent att skapas av den nod som befinner sig närmast agenten efter en bestämd tid. I studien har vi dock valt att endast studera fallet då vi inte tillåter att en ny agent skapas om den gamla försvinner. Detta för att just se om agenten försvinner på vägen p g a det svarta hålet.

Noder som detekterar målet kommer att skicka över sensordata en gång i sekunden till agenten. Agenten kan därefter fusionera all data från sensorerna och kan på så vis ge ett säkrare svar på t ex målets position och vad det är för typ av mål, t ex typ av fordon.

5.1.1 Tillförlitlighet hos positionsbestämningen av målet

För att uppskatta positionen på det mål som detekteras måste två eller fler sensorer samarbeta. Om två sensorer som ligger på samma linje som målet skattar positionen blir inte skattningen speciellt bra. Detta beror på att sensorena endast kan bestämma att målet befinner sig på linjen, men osäkerheten (variansen) längs linjen är oändlig.

I studien beräknar vi tillförlitligheten hos uppskattningen av positionsbestämningen för målet som datafusionen ger. Detta görs genom att ta fram kovariansmatrisen för positionsbestämningen [15]. Kovariansmatrisen ger ett mått på samverkan mellan egenskaperna, dvs. i detta fall mellan sensorernas och målets koordinater.

Målets position anges av x (x-koordinat och y-koordinat) och sensorns position av p_i (x-koordinat och y-koordinat). Sensorns uppskattning av bäringen kan antas ha standardavvikelsen σ = 0.2. Mål-sensor vektorn är

$$\Delta_i = p_i - x, \tag{5.1}$$

och mål-sensor avståndet är

$$d_i = \sqrt{(x_{p_i} - x_x)^2 + (y_{p_i} - y_x)^2}. \tag{5.2}$$

Det normaliserade ortogonalkomplementet definieras som

$$\Gamma_i = [\frac{\Delta_i}{d_i}] \perp . \tag{5.3}$$

Det normaliserade ortogonalkomplementet är vektorn som är ortogonal mot vektorn mellan målet och sensorn. I dess riktning har vi en reducerad osäkerhet medan längs vektorn mellan målet och sensorn är däremot osäkerheten obegränsad. För att reducera denna osäkerhet krävs därmed att en annan sensor ger en normaliserad ortogonalvektor med annan riktning, helst ortogonal, mot den första normaliserade ortogonalvektorn.

Kovariansmatrisen, P_i, är en uppskattning av kovariansmatrisen för positionsbestämningen från sensor i vid detektion av målet x. Inversen av denna matris är

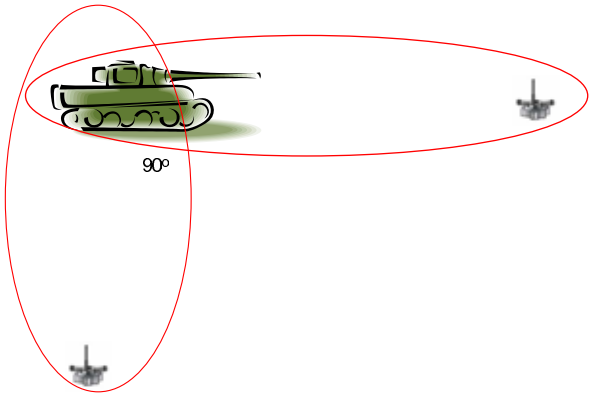
$$P_i^{-1} = \frac{\Gamma_i \Gamma_i^T}{(d_i \sigma)^2}. \tag{5.4}$$

Kovariansmatrisen (inversen) för alla sensorers detektioner av x ges av

$$P^{-1} = \Sigma_i P_i^{-1}. \tag{5.5}$$

I denna summeras informationen från respektive sensor och det är alltså önskvärt att sensorerna tillsammans bidrar med information i x- och y-riktning. Kovariansmatrisen ger ett osäkerhetsområde format som en plan ellips, vilket motsvarar osäkerheten hos uppskattningen av målets position.

Summan av diagonalelementen i kovariansmatrisen, P, ger ett spår-värde (eng. trace). Ett högt spår-värde betyder stor osäkerhet hos positionsbestämningen och ett lågt spår-värde betyder en liten osäkerhet hos positionsbestämningen.

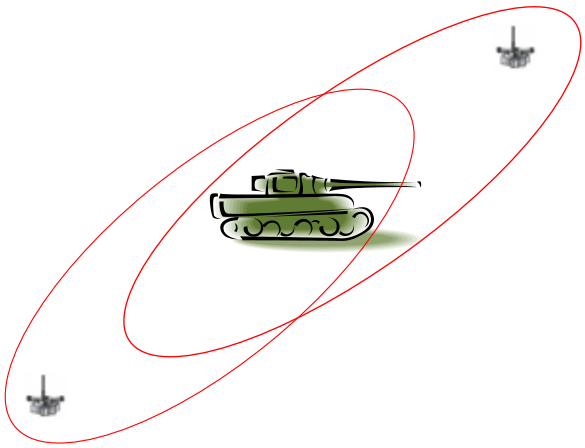


Figur 5.2: Sensorena ligger med 90 graders vinkel sett utifrån målet, vilket ger god information.

Värdena i kovaraiansmatrisen beror alltså på respektive sensors position i förhållande till målet, antalet sensorer och avstånd mellan respektive sensor och målet. Om två sensorer detekterar ett mål och de befinner sig med 90 graders vinkel mellan varandra sett utifrån målet och med ett avstånd på 20 meter från målet kommer spår-värdet bli 32, se Figur 5.2. Jämför detta mot ett spår-värde som är obegränsat om sensorena ligger på samma avstånd från målet men på samma linje som målet, se Figur 5.3, där dessa sensorer bidrar med väldigt lite information tillsammans.

5.2 Kanaltilldelning

Att använda *Time Division Multiple Access*, TDMA, som kanaltilldelninsprotokoll är motiverat ur energisynpunkt eftersom det ger möjlighet för respektive sensorer att gå ner i sov-mod i de tidluckor noden inte ska sända i. Dock är denna typ av konfliktfritt protokoll mindre lämpat för det trafikflöde som datafusionen ger. Datafusionen innebär nämligen att mycket trafik skickas lokalt i det område i sensornätet där det upptäckta målet befinner sig. I ett nät med



Figur 5.3: Sensorena ligger på samma linje som målet, vilket ger dålig information.

många noder och få som sänder innebär det att de noder som behöver sända måste vänta länge på sina tidluckor medan övriga tidluckor går tomma. Detta leder till ökade fördröjningar. En viktigt del i datafusionen är att flytta över agenten som är av relativt stor meddelandelängd. Om detta tar för lång tid p g a att noden måste vänta länge på sin tidlucka så kommer målet att hinna förflytta sig långt bort i nätet och agenten börjar då att släpa efter målet avsevärt. För att förbättra kanaltilldelningen bör därmed kanaltilldelningsprotokollet vara trafikadaptivt, dvs. en nod tilldelas mer kanalresurser vid behov. Om nätet består av väldigt många sensorer bör även någon form av spatiell återanvändning av kanalresurserna användas.

För denna studie har vi valt att använda en form av optimal TDMA. För varje tidlucka får den nod sända som har det äldsta paketet i kön. Därmed blir även protokollet trafikadaptivt eftersom noden allokeras tidluckor efter hur mycket den har att sända. Motiveringen till att vi valt denna optimala kanaltilldelning är för att minska risken av att agenten försvinner p g a att det tar alldeles för lång tid för noden som har agenten att få sina tidluckor och därmed att få över agenten. I studien är syftet att studera om agenten försvinner p g a routingattacken och

inte till följd av dåligt resursutnyttjande.

5.3 Autentisering

Autentiseringen görs med hjälp av en publik nyckel på 32 bitar. Detta innebär att varje informationsblock kommer innehålla extra bitar motsvarande storleken på nyckeln för en checksumma. Eftersom autentiseringen kräver att varje informationsblock är en multipel av nyckelstorleken kan det innebära att informationsblocket måste fyllas ut med nollor eller ettor för att uppnå denna storlek, så kallad padding. Paddingdata kommer dock att slängas när informationen tas emot i destinationsnoden. Autentiseringen av data sker både för routingtrafik, dvs. av AODV-paket så som t ex RREQ och RREP, och av sensordata.

När autentisering används ignoreras information från det svarta hålet eftersom noden inte kan styrka sin identitet. Noden som blivit ett svart hål är fortfarande en deltagare i nätet och har därmed allokerade tidluckor och kan utbyta trafik med de andra noderna därför att övriga noder inte kan avgöra om blocket är riktigt eller ej förrän ett informationsblock är mottaget. Om blocket inte klarar en autentisering kommer övriga noder aldrig att skicka information till den noden eller behandla mottagen information därifrån.

Kapitel 6

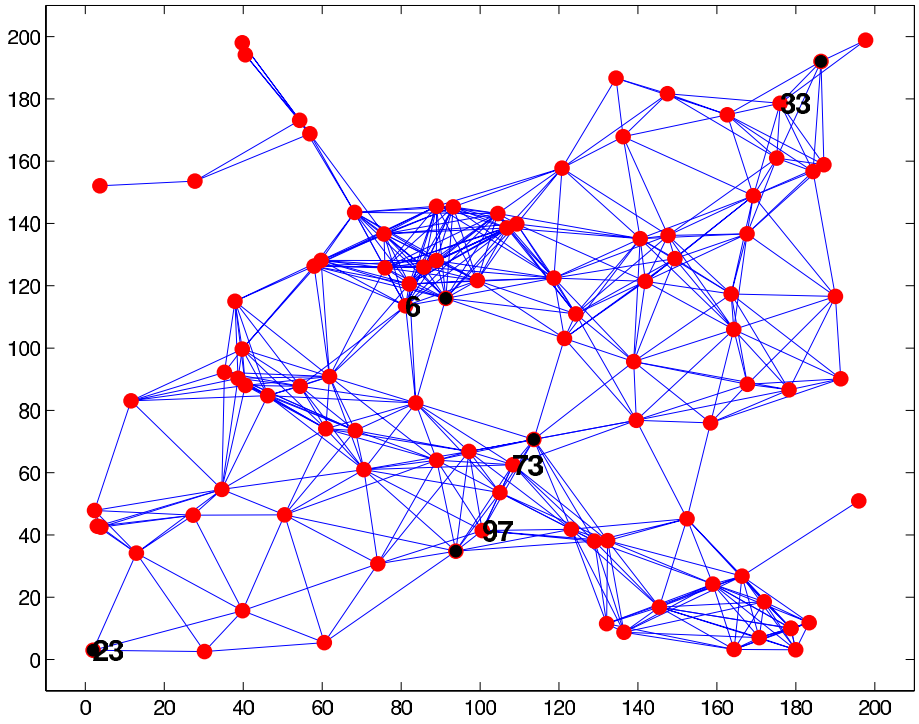
Scenario och Utvärdering

6.1 Scenario

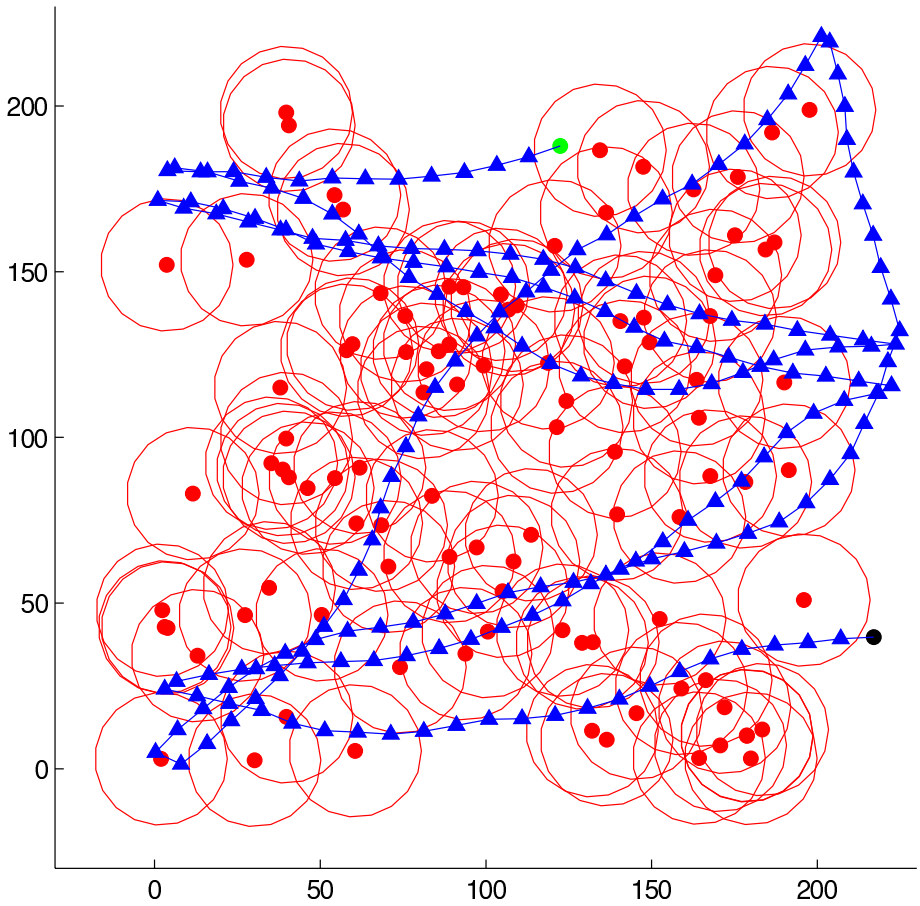
Det scenario som studeras är ett nät bestående av 100 sensorer utspridda över en yta på 200x200 m^2. Kommunikationsräckvidden hos sensorerna är cirka 20-30 meter, se Figur 6.1. Datatakten i nätet är 150 kbps och frekvensen ligger på 300 MHz. Uteffekten är satt så att nätet är förbundet, dock har vi ingen bra vågutbredningsmodell på så låg höjd. Detektionsområdet för respektive sensor är satt till 20 meter och visas med den yttre röda cirkeln, se Figur 6.2.

Antal objekt, t ex ett fordon, som detekteras i nätet är ett och rör sig i nätet med en hastighet på 10 m/s. Den blå linjen visar hur agenten har rört sig under 200 sekunder, med start i den gröna punkten och slutpunkt i den svarta, se Figur 6.2. I Tabellen 6.1 visas de scenarier som studeras. Simuleringstiden för varje scenario är 400 sekunder.

Det svarta hålet är en nod i sensornätet som har blivit övertagen av någon obehörig och manipulerad med, dock är inte nyckeln i noden röjd. Om autentisering används i nätet kan inte den tillkomna noden komma in i nätet eftersom den inte kan autentisera sig. För scenariet med ett svart hål testas fem olika positioner för det svarta hålet, allt från en position i ytterkant till centralt i nätet. De sensorer som agerar svarta hål är nod 6, 23, 33, 73 och 97 och visas i Figur 6.1 som svarta noder.



Figur 6.1: Sensornätet med 100 noder slumpmässigt utspridda över en yta på 200*200 m^2. Visar nättopologin med de länkar som finns mellan noderna, dvs. de sensorer som kan kommunicera. De svarta noderna är de som agerar svarta hål, vilka är nod 6, 23, 33, 73 och 97.



Figur 6.2: Agentens rörelse genom nätet under 200 sekunder. Startar i den gröna punkten och slutar i den svarta. De yttre röda cirklarna anger respektive sensors detektionsområde.

Tabell 6.1: Studerade scenarier

Scenario	Svarta hål	Autentisering	AODV	AODV-multipath
Scenario 1			x	
Scenario 2				x
Scenario 3		x	x	
Scenario 4	x		x	
Scenario 5	x	x	x	
Scenario 6	x			x

6.1.1 Parametrar för datafusionen i modellen

Sensorerna antas kunna detektera ett mål på 20 meters avstånd. Varje sensor som har detekterat ett mål kommer att skicka sensordata till agenten en gång i sekunden. Om agenten håller på att flyttas till en ny nod kommer sensordata skickas till denna. Agenten kan inte skickas vidare till någon ny nod förrän den har flyttats över helt.

Beräkningen av spår-värdet (se kapitel 4) som anger tillförlitligheten hos den uppskattade positionen av målet sker en gång i sekunden och baseras på all sensordata som anlänt senaste sekunden från andra sensorer i nätet samt data från noden som äger agenten om den detekterar objektet. Sensordata som är äldre än en sekund tas därmed inte med i beräkningen av spår-värdet. Likaså avgörs storleken på agenten (antalet bitar) av hur många sensorer som skickat sensordata till agenten under den senaste sekunden. Detta innebär att vid flytt av agenten så beror dess storlek på inkommen data under senaste sekunden. Det finns dock en minsta storlek på agenten satt till den storlek som data från en sensor ger, vilket motsvarar ägaren själv.

6.2 Utvärdering

För att utvärdera och jämföra de studerade scenarierna använder vi ett simuleringsverktyg som modellerar ett kommunikationsnät. Programmet speglar i princip de olika lagren i protokollstacken med applikations-, transport-, nät- och länklagret.

6.2.1 Utvärderingsparametrar

En utvärderingsparameter som används är totalt antal skickade bitar och totalt antal mottagna bitar för overheadtrafik, i detta fall extra bitar för autentiseringen. Antalet skickade bitar respektive mottagna bitar för respektive scenario med autentisering jämförs mot scenarierna då ingen autentisering används. Detta ger en uppfattning om hur energikrävande metoden för autentisering är. Detta eftersom att varje skickat och mottaget paket förbrukar energi i noden.

Ytterligare en utvärderingsparameter är spår-värdet som återspeglar tillförligheten hos uppskattningen av målets position. Spår-värdet ger en uppfattning av hur väl datafusionen fungerar i nätet. Vi antar att ett spår-värde som är mindre eller lika med 100[1] är acceptabelt för det scenario vi studerar. Den utvärderingsparameter som används är hur stor del av simuleringstiden som spår-värdet ligger under eller är lika med detta värde.

[1]Räknar med en standardavikelse på 7 meter, vilket motsvarar en varians på $7^2 = 49$. Spår P är i någon bemärkelse summan av x och y riktningens varianser, så därav spår-värdet $P =< 100$.

Kapitel 7

Resultat

7.1 Autentisering

I Tabell 7.1 ges resultaten för scenariot med respektive utan autentisering. Att använda autentisering ger en ökad mängd overheadtrafik för nätet att hantera. Mängden sensordata ökar som skickas respektive tas emot ökar med cirka 7 procent när autentisering används. Ökningen av mängden routingdata som skickas är cirka 36 procent och cirka 34 procent för mottagen data. Anledningen till ökningen är högre för routingdata beror på att de extra bitar som läggs till meddelandet för autentisering utgör en större andel av ett routingmeddelande än av ett sensordatameddelande.

Av overheadtrafiken utgör bitpaddingen cirka 40 procent. Det kan därmed vara lämplig att avpassa meddelande-/paketstorleken till autentiseringens blockstorlek för att minimera bitpaddingen som inte innehåller någon nyttodata.

Datafusionen

I Tabell 7.2 visas andelen av tiden som spår-värdet ligger under eller lika med 100 för respektive scenario. Om ett svart hål introduceras i nätet när det inte finns något säkerhetsskydd är det risk att agenten försvinner i det svarta hålet, vilket är fallet när nod 6, 33 respektive 73 agerar som ett svart hål, se Figur 6.1. I dessa fall suger noderna åt sig agenten genom att det utger sig för att vara destinationen. Detta sker i dessa fall precis i början av simuleringen, därmed fås aldrig något spår-värde. Alternativt så beslutar sig applikationen för att skicka

Tabell 7.1: Resultat från scenariet med respektive utan autentisering.

Scenario	Sänd sensordata [kbit]	Mottagen sensordata [kbit]	Ökning [%]	Sänd routingdata [kbit]	Ökning [%]	Mottagen routingdata [kbit]	Ökning [%]
Utan autentisering	≈ 885	≈ 885	-	≈ 191	-	≈ 1387	-
Med autentisering	≈ 950	≈ 950	≈ 7	≈ 259	≈ 36	≈ 1854	≈ 34

agenten till en nod som är ett svart hål, vilket är fallet när nod 23 respektive 97 agerar svart hål. Om ett svart hål ligger längre bort än destinationen kommer flytten av agent eller skickande av sensordata att fungera med vanliga AODV. Att sensordata försvinner p g a ett svart hål inte skickar sensordata eller suger åt sig sensordata från andra noder verkar inte ha så stor effekt på datafusionen så länge tillräckligt många noder med lämplig position når fram med sin sensordata till agenten.

Om autentisering används så fungerar datafusionen till stor del opåverkad av ett svart hål. Den knappt märkbara försämringen av tiden för scenarierna med ett svart hål i nod 33, 73 respektive 97 beror på att applikationen skulle ha skickat agenten till dessa noder, men tvingas att välja en nod som ligger något sämre till för att inte skicka agenten till det svarta hålet.

7.2 Multipla vägar

Scenariot utan något svart hål och AODV med multipla vägar ger endast ett litet antal multipla vägar på cirka 13 procent av det totala antalet vägar som används.

Tabell 7.2: Andel av simuleringstiden i procent som spår-värdet ligger under eller lika med 100.

Scenario, procent av tiden	Utan autentisering	Med autentisering
Inget svart hål	44.75	44.75
Svart hål i nod 6	0	44.75
Svart hål i nod 23	12.00	44.75
Svart hål i nod 33	0	44.50
Svart hål i nod 73	0	44.25
Svart hål i nod 97	10.75	44.00

Detta beror på att om nod A hör när destinationsnoden, nod B, svarar med ett RREP kommer nod A att spara denna information i sin routingtabell. När nod A sedan vid ett senare tillfälle ska sända till nod B har den redan informationen om vägen till nod A sparad och kan skicka iväg sin data direkt. Nod A kommer därmed aldrig skicka ut något RREQ och det ges ingen möjlighet att hitta multipla vägar. De få gånger som det är möjligt att hitta multipla vägar är när en nod saknar information om vägen till destinationsnoden och måste skicka ut ett RREQ. För att försöka öka mängden hittade multipla vägar kan en nod förbjudas att spara information om RREP som inte berör den, dock är detta mycket onödigt ur resurssynpunkt.

Köerna växer snabbt vid AODV med multipla vägar vid den låga datatakten på 150 kbps. Om datatakten höjs till 500 kbps så försvinner denna överbelastning. Anledningen till överbelastningen för AODV med multipla är att denna generar mycket mer routingtrafik än vanlig AODV.

Resultaten från scenarierna med multipla vägar visar att det inte fungerar speciellt bra med multipla vägar för att komma runt ett svart hål i nätet. Om det svarta hålet ligger närmare noden som skickar ut RREQ än destinationsnoden kommer routingprotokollet endast att hitta multipla vägar till det svarta hålet. Detta beror på att RREP:et från det svarta hålet når källnoden fortare än RREP:et från den riktiga destinationen. Om det svarta hålet begränsas så att de endast får svara med ett RREP oavsett antalet RREQ hittar däremot routingprotokollet även en väg till den verkliga destinationen. Problemet med denna begränsning är dock att hela nätet måste genomsökas för att vara säker på att den verkliga destinationen hittats, vilket är väldigt resurskrävande. Det måste även finnas en

inbyggd teknik i noderna för att de ska kunna upptäcka att en nod svarar med flera RREP och därmed kunna ignorera dessa RREP.

Ytterligare en svaghet med AODV med multipla vägar är att ett svart hål kan sätta ett väldigt högt sekvensnummer, alltså inte enligt standarden, vilket gör att protokollet endast hittar vägar till det svarta hålet. Detta beror på att protokollet endast använder vägar med samma sekvensnummer och väljer den väg som har högst sekvensnummer. För att undvika att detta sker måste noderna ha en teknik för att kunna upptäcka att en nod alltid svarar med ett högt sekvensnummer och eventuellt exkludera denna nod, dvs. någon form av intrångsdetektering. Detta kräver extra resurser av noderna.

Om det svarta hålet inte sätter ett högt sekvensnummer, dvs. sätter ett korrekt sekvensnummer, och inte svarar med mer än ett RREP så kan multipla vägar för scenarierna hantera ett svart hål tills det att applikationen (dvs. datafusionen) beslutar sig för att skicka agenten till ett svart hål, vilket inte hindras av multipla vägar.

Kapitel 8

Slutsatser

För datafusionen är det mest kritiska själva flytten av agenten, dvs. effekten av att det svarta hålet suger åt sig agenten så att den försvinner. Dock kan datafusionen skapa en ny agent efter ett tag vilket minimerar skadan av detta.

Att använda autentisering för routingtrafiken i AODV skyddar effektivt mot ett svart hål samt att autentisera sensordata förhindrar att en nod kan skicka falsk sensordata. Detta kostar dock energimässigt genom att extra bitar måste skickas i nätet. Studien visar på en ökningen av mängden routingdata med omkring 34 procent och omkring 7 procent för sensordata.

Att använda AODV med multipla vägar för att skydda sig mot ett svart hål är inte effektivt. I det fall det svarta hålet befinner sig närmare källan än destinationen kommer AODV med multipla vägar endast hitta vägar till det svarta hålet. Om däremot destinationen ligger närmare källan kommer multipla vägar hittas, fast i detta fall fungerar det lika bra med vanlig AODV, dvs. multipla vägar fyller ingen funktion. Om det svarta hålet däremot endast får svara med ett RREP för ett RREQ och inte sätta ett falskt sekvensnummer kommer multipla vägar hittas även i det fall att det svarta hålet ligger närmare källan än destinationen. Ett problem med detta är att hela nätet måste genomsökas för att protokollet ska vara säker på att den hittat en väg till den riktiga destinationen. Det finns heller ingen som hindrar att applikationen skickar agenten till det svarta hålet där den försvinner.

Att försöka hantera attacken av ett svart hål på routingnivå med metoden multipla vägar är ingen lösning. Denna metod har ett antal begränsningar samt är mycket resurskrävande. Autentisering däremot skyddar mott attacken av ett

svart hål och är inte lika resurskrävande som multipla vägar med avseende på mängden skickad och mottagen data.

Kapitel 9

Diskussion

Lämplig kanaltilldelning i sensornät

Det trafikmönster som flytt av agent och datafusionen ger är olikt ett trafikmönster där kommunikation sker mer slumpmässigt i nätet. Datafusionen och flytt av agenten sker relativt lokalt och berör endast ett begränsat antal noder. För att hantera denna typ av trafik om en kanaltilldelning av TDMA-typ används krävs det någon form av trafikadaptivitet, dvs. att noder som har behov av mer kanalresurser resursuttnyttjandet om nätet består av väldigt många sensorer. Det behövs troligen även spatiell återanvändning för att öka resursuttnyttjandet om nätet består av väldigt många sensorer. Trafikadaptiviteten och den spatiella återanvändningen kostar dock en hel del i overheadtrafik. CSMA däremot saknar denna overheadtrafik och är troligen mer lämpad för att hantera den typ av trafik som vi modellerat för datafusionen. Dock är denna typ av protokoll mindre bra ur resurssynpunkt eftersom noderna får svårare att gå ner i sov-mod samt att det kan behövas omsändningar p g a kollisioner på kanalen. Därmed är kanaltilldelning av TDMA-typ troligen att föredra ur energisynpunkt, medan CSMA är att föredra sett till trafiken i nätet.

Anpassad routing för flytt av agent i sensornät

Det vi noterat är att det verkar vara viktigt att flytten av agenten går relativt snabbt eftersom objektet som följs annars hinner försvinna bort i nätet. Om inte agenten befinner sig nära objektet innebär det att sensordata från andra sensorer

måste skickas längre väg för att nå agenten och ju längre bort objektet är från agenten desto längre väg måste sensordata skickas och nya vägar måste hittas i nätet för att nå fram till agenten, dvs. det i sin tur generar mer och mer routing-trafik och nyttotrafiken måste skickas längre och längre väg för att nå agenten.

I det flesta fall sker flytt av agent och datafusionen väldigt lokalt och berör i många fall endast grannoder, dvs. ett hopp bort, vilket inte kräver någon direkt routing. Dock är detta helt beroende av hur nättopologin ser ut, dvs. vilka noder som kan kommunicera med varandra, och hur objektet rör sig i nätet.

I studien frågar noderna om en väg vid det tillfälle de behöver en väg för flytta agenten eller för att skicka sensordata. Detta bör göras under en initieringsfas under vilken noderna sätter upp sina vägar. Därmed har noden redan en väg när den ska skicka data. En idé skulle kunna vara att utse vissa noder som potentiella ägare av agenten och sätta upp vägar till dessa i förväg så att flytten av agenten kan göras snabbt och effektivt, eftersom vägen dit agenten ska flyttas redan är känd. Dessa noder kan därmed väljas strategiskt utifrån lämplig position i nätet och inte flytta agenten ”slumpmässigt” i nätet. Detta liknar en uppbyggnad av nätet i form av kluster med dessa noder som klusterhuvuden.

Litteraturförteckning

[1] Smarta sensorer kräver smart ström. *Elektroniktidningen*, (4), mars 2005.

[2] L. Eschenauer och V. D.Gligor. A Key-Management Scheme for Distributed Sensor Networks. I *CCS'02*, s 18–22, 2002.

[3] R. et.al. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, s 40–50, mar 2002.

[4] L. Farman och U. Sterner. En generell trafikmodell för datafusion i trådlösa marksensornät. Teknisk rapport FOI-R–1470–SE, Totalförsvarets Forskningsinstitut, Linköping, Sverige, december 2004.

[5] L. Farman och L. Westerdahl. Säkerhet i trådlösa marksensornät - Förstudie. Memo 1113, december 2004.

[6] E. Hansson, J. Grönkvist och J. Nilsson. Intrångsdetektering i mobila ad hoc-nät. Användarrapport FOI-R–1375–SE, Totalförsvarets Forskningsinstitut, Linköping, Sverige, November 2004.

[7] J. Hill, R. Szewczyk och mfl. System Architecture Directions for Networked Sensors. I *ACM ASPLOS IX*, s 93–104, november 2000.

[8] M. Holmberg, L. Andris och R. Lennartsson. Slutrapport för projektet interaktiva adaptiva marksensornät (iam). Användarapport FOI-R–1450–SE, Totalförsvarets Forskningsinstitut, Linköping, Sverige, december 2004.

[9] D. Hwang, C. Lai och mfl. Energy tradeoffs in Distributed Sensor Networks. I *Ad-Hoc, Mobile, and Wireless Networks: Third Inernational Conference*, Vancover, Canada, juli 22-24 2004.

[10] K. Jacobsson. Arbetsplan samverkan LedsysT -FOI IPT NETS 2004. FMV LT90 03-00060101, april 2004.

[11] C. Karlof, N. Sastry och mfl. Tinysec: A Link Layer Security Architecture for Wireless Sensor Networks. I *SenSys'04, Baltimore, MD, USA*, 3-5 nov 2004.

[12] M. e. Lundberg. Power characterization of a bluetooth-equipped sensor node. I *REALWSN*, 2005.

[13] C. E. Perkins, E. M. Belding-Royer och S. R. Das. Ad hoc on-demand distance vector routing protocol. Internet-draft, IETF MANET Working Group, februari 2003.

[14] K. Persson. Routing med garanterad tjänstekvalitet i taktiska mobila ad hoc-nät. Metodrapport FOI-R–0886—SE, Totalförsvarets Forskningsinstitut, Linköping, Sverige, juni 2003.

[15] P. Skogler, M. Ulvklo och J. Nygårds. Utveckling av ett ramverk för eo/ir-överföring. En introduktion till samtidig lokalisering coh kartgenerering. Vetenskaplig rapport FOI-R–1031–SE, Totalförsvarets Forskningsinstitut, Linköping, Sverige, december 2003.

[16] Y. Zhenqiang, V. Srikanth och mfl. A routing framework for providing robustness to node failures in mobile ad hoc networks. *Ad Hoc Networks*, (2):87–107, 2004.