ESTUDO DE MECANISMOS DE SEGURANÇA PARA PROTEÇÃO DO ROTEAMENTO EM REDES DE SENSORES SEM FIO

SÉRGIO DE OLIVEIRA

ORIENTADOR: PROF. JOSÉ MARCOS SILVA NOGUEIRA

CO-ORIENTADORA: PROFA. HAO CHI WONG

ESTUDO DE MECANISMOS DE SEGURANÇA PARA PROTEÇÃO DO ROTEAMENTO EM REDES DE SENSORES SEM FIO

Tese a ser apresentada ao
Programa de Pós-Graduação em Ciência da Computação do
Instituto de Ciências Exatas da
Universidade Federal de Minas Gerais
como requisito parcial para a obtenção do grau de
Doutor em Ciência da Computação.

Belo Horizonte,

2008

Oliveira, Sérgio de.

Estudo de mecanismos de segurança para proteção do roteamento em redes de sensores sem fio [manuscrito] / Sérgio de Oliveira. – 2008.
xii, 181 f., enc. : il.

Orientador: José Marcos Silva Nogueira
Co-orientadora: Hao Chi Wong

Tese (doutorado) – Universidade Federal de Minas Gerais. Departamento de Ciência da Computação.

1. Computação – Teses. 2 Redes de Computadores – Teses. 3. Redes de sensores – Teses. I. Nogueira, José Marcos Silva. II Hao, Chi Wong. III Universidade Federal de Minas Gerais. Departamento de Ciência da Computação. IV. Título.

CDU 519.6*22

À minha filha Elisa,

Agradecimentos

Certamente este trabalho só pôde ser desenvolvido pela presença de Deus na minha vida. Sem Ele, eu nem teria acreditado no sucesso. Só ele é capaz de transformar um jovem sonhador, aluno de uma escola pública de uma cidade do interior, em um acadêmico prestes a defender uma tese. Só tenho a agradecer a Deus. Ele sabe sempre o que é melhor para todos nós. E nos dá força para superar os obstáculos. E já me deu tanta coisa, que não sou digno de pedir mais nada. Agradeço a força que Ele me deu e ponho em suas mãos todo o mérito do meu trabalho.

Agradeço também a Universidade Federal de Minais Gerais, referência entre as universidades nacionais e internacionais, por ter me dado essa oportunidade. Também agradeço ao Departamento de Ciência da Computação, através de seus professores, por manter sua qualidade incontestável e possibilitar o desenvolvimento desta tese, fornecendo todos os recursos e subsídios para tanto.

Agradeço especialmente ao meu orientador, professor José Marcos Nogueira, professor e amigo, me deu apoio em todos os momentos. E dedicou inúmeras horas ao desenvolvimento deste trabalho. Sem ele, este trabalho não teria o mesmo resultado.

Agradeço também à professora Hao Chi Wong, minha co-orientadora, pela sua ajuda neste trabalho.

Agradeço também a todos as estiveram juntos nestes anos de trabalho e dedicação. Eles entenderam quando eu precisei abdicar de momentos felizes para dedicar a este trabalho. Espero retribuir a todos, dedicando mais tempo à amizade, ao companheirismo e ao amor.

Ó Deus, escuta a minha oração, pois estou em dificuldades! Salva a minha vida,
pois tenho medo dos meus inimigos.
Protege-me dos planos que os maus fazem contra mim;
livra-me dos bandos de homens perversos.
Os maus afiam a língua como espada e apontam
como flechas as suas palavras cheias de veneno.
Eles agem depressa para espalhar as suas mentiras vergonhosas e destroem os
bons com calúnias covardes.
Eles se animam uns aos outros para fazer o mal;
falam dos lugares onde vão colocar as suas armadilhas
e pensam que ninguém pode vê-los.
Fazem planos cheios de maldade e dizem:
"Planejamos um crime perfeito." O coração e a mente
do ser humano são um mistério.
Salmo 64

Resumo

Redes de sensores sem fio (RSSF) estão mais sujeitas à ação de um inimigo que as redes convencionais devido às suas limitações de hardware e de energia e devido ao ambiente hostil em que podem ser inseridas. Esse cenário é muito favorável aos ataques de negação de serviço, especialmente na função de roteamento, que é crítica em uma rede. Este trabalho está focalizado na proteção do roteamento em RSSF. Para tanto apresenta um estudo sobre a segurança nas RSSF com enfoque no roteamento e propõe uma arquitetura de gerenciamento que possibilita estender o tempo de vida da rede pelo uso racional das diversas soluções de segurança e dois mecanismos de segurança. O primeiro mecanismo apresentado é um protocolo de estabelecimento de chaves proposto para que nós sensores vizinhos utilizem algoritmos criptográficos com o objetivo de garantir o controle de acesso no enlace, inibindo a presença de nós intrusos à rede. O segundo mecanismo é um algoritmo de roteamento com rotas alternativas para aumentar a resiliência da rede à presença de intrusos e, ainda, possibilitar a detecção de nós intrusos que estejam promovendo ataques de negação de serviço no roteamento. A arquitetura de gerenciamento de segurança apresentada possibilita que os mecanismos de segurança sejam usados apenas quando necessário, evitando, assim, um consumo desnecessário de energia quando não existe a presença de intrusos. Esse trabalho avalia o custo computacional, o consumo de energia, a eficácia e a escalabilidade das soluções apresentadas, indicando sua viabilidade para RSSF. Também são avaliados os impactos dos ataques e a eficácia dos mecanismos propostos para a defesa da rede contra os ataques. Os resultados indicam que a solução tem escalabilidade, é eficaz e o consumo adicional de energia pelas soluções apresentadas não afetam significativamente o tempo de vida da rede.

Palavras-chaves: Redes de sensores, segurança, DoS

Abstract

Wireless Sensor Networks (WSN) are more subject to enemy action than conventional networks due to their hardware and energy constraints and due to hostile environment in which they can be inserted. This scenario is too favorable to denial of service attacks, especially in routing function, one critical function in any network. This work is focused in routing protecting in WSN. It presents a study on WSN security focusing on routing and proposes an architecture management to extend the network lifetime by setting up the several security solutions and two mechanisms for security. The first mechanism is a key establishment protocol proposed to neighboring nodes use cryptographic algorithms to ensure the link layer access control, inhibiting intruder nodes presence in network. The second mechanism is a routing algorithm with alternative routes to increase network resilience against intruders' presence and, moreover, enable intruder detection of nodes running denial of service attacks on routing. The architecture of security management presented enables that security mechanisms are used only when necessary, thus avoiding unnecessary energy consumption when there is no intruders' presence. This work evaluates the computational cost, energy consumption, efficiency and scalability of the solutions presented, indicating their viability for WSN. The impact of the attacks and the effectiveness of the proposed mechanisms to protect the network against attacks are also analyzed. The results show the work has scalability, is efficient and the additional energy consumption don't decrease significantly the network time life.

Keywords: Sensor networks, security, DoS

Índice de Figuras

Índice de Tabelas

Sumário

Capítulo 1

Introdução

Novos modelos de rede estão trazendo novos paradigmas para a computação, com novas aplicações e necessidades. A computação ubíqua[1], que representa o uso de computação nos diversos ambientes e situações, não se restringindo aos ambientes computacionais convencionais, trouxe inúmeras possibilidades de aplicações computacionais, entre elas as Redes de Sensores sem Fio.

Redes de sensores sem fio (RSSF) são redes *ad hoc* formadas por elementos miniaturizados, chamados nós sensores, equipados com um conjunto de sensores e capazes de comunicação sem fio [2]. Podem conter desde algumas unidades até milhares de nós. Essas redes contam com um elemento central, conhecido como estação base ou nó sorvedouro, tem o objetivo de recolher as informações coletadas para apresentação aos usuários da rede. A estação base pode também enviar comandos aos nós para solicitar informações, bem como configurar os nós ou funções da rede.

Vários tipos de aplicações são propostos para as RSSF, normalmente relacionados ao monitoramento de ambientes, com finalidades diversas. Entre as aplicações, podemos citar o monitoramento ambiental, visando à obtenção de dados sobre a flora e a fauna, o controle de incêndios, em edifícios, florestas e indústrias, e a espionagem militar, coletando informações em campos de batalha ou em território inimigo.

Um requisito dos mais importantes das RSSF é a necessidade de baixo consumo de energia. Devido a seu tamanho reduzido, as fontes de energia são muito limitadas e as mais utilizadas são pequenas baterias, dificilmente substituíveis ou recarregáveis. Assim, para garantir o funcionamento do nó por um período de tempo especificado, é preciso que todos os seus componentes sejam projetados para um consumo adequado de energia, em geral muito baixo.

Como nas redes convencionais, a comunicação nas redes de sensores é modularizada em camadas formando uma pilha de protocolos. Devido à simplicidade do hardware, as funções são mais simples e otimizadas que nas

redes convencionais. A camada de rede tem papel especial, devido às características da rede, que deve incluir a capacidade de auto-organização, mesmo sem prévio conhecimento da localização específica de cada nó na rede. O roteamento deve ser proposto em função dessas características.

A função de roteamento é uma função de rede essencial nas RSSF. Ela pode ser dividida em duas subfunções: estabelecimento de rotas e encaminhamento de pacotes. O estabelecimento de rotas tem por objetivo criar os caminhos entre os nós e a estação base. A função de encaminhamento tem por objetivo levar os pacotes até seu destino final. Em redes de sensores, o conceito de disseminação de dados pode substituir o conceito de encaminhamento, pois o objetivo do roteamento não é levar um pacote específico a um destino, mas sim seu conteúdo. O importante é que as informações cheguem ao seu destino e essas informações podem estar replicadas em diversos pacotes, obtidas por diversos nós diferentes, ou sumarizadas em poucos pacotes. Assim, a entrega de um pacote não é importante, se o dado por ele representado puder chegar a seu destino a partir de outro nó origem.

Segurança é um requisito essencial para todo tipo de rede sujeito à presença de intrusos. Como as RSSF normalmente são usadas em ambientes abertos, isso as torna muito vulneráveis à presença de intrusos [3]. Diversos mecanismos podem ser usados para estender requisitos de segurança, especialmente aqueles baseados em algoritmos criptográficos. Várias aplicações de RSSF possuem requisitos de segurança [4]: aplicações militares, monitoramento comercial e industrial e até mesmo o monitoramento civil, onde a privacidade precisa ser respeitada, são exemplos de uso de RSSF com requisitos de segurança.

A presença de um intruso num caminho por onde passam os dados com destino a um nó sorvedouro interferindo na função de encaminhamento pode inibir toda a produção da rede, prejudicando o alcance de seus objetivos. Este trabalho está focado em mecanismos de segurança em RSSF para proteção das funções de roteamento, especialmente o encaminhamento de pacotes. Mecanismos simples e eficientes de segurança em RSSF, que serão apresentados

nesta tese, podem ser suficientes para evitar a presença de um inimigo no roteamento.

Alguns mecanismos de segurança usados em redes convencionais, como criptografia por chave pública e o uso de um centro de distribuição de chaves não são viáveis para redes de sensores devido às suas limitações energéticas e computacionais.

1.1 Delimitação do problema

Antes de entrar diretamente no tratamento da proteção contra os ataques, é necessário delimitar o problema principal a ser tratado. Os ataques de negação de serviço em RSSF efetuados sobre a função de roteamento, impedindo o encaminhamento de pacotes, são alvo deste trabalho. Esses ataques podem ser altamente prejudiciais ao funcionamento da rede. Além disso, as abordagens para as redes de sensores tendem a ser muito diferentes daquelas existentes para as redes convencionais, devido às limitações existentes nos ambientes de RSSF.

Diversas aplicações de RSSF definem requisitos de segurança, especialmente disponibilidade. Nessas redes, é possível a existência de entidades com objetivos contrários aos objetivos da rede, que podem agir para inutilizar seus serviços. Por isso é importante protegê-las contra os ataques de negação de serviço, ataques contra a disponibilidade. Esses ataques podem ser executados com muita facilidade por um inimigo e, se não forem tratados, podem inutilizar toda a rede, ou grande parte dela.

Este trabalho está particularmente interessado nas redes de sensores planas, nas quais todos os nós têm funções semelhantes, homogêneas, com o mesmo hardware utilizado para todos os nós, com o número de nós participantes variando entre 50 e 1000 nós e aplicações que podem durar anos. Esse tipo de rede foi escolhido por ter um custo significativamente mais baixo de desenvolvimento, montagem e distribuição. Várias aplicações de redes de sensores podem utilizar esses tipos de nós, como monitoração ambiental e até mesmo uso militar. Esta tese considera, ainda, somente as RSSF estáticas, ou seja, sem movimentação dos nós, mas que permitem a adição e revogação dos nós e considera, ainda, a perda de nós pela exaustão de sua bateria. Essa

restrição exige o desenvolvimento de soluções mais específicas e eficientes, além de ser a opção mais destacada na literatura e com amplo campo de aplicação, na monitoração de áreas não habitáveis e com nós com recursos extremamente limitados, o que restringe sua movimentação.

Os ataques de negação de serviço em RSSF foram inicialmente apresentados por Wood e Stankovic [3]. Nesse trabalho, os ataques foram classificados de acordo com a camada da pilha de protocolos onde atuam. Karlof e Wagner [5] detalharam os possíveis ataques de negação de serviço no roteamento, indicando quais as propostas de algoritmos de roteamento estariam sujeitas aos referidos ataques. Os ataques do tipo negação de serviço no roteamento foram eleitos como o problema a ser tratado nesta tese, devido à sua abrangência e facilidade de execução, uma vez que um simples nós pode interferir no roteamento em uma das rotas principais, silenciando parte da rede.

Embora as camadas superiores não necessitem de fluxos de transmissão confiáveis nas RSSF, o serviço de roteamento é considerado como crítico. Em redes de larga escala, mensagens podem passar por muitos passos até alcançar seu destino. Infelizmente, quando o custo agregado de encaminhamento de um pacote aumenta, também aumenta a probabilidade de um ataque ser efetivado para descartar ou redirecionar o pacote ao longo do caminho [3]. A forma dinâmica de estabelecimento de rotas facilita a inserção de um nó intruso nas rotas principais. Como um único nó pode ser responsável por repassar os pacotes de vários outros nós, a presença de um intruso em uma rota principal pode afetar a disseminação de dados de muitos outros nós.

Na ocorrência de um ataque de negação de serviço no encaminhamento de pacotes, a rede pode se comportar de duas formas: defender-se do ataque e continuar funcionando normalmente, sem permitir o acesso do intruso à rede, tampouco sofrer os efeitos da sua presença, ou não se defender do ataque, o que pode levá-la a ter a produção reduzida, pelo silenciamento de alguns nós, ou até mesmo a interrupção total de seu funcionamento. Este trabalho busca investigar e propor mecanismos para eliminar o efeito da presença de um intruso executando um ataque de negação de serviço no roteamento.

Algumas considerações devem ser feitas para permitir a delimitação do problema. A primeira é que, devido à comunicação sem fio, o enlace de rádio não é seguro. Outra consideração válida é que se o proprietário da rede pode lançar nós autênticos, então um inimigo também pode lançar nós maliciosos com capacidades de hardware similares. Um inimigo pode também capturar nós legítimos, ler e escrever dados da sua memória, pois os nós não contam com proteção física e estão sujeitos a ataques que utilizam do acesso físico aos nós, ataques conhecidos como *Tampering*. Ele pode também substituir o código de um nó e até clonar nós legítimos que podem ter seu código substituído. Nós inseridos por um inimigo podem ainda, eventualmente, contar com um enlace de rádio de baixa latência e alta largura de banda [5]. Devido à larga escala da rede, porém, não será considerado que o inimigo possa eliminar ou adulterar todos os nós autênticos ou uma parcela significativa desses nós.

Algumas distinções entre as diversas possibilidades de nós maliciosos devem também ser consideradas. A primeira é que os nós maliciosos podem ser nós sensores limitados, como os demais elementos da rede, com as mesmas limitações de processamento, comunicação, memória e energia, ou podem ser nós mais poderosos, como computadores portáteis ou seus equivalentes. Esses últimos elementos têm processamento, memória e capacidade de comunicação em níveis muito superiores aos nós da rede, podendo usar esses recursos para efetuar os ataques.

A segunda distinção a ser considerada é em relação à presença de nós maliciosos na rede. Os nós maliciosos serão considerados como internos se puderem participar normalmente das comunicações da rede, conhecendo todas as chaves e protocolos necessários. Do contrário, se faltar alguma informação necessária para a participação nos protocolos de rede, os nós maliciosos são considerados como externos.

A proteção do roteamento contra ataques de negação de serviço não pode ser realizada pelas soluções convencionais existentes em outras redes, em razão de várias características peculiares das RSSF, incluindo a maior vulnerabilidade dos nós e as limitações computacionais e de energia dos nós. Algumas características aumentam a sua vulnerabilidade, como a comunicação

sem fio e a simplicidade do hardware. Adicionando isso à localização e operação da rede, muitas vezes em ambientes abertos e de acesso não controlado, a proteção dos elementos da rede contra *Tampering* fica inviável. Um inimigo pode estar presente no local e no momento da obtenção dos dados coletados, na sua transmissão sem fio, ou ainda atuando nos diversos protocolos utilizados nessas redes.

Mecanismos de segurança usados em redes convencionais, como criptografia, só podem ser usados em RSSF se observadas as restrições de processamento, energia e comunicação. Mecanismos baseados em algoritmos de criptografia de chave pública ou o uso de um centro de distribuição de chaves (KDC - *Key Distribution Center*) devem ser descartados, devido ao alto custo de processamento ou comunicação apresentados. Dessa forma, os algoritmos criptográficos estão limitados àqueles que usam chaves simétricas e a distribuição de chaves deve ocorrer com baixo custo de comunicação. Essas características exigem novas abordagens para que seja possível atender aos requisitos de segurança necessários a algumas aplicações.

Diversos mecanismos podem ser usados para evitar a ação de um inimigo em RSSF. O controle de acesso à rede pode impedir a entrada de nós intrusos, a detecção de intrusos pode apontar os nós intrusos em ação na rede bem como mecanismos de revogação de nós podem isolar os nós intrusos, de forma que sua ação não tenha mais efeito. A combinação de mecanismos de segurança é necessária para aumentar a proteção da rede e a imunidade a diversos tipos de ataque.

Diversos trabalhos já foram propostos na literatura para restringir a presença do inimigo. O efeito de nós maliciosos externos pode ser anulado pelo uso de algoritmos criptográficos [6], garantindo o controle de acesso a rede. O problema, nesse caso, resume-se a um gerenciamento adequado de chaves, incluindo-se a distribuição, armazenamento, revogação e renovação das chaves [7]. Para anular o efeito dos nós maliciosos internos, outras abordagens precisam ser utilizadas, como detecção e revogação de intrusos [8][9]. Abordagens adequadas de gerenciamento de chaves podem, ainda, impedir a descoberta das chaves por um inimigo, dificultando a inserção de nós maliciosos

internos. No terceiro capítulo desta tese, será feita uma revisão da literatura mostrando em detalhes as diversas abordagens conhecidas.

As soluções já existentes na literatura não tratam o problema de proteção do roteamento de forma suficiente. Algumas apresentam vulnerabilidades que podem ser exploradas por um inimigo. Outras exigem hardware mais poderoso. Outras, ainda, aumentam muito o consumo de energia dos nós, inviabilizando seu uso nas redes que apresentam hardware mais limitado. Embora alguns ataques sejam tratados de forma eficaz e sem aumentar o consumo ou a necessidade de hardware adicional, outros ainda permanecem sem boas soluções, especialmente os ataques conhecidos como *Wormhole* e *Hello Flood* [5].

Os objetivos desta tese serão mostrados a seguir, utilizando abordagens para a resolução do problema apresentado.

1.2 Objetivos

O objetivo principal deste trabalho é o estudo e proposição de mecanismos para a proteção da função de roteamento em RSSF. Esses mecanismos visam eliminar ou reduzir os efeitos de ataques do tipo negação de serviço que atuam na função de roteamento. Faz parte dos objetivos desta tese desenvolver métodos, técnicas e algoritmos para reduzir os efeitos dos ataques de negação de serviço sobre uma RSSF e permitir que a mesma continue funcionando normalmente, mesmo durante um ataque.

Um protocolo de roteamento seguro deve garantir alguns requisitos no estabelecimento das rotas: confidencialidade, autenticidade e integridade. Além disso, a função de encaminhamento de pacotes deve estar sempre disponível, operacional. O objetivo é garantir esses requisitos mesmo na presença de quaisquer tipos de inimigos.

O primeiro objetivo desta tese é fazer um estudo sobre os diversos tipos de ataques de negação de serviço que podem comprometer o serviço de encaminhamento de pacotes em uma RSSF. Esses ataques serão avaliados com o objetivo de verificar os seus alcances e efeitos, para ataques realizados isoladamente, de forma distribuída, ou ainda em conjunto com outros ataques.

O segundo objetivo desta tese é desenvolver mecanismos para eliminar o efeito de nós maliciosos externos. Ataques promovidos por nós maliciosos externos podem ser defendidos pelo uso de mecanismos de criptografia [6], através de um rigoroso controle de acesso na camada de enlace. Para tanto, um protocolo de distribuição de chaves é proposto, para que os mecanismos de criptografia [6] possam ser usados de forma satisfatória.

O terceiro objetivo é propor mecanismos de detecção e revogação de intrusos, com o objetivo de eliminar os nós maliciosos internos da rede e, conseqüentemente, seus ataques.

O quarto objetivo é manter o consumo de energia controlado, para manter a longevidade da rede. Como o gerenciamento de energia é crítico em redes de sensores, para atingir esse objetivo mecanismos de gerenciamento dos diversos elementos de segurança fazem-se necessários, permitindo seu uso de forma racional para não gerar demanda excessiva de energia, o que ocasionaria uma redução do tempo de vida na rede. Esta tese deve prever o gerenciamento de segurança, de forma a permitir o uso racional dos mecanismos de segurança, preservando o tempo de vida da rede.

1.3 Contribuições

A partir dos objetivos definidos para o trabalho de pesquisa desta tese, vários mecanismos foram pesquisados para reduzir ou eliminar os efeitos de ataques efetuados por nós maliciosos externos e internos. Para ataques externos, a execução do controle de acesso utilizando algoritmos criptográficos exigiu a proposição de um protocolo de gerenciamento de chaves com baixo consumo de energia e resistente a ataques. Para ataques internos, a solução recaiu sobre um protocolo de roteamento resiliente à intrusão e um algoritmo de detecção e revogação de intrusos. Todos esses mecanismos foram concebidos considerando as limitações das RSSF.

Assim, no desenvolvimento deste trabalho, as seguintes contribuições foram alcançadas:

1. Avaliação dos impactos de ataques de negação de serviço no roteamento em RSSF, incluindo os ataques apresentados na literatura

[5][3], de forma a indicar a extensão e a facilidade de efetuar e se defender de cada ataque;

2. Proposição e avaliação de uma arquitetura de gerenciamento de segurança que permite a configuração e o acionamento de diversos componentes de segurança, incluindo alguns apresentados neste trabalho e outros presentes na literatura, visando sua utilização apenas quando necessário e racionalização seu uso e seu consumo extra de energia;
3. Proposição, avaliação, implementação e testes de um novo protocolo de estabelecimento de chaves para controle de acesso na camada de enlace, que permite a cada nó estabelecer chaves com todos os seus vizinhos e, a partir de então, realizar o controle de acesso para impedir a entrada de nós intrusos;
4. Proposição e avaliação de um algoritmo de roteamento com rotas alternativas em RSSF, para aumentar a resiliência da rede na presença de intrusos e permitir a detecção de intrusos de forma eficiente.

A arquitetura de gerenciamento apresentada neste trabalho foi proposta visando à extensão do tempo de vida da rede, pois permite acionar e configurar os diversos componentes de segurança, incluindo os mecanismos apresentados nesta tese, bem como outros disponíveis na literatura, promovendo sua utilização apenas quando necessário. Sua aplicação possibilita manter as vantagens dos mecanismos de segurança, especialmente porque evita os ataques de negação de serviço aqui apresentados e preserva, também, o consumo de energia dentro do mínimo necessário diante das ameaças descobertas.

O protocolo de estabelecimento de chaves foi chamado de NEKAP, um acrônimo para *Neighborhood-based Key Agreement Protocol*. Esse protocolo permite o estabelecimento de chaves entre cada nó e aqueles considerados como seus vizinhos, por estarem no raio de alcance do rádio, de forma que as informações trocadas por esses nós sejam autenticadas com essas chaves, assim evitando a comunicação de nós intrusos. Esse protocolo foi implementado e avaliado quanto à segurança, escalabilidade, desempenho e disponibilidade.

O uso de rotas alternativas foi proposto visando a aumentar a resiliência da rede à presença de intrusos e ainda prover um mecanismo eficiente para a detecção de intrusos. Assim, toda informação que deve ser encaminhada por um nó tem dois caminhos para chegar à estação base. Caso um dos caminhos tenha um intruso, os dados podem ser encaminhados pelo caminho alternativo. Sobre esse mecanismo foi implementado, ainda, um algoritmo que permite a detecção de intrusos, através da análise do fluxo de pacotes em cada uma das rotas. O algoritmo para detecção de intrusos é apresentado neste trabalho e também representa uma contribuição importante, diretamente associada à proposição de rotas alternativas.

As contribuições deste trabalho podem ser destacadas no aumento da segurança no roteamento em RSSF, com o protocolo de estabelecimento de chaves, a avaliação do impacto de ataques de negação de serviço distribuídos e com uso de rotas alternativas.

1.4 Apresentação do documento

O conteúdo desta tese é apresentado em oito capítulos, desde a definição do problema, até a conclusão e apresentação dos trabalhos futuros.

Os três primeiros capítulos são introdutórios: este primeiro capítulo apresenta o problema a ser abordado nesta tese, seus objetivos e contribuições realizadas; o segundo capítulo apresenta todos os conceitos necessários para entendimento desse trabalho, incluindo a descrição das RSSF e uma discussão a respeito dos requisitos de segurança e ameaças existentes nesse ambiente; o terceiro capítulo apresenta os trabalhos relacionados.

O quarto capítulo, a partir do qual são apresentadas as contribuições deste trabalho, propõe uma arquitetura de gerenciamento de segurança para redes de sensores sem fio, definindo os principais componentes de segurança e as interações entre eles.

O quinto capítulo apresenta um estudo sobre o impacto dos ataques de negação de serviço distribuído nas RSSF. São realizadas simulações para verificar a redução da produção total da rede em função do número de intrusos. O sexto capítulo apresenta o protocolo de distribuição de chaves, bem como sua avaliação de segurança, desempenho e viabilidade. O sétimo capítulo apresenta

uma modificação em um algoritmo de roteamento da literatura, incluindo rotas múltiplas de forma alternada, abordagem que é usada para aumentar a resiliência e permitir a detecção de intrusos de forma eficiente, mesmo na presença de um grande número de intrusos. A validação da arquitetura de gerenciamento de segurança é apresentando no oitavo capítulo, que inclui sua avaliação no cenário de redes de sensores sem fio.

Por fim, o nono capítulo apresenta as conclusões e os trabalhos futuros vislumbrados no desenvolvimento deste trabalho. Esta tese conta ainda com um apêndice, onde são relacionadas às publicações obtidas com este trabalho.

Capítulo 2

Conceitos Preliminares

Este capítulo apresenta um breve tutorial sobre os conceitos importantes para esta tese. Serão apresentados modelos de organização de rede, modelos de comunicação de dados e princípios de segurança aplicáveis a essas redes.

2.1 RSSF

As redes de sensores sem fio surgiram para permitir o monitoramento de locais sem infra-estrutura estabelecida. Formadas por dezenas, centenas e até milhares de nós, as RSSF devem atender a alguns requisitos, como comunicação sem fio, auto-organização, autonomia e tamanho reduzido. Com o foco nesses requisitos, os nós sensores, protocolos e padrões foram desenvolvidos diferindo significativamente daqueles usados nas redes de computadores convencionais.

Redes de sensores sem fio possibilitam avanços importantes na forma de monitoramento de ambientes. Seu uso permite cobrir vastas áreas a baixo custo. As aplicações propostas são as mais diversas, desde espionagem em caso de guerra até a supervisão de áreas de preservação ambiental. Para cada aplicação, é necessário definir o número de nós sensores utilizados, o alcance do rádio, as informações a serem coletadas e os requisitos de segurança necessários.

Diversas aplicações estão previstas para RSSF e, entre elas, podemos citar:

- Localização de focos de incêndio e desmatamento em florestas: Sensores distribuídos por uma região florestal podem coletar e enviar informações diversas, como localização de focos de incêndio, ações de desmatamento e erosão e, até mesmo, distribuição da fauna;
- Espionagem em território inimigo: sensores depositados por aviões podem espionar regiões inimigas, coletando informações diversas, como temperatura, umidade, movimentações de tropas ou informações meteorológicas.

- Aplicações médicas: ambientes especiais de recuperação de pacientes podem contar com sensores que monitoram as atividades e sinais vitais do paciente. Podem estar presentes no ambiente ou no próprio paciente, de forma a disponibilizar um conjunto muito maior de informações para um diagnóstico mais preciso dos profissionais da saúde.
- Monitoração de trânsito: em grandes centros urbanos, sensores podem ser usados em diversos trechos de trânsito elevado, de forma a coletar informações que podem ser usadas para controlar semáforos, acesso a pontes e viadutos e até mesmo provocar o desvio de áreas sobrecarregadas.

Várias dessas aplicações possuem requisitos de segurança, pois podem contar com a presença de um inimigo, representado por algum elemento interessado no funcionamento incorreto da rede ou na sua paralisação.

Em todas as aplicações, o hardware a ser utilizado nas RSSF é bem mais simples que o hardware utilizado em computadores pessoais. Normalmente a configuração representa um sistema embutido, com microcontroladores que utilizam quantidades reduzidas de memória e alguns pinos de entrada e saída, os quais recebem os sinais dos sensores. O rádio tem curto alcance e normalmente é implementado em um único chip, que recebe e transfere os dados diretamente para o processador. Todos os componentes devem apresentar baixíssimo consumo de energia, de modo que uma fonte de energia barata e reduzida seja capaz de manter o sistema operante pelo período de tempo necessário para a aplicação.

2.2 Classificações de RSSF

As redes de sensores sem fio podem ser classificadas quanto a fatores como composição, organização funcional, mobilidade e missão. A definição de determinadas características para uma RSSF dá-se em função dos requisitos da aplicação a que se destina, custo e confiabilidade da rede. As classificações aqui apresentadas foram baseadas no trabalho de Ruiz *et al.* [10].

2.2.1 Composição

Uma RSSF pode ser constituída de diversos tipos de nó sensores, com características variadas de hardware e software. De acordo com as características construtivas dos nós sensores que compõe uma RSSF, ela pode ser classificada em:

- Rede homogênea, quando todos os nós são semelhantes, em termos de recursos de hardware e software. Os mesmos recursos computacionais e bateria são encontrados em todos os sensores. Embora sejam idênticos em sua estrutura, alguns nós sensores podem realizar funções especiais, diferindo em seu funcionamento dos demais nós;
- Rede heterogênea, quando há diferenças entre os nós sensores em termos de hardware e software. Nós sensores mais robustos podem ser usados para funções especiais, como roteamento de pacotes, armazenamento e controle de chaves criptográficas, e outras, exigindo maior poder computacional e energético.

A diferenciação dos sensores é uma alternativa que pode aumentar o custo da rede significativamente, mas ao mesmo tempo aumenta o poder da rede como um todo. Nós sensores com hardware mais robusto podem agregar outras funcionalidades e permitir a realização de tarefas que não podem ser executadas na grande maioria dos nós sensores.

Os nós que contam com hardware mais poderoso podem executar tarefas especiais para garantir os requisitos de segurança da rede. Soluções de segurança, como algoritmos baseados em chave pública, não podem ser executadas nos nós mais simples devido às suas limitações de hardware. Mas, possivelmente, nós mais poderosos em redes heterogêneas podem executar esses algoritmos, desempenhando papel de destaque nas soluções de segurança.

Redes homogêneas, no entanto, têm custo significativamente menor, além de ser mais facilmente distribuídas, pois não é necessário prever, a priori, os pontos de instalação dos nós mais robustos. Este trabalho vai considerar somente as redes homogêneas.

2.2.2 Organização funcional

De acordo com a funcionalidade que cada nó sensor assume na rede, uma RSSF pode ser classificada em:

- Rede plana, na qual a funcionalidade de todos os sensores é idêntica, ou seja, todos os sensores podem executar as mesmas tarefas, sejam elas de roteamento, coleta de dados, criptografia ou troca de chaves;
- Rede hierárquica, na qual alguns nós sensores assumem funções especiais, ou seja, existe divisão de funções entre os nós. Estas funções podem exigir um maior poder computacional e até mesmo um consumo maior de bateria.

A organização da rede depende também da sua composição. Redes heterogêneas podem ser hierárquicas na sua organização, centralizando algumas funções nos nós com maior poder computacional e energético. Redes homogêneas também podem ser hierárquicas, nomeando alguns nós para realizarem funções especiais. Como estas funções demandam maior consumo de energia, os nós podem se revezar na sua execução. Uma forma de escolher os nós que devem realizar funções especiais é realizar eleições periódicas, nomeando nós mais bem localizados e com maiores reservas energéticas para realizar tarefas especiais, que possam demandar mais energia.

A organização em grupo está presente nas redes hierárquicas. Um grupo, ou *cluster*, de nós sensores é a menor forma de organização. O nó responsável pelas funções especiais de um grupo é conhecido como cabeça do grupo, ou *cluster head*. O nó cabeça do grupo pode ser um nó com capacidade maior de processamento e energia, caso a rede seja heterogênea, ou um nó eleito para esta função, caso a rede seja homogênea.

As redes hierárquicas podem contar com nós com funções especiais nas soluções de segurança, como distribuição de chaves, ou encriptação e decriptação de mensagens. Mecanismos usados para economizar energia, como fusão ou agregação de dados, necessitam de nós com funções especiais, em redes hierárquicas. Se essas soluções são usadas em ambientes seguros, esses nós também precisam de funções especiais de segurança.

Este trabalho foi proposto para redes planas e homogêneas. As soluções de segurança para redes hierárquicas heterogêneas normalmente utilizam algoritmos de chave pública nos nós intermediários, o que facilita o projeto de segurança dessas redes. Redes hierárquicas homogêneas, por sua vez, não devem concentrar funções especiais de segurança em poucos nós. Isso porque nessas redes é necessário o rodízio dos nós com funções especiais, para evitar a sua exaustão de energia. Mas o rodízio das funções de segurança pode levar um nó intruso a assumir essas funções, possibilitando um ataque de maiores proporções. Assim, esse trabalho está focado nas redes planas e homogêneas, vislumbrando maiores contribuições nessas redes.

2.2.3 Mobilidade

Quanto à possibilidade de locomoção dos nós, uma RSSF pode ser:

- Estacionária, quando os nós ficam fixos no local onde foram depositados inicialmente. O depósito dos nós pode ser feito de forma aleatória, como por exemplo, através de um avião lançador, ou de forma organizada, manualmente, em localizações pré-determinadas.
- Móvel, quando os nós não têm posição fixa e podem ser movimentados a todo momento. Devido ao movimento, toda a topologia da rede é dinâmica. Os diversos protocolos de rede devem ser desenvolvidos prevendo a mobilidade.

As RSSF estacionárias podem permitir pequenos deslocamentos de nós sensores por ação de agente externo. Um animal, por exemplo, pode deslocar um nó sensor por certa distância. O deslocamento fará com que o nó sensor permaneça inoperante até que a rede seja reconfigurada. A ocorrência de etapas de reconfiguração pode reintegrar nós sensores deslocados à rede.

As RSSF móveis exigem soluções diferenciadas de segurança. Por exemplo, caso os nós detenham chaves individuais, os mecanismos de segurança devem garantir a validade das chaves em qualquer ponto da rede, ou o restabelecimento contínuo das chaves a cada deslocamento de nós. Essas limitações dificultam as soluções de segurança para essas redes. As redes estacionárias, por sua vez, podem contar com soluções mais simples.

2.2.4 Missão

De acordo com a missão para a qual é utilizada uma RSSF, podem ser necessários requisitos de segurança. As RSSF podem então ser classificadas de acordo com sua missão em:

- Redes de missão comum, onde as aplicações não exigem requisitos de segurança. Aplicável somente em aplicações em ambientes totalmente fechados, inalcançáveis à ação do inimigo;
- Redes de missão crítica, com aplicações que demandam requisitos de segurança.

É difícil conceber aplicações que não demandem um nível mínimo de segurança. A presença de invasores pode ser detectada em diversos tipos de aplicações de rede. Até aplicações simples de monitoramento ambiental, por exemplo, podem despertar interesse de empresas que exploram o ambiente como madeireiras e garimpeiros.

2.3 Arquitetura dos nós sensores

A arquitetura de hardware e software dos nós sensores deve ser desenvolvida considerando suas funcionalidades: coleta de dados ambientais e transmissão destes dados à estação base. O nó sensor deve ser descartável, uma vez que é difícil recuperá-lo no ambiente. O nó sensor deve também ficar oculto no ambiente, sem despertar a atenção de pessoas ou animais. Desta forma, os nós sensores utilizados em redes de sensores sem fio devem então atender aos seguintes requisitos:

- Baixo custo, para permitir o uso em grande número e também seu descarte ao fim do tempo de vida da bateria;
- Baixo consumo de energia, permitindo uma vida prolongada com uma bateria compacta;
- Tamanho reduzido, permitindo que fique oculto no ambiente;
- Comunicação sem fio, para transmissão dos dados até a estação base.

Para atender a estes requisitos visando à obtenção de plena funcionalidade de uma RSSF, um nó sensor deve contar com os seguintes elementos:

- Microcontrolador – Incluindo alguns portos de entrada e saída, memória de programa e de dados;
- Bateria – Para fornecer energia. Devem ser considerados volume, capacidade inicial e comportamento diante de variações de temperatura;
- Transceptor – Responsável pela comunicação sem fio;
- Sensores – Convertem grandezas do ambiente, como temperatura, pressão e outras, em grandezas elétricas a serem usadas pelos nós.

Esses elementos serão aqui apresentados visando detalhar as limitações que devem existir nas aplicações para RSSF. Essas limitações são extremamente relevantes para a definição de soluções de segurança, pois essas soluções devem ser capazes de executar no hardware disponível nos sensores.

2.3.1 Microcontroladores

Vários microcontroladores têm sido propostos para serem utilizados nos nós sensores. As propostas de microcontroladores devem atender aos requisitos acima mencionados relativos a baixo consumo e custo. Os principais microcontroladores utilizados nas propostas de nós sensores existentes estão mostrados na Tabela 2.1.

Tabela 2.1 - Microcontroladores comerciais usados em nós sensores

Microcontrolador	Palavra	Freqüência	Memória de programa	Memória RAM	Fabricante
ATMega 128 [11]	8 bits	8 MHz	128 K	4 Kbytes	Atmel
ATMega 8535 [12]	8 bits	8 MHz	8 K	512 bytes	Atmel
MSP430x149 [13]	16 bits	1 MHz	60 K	4 Kbytes	Texas

Todos esses microcontroladores apresentam consumo de energia muito baixo, sendo conhecidos como *ultra low-power*. A memória de programa é especificada em número de instruções e não em bytes, uma vez que cada microcontrolador possui tamanho específico para sua instrução.

Este trabalho leva em consideração apenas os nós sensores que utilizam os microcontroladores acima mencionados, ou outros equivalentes em preço e consumo de energia. Suas limitações serão assumidas, de forma que não serão

consideradas soluções propostas para outros tipos de nós sensores. O motivo para esta escolha recai sobre as aplicações previstas para as RSSF, suas características e funcionalidades, conforme mencionado acima.

O hardware simplificado desses microcontroladores conta com um conjunto reduzido de instruções. Dentre os microcontroladores apresentados, apenas o MSP conta com multiplicador. Essas limitações influem diretamente no tempo necessário para processar os algoritmos criptográficos que serão usados nesse trabalho.

2.3.2 Sistema operacional: Tiny OS

TinyOS [14] é um sistema operacional dirigido a eventos para RSSF. Tem sido amplamente utilizado graças às suas exigências mínimas de hardware, podendo facilmente ser executado em microcontroladores de oito bits e ocupando poucos kilobytes de código.

O projeto do Tiny OS é baseado na simplicidade e define: um protocolo de roteamento, descrito na seção 2.5.4, baseado em uma árvore geradora; um tamanho máximo de pacote de 36 bytes, conforme a Tabela 2.2, e endereçamento com uso dois bytes. A Tabela 2.2 mostra o formato do pacote enviado pelo Tiny OS para disseminação de dados e mensagens de controle.

Tabela 2.2 - Campos do pacote do TinyOS

Field	*Length*
Destination ID	2 bytes
Active message handler	1 byte
Group ID	1 byte
Data length	1 byte
Data	29 bytes (max)
CRC	2 bytes

Um suplemento ao TinyOS foi proposto por Karlof *et al*. [6], chamado TinySec, incluindo rotinas de criptografia para encriptação e assinatura das mensagens. Esse trabalho, porém, não contempla a distribuição de chaves, que deve ser implementada de alguma forma pra viabilizar seu uso em RSSF. Esta tese apresenta uma proposta de distribuição de chaves adequada para o uso com o TinySec.

2.3.3 Exemplos de nós sensores

Atendendo aos requisitos propostos para RSSF, alguns nós foram desenvolvidos e propostos comercialmente ou para experimentos científicos e tecnológicos. Alguns deles serão mostrados nesta seção.

Motes

Desenvolvidos pela Universidade de Berkeley, os nós sensores conhecidos como Motes foram propostos especificamente para RSSF. São encontrados em diversos tamanhos e capacidades, entre eles, os pioneiros: Macro Motes e Rene Motes. Alguns já estão disponíveis comercialmente, como o nó Mica Motes.

As versões comerciais mais recentes disponíveis são:

- Mica2 Motes: Nó sensor que utiliza duas baterias AA, com duração de até um ano em modo sleep, nas freqüências 433, 869 e 916 MHz, em rádio multifreqüência, microcontrolador Atmel Atmega 128L [11], dimensões de 58 x 32 x 7 mm, excluindo o compartimento das baterias e largura de banda de 38,4 Kbaud;
- Mica2Dot Motes: Nó sensor com as mesmas características do Mica2, exceto pelo tamanho reduzido, em forma de moeda, com 25 mm de diâmetro e 6 mm de altura.

SmartDust

Outro exemplo de nó sensor está sendo desenvolvido também pela Universidade de Berkeley sob o nome SmartDust [15]. O objetivo é desenvolver um nó sensor cujo volume não ultrapasse 1 mm^3. O desenvolvimento atual do projeto já conta com um sensor de 100 mm^3 cujas características são:

- CPU 8-bits, 4MHz, AT90LS8535
- Rádio de Comunicação de 916 MHz
- Largura de banda de 10 kbps
- Sistema Operacional Tiny OS
- Espaço ocupado pelo SO: 3500 instruções
- Espaço disponível para aplicações: 4500 instruções

EYES

O nó sensor desenvolvido pela University of Twente [16], Holanda, no projeto conhecido como Eyes, conta com as seguintes características:

- Microcontrolador MSP 430x149 da Texas Instruments, com 1MHz, memória RAM de 2kbytes e memória de programa de 60 K instruções;
- Rádio TR1001, operando na freqüência de 868.35 MHz;
- Bateria de lítio e dióxido de manganês, com capacidade para operar em modo ativo por 2,7 dias;
- Largura de banda de 115,2 Kbps.

Tmote Sky

O nó Tmote Sky [17], desenvolvido e comercializado pela empresa *Moteiv Corporation*, criada por ex-alunos da Universidade de Berkeley, conta com as seguintes características:

- Microcontrolador MSP 430 da Texas Instruments, com 8MHz, memória RAM de 10 kbytes e memória de programa de 48 K instruções;
- Comunicação compatível com o padrão IEEE 802.15.4, através de *transceiver* fabricado pela Chipcon, com largura de banda de 250 kbps;
- Sensores de humidade, temperatura e luz integrados.

Durante o desenvolvimento desta tese, alguns nós Motes Mica2 e Tmotes Sky estiveram disponíveis para implementação e testes. As soluções aqui apresentadas foram implementadas e testadas nesses nós usando o sistema operacional Tiny OS.

2.4 Autoconfiguração

O dinamismo das RSSF exige que estas se organizem de forma automática. A autoconfiguração é uma função da rede que permite a manutenção dos seus serviços em caso de perdas de nós, bem como a agregação de novos nós, aumentando assim a densidade da rede e a disponibilidade dos dados [10].

A autoconfiguração também é uma função de RSSF estacionárias, pois essas redes também permitem algum dinamismo. Para entender o dinamismo é necessário conhecer as etapas de funcionamento de uma RSSF:

1. Os nós sensores podem ser depositados manualmente, em locais previamente determinados ou serem lançados, como no caso de lançamento por avião. Nesse caso, sua disposição na rede deve seguir um modelo probabilístico;
2. Após a deposição dos nós sensores, eles devem descobrir informações sobre sua localização, como quais são os nós vizinhos, alcançáveis através da comunicação de rádio, e determinar os parâmetros necessários para seu funcionamento, incluindo as rotas para o fluxo de pacotes entre os nós sensores e a estação base;
3. Durante o funcionamento normal da rede, as baterias dos nós sensores vão se exaurindo, de forma que alguns nós param de funcionar. A densidade de nós sensores vai diminuindo e novas etapas de reconfiguração devem ocorrer periodicamente. Nós sensores responsáveis pelo roteamento tendem a exaurir sua bateria mais rapidamente;
4. Nós sensores podem ser adicionados através de novos lançamentos. Novas etapas de reconfiguração devem acontecer, para permitir a integração destes novos sensores à rede.

Durante o funcionamento da rede, a inserção de novos nós pode ser feita, especialmente para aumentar a densidade da rede possivelmente afetada pela perda de nós por término da carga da bateria.

Qualquer proposta para RSSF deve considerar a possibilidade de autoconfiguração em todas as fases descritas acima. O projeto de segurança deve, ainda, considerar a possibilidade de integração de novos nós, bem como a reconfiguração em caso de falhas. Caso não sejam consideradas todas as fases da rede, esta pode ter suas funcionalidades ou tempo de vida reduzido ou ainda, o invasor pode se aproveitar de uma possibilidade não coberta pelos mecanismos de segurança da rede para incluir nós adulterados com objetivos escusos.

2.5 Arquitetura de rede

O objetivo de aumentar a longevidade de uma rede de sensores pela via da economia de energia obriga o uso de um modelo simples e funcional para a rede. Na arquitetura de comunicação de uma RSSF são suportadas até cinco camadas: aplicação, transporte, rede, enlace e física.

Esta seção apresenta as principais propostas para as camadas física, enlace e rede. Isto se dá porque o objetivo deste trabalho é desenvolver mecanismos de segurança para redes de sensores sem fio e todas as propostas aqui apresentadas utilizam apenas as camadas inferiores do modelo de rede. Também será apresentada nesta seção a funcionalidade de autoconfiguração, que permite que a rede inicie sua operação em ambientes onde não é possível a intervenção humana.

O estudo de segurança envolve um estudo apurado do modelo de rede utilizado, com vistas a identificar possíveis pontos de falhas e garantir a segurança em cada camada.

2.5.1 Camada física

A camada física é constituída por um meio sem fio, utilizando um transceptor e possivelmente uma antena, com as opções:

- Óptica (laser), ou LED infravermelho, que tem como vantagem o baixo consumo e como desvantagem a necessidade de visibilidade direta para transmissão de dados; e
- Rádio-freqüência: algumas faixas de freqüências estão disponíveis, entre elas 315, 433, 869, 915 e 2400 MHz. Estas freqüências são reservadas para uso médico, privativo e pesquisas, ou para dispositivos de curto alcance.

Alguns *tranceivers*, como o CC1000 [18], possuem capacidade de variar o alcance de transmissão em função da energia gasta para transmitir. Este aspecto é especialmente interessante para alguns algoritmos de roteamento. Nesse tipo de dispositivo são oferecidas 30 possibilidades de configuração de potência de transmissão.

Características da camada física podem ser usadas para efetuar ataques. Wood and Stankovic [3] apresentam ataques de negação de serviço na camada física, destacando os ataques conhecidos como *Jamming* ou Interferência, onde um inimigo gera um sinal na mesma freqüência utilizada pelos nós.

2.5.2 Camada de enlace

Várias propostas têm sido apresentadas para a camada de enlace em RSSF. A subcamada de acesso ao meio (MAC) tem papel de destaque, pois depende do sistema de transmissão e vai influenciar diretamente o consumo de energia.

As primeiras propostas de protocolos para a camada de enlace utilizadas não foram projetadas especificamente para RSSF. Por isso, não atendem a uma série de requisitos de RSSF, como a ausência de infra-estrutura física e a necessidade de economia de energia. Entre essas propostas, estão as tecnologias de acesso usadas em telefonia celular, como TDMA e CDMA e as camadas de enlace de redes sem fio tradicionais, como 802.11, Hiperlan e HomeRF.

O 15º grupo de trabalho do IEEE 802, especializado na padronização de redes conhecidas como PAN, ou *Personal Area Networks*, no subgrupo 4, definiu o padrão IEEE 802.15.4, cujas características incluem baixa taxa de transmissão, com o objetivo de estender a durabilidade das baterias. Os protocolos de alto nível que utilizam esse padrão foram publicados com o nome de Zigbee. As principais características desse padrão são [19]:

- Três freqüências de operação: 2,4 GHz, 915 MHz e 868 MHz;
- Alcance máximo de 150 metros;
- Três classes de dispositivos: coordenador, roteador e ponto final. O primeiro pode atuar como estação base, o segundo como roteador e o terceiro apenas é capaz de coletar e enviar os dados coletados;
- Número máximo de nós igual a 65535.

Outra proposta que foi também foi testada em RSSF é a utilização do protocolo conhecido como *Bluetooth* [20]. Esse protocolo, inicialmente projetado para interligação de redes pessoais pequenas, com a ligação de um computador e seus periféricos, tem o número de nós limitados a 7 por célula, embora uma rede possa ser composta de várias células.

A Tabela 2.3 apresenta uma comparação entre o *Bluetooth* e o *Zigbee*. Observa-se claramente que os recursos exigidos pelo *Bluetooth* são maiores, bem como a largura de banda ofertada. Para o modelo de rede que será usado nesta tese, o *Zigbee* é mais adequado.

Tabela 2.3 - Comparação entre Bluetooth e Zigbee

Padrão	Padrão para as Camadas Inferiores	Taxa de transferência	Duração média das baterias	Memória necessária	Nós	Alcance
Bluetooth	802.15.1	1 Mbps	1 a 7 dias ininterruptos	Cerca de 250 KB	7	1 a 10 m
ZigBee	802.15.4	250 Kbps	100 a 1000 dias ininterruptos	4 a 32 KB	65535	100 m

O protocolo escolhido na camada de enlace pode ter implicações nos requisitos de segurança da rede porque alguns ataques de negação de serviço exploram vulnerabilidades da camada de enlace. Wood and Stankovic [3] também apresentam ataques de negação de serviço na camada de enlace, destacando os seguintes ataques:

- Colisão, no qual um inimigo envia uma mensagem sempre que percebe que outro nó também enviou, forçando uma colisão;
- Exaustão: colisões repetitivas forçadas por um inimigo podem causar a exaustão da bateria.
- *Unfairness:* usado em redes com esquemas cooperativos de prioridades. Nesse ataque, nós intrusos visam manter sempre a prioridade no acesso ao meio, impedindo a transmissão de outros nós.

O controle de acesso na camada de enlace, realizando o descarte dos pacotes não autenticados, pode evitar os ataques realizados nas camadas superiores. O descarte dos pacotes usados para a realização dos ataques impede sua efetivação. A proposta de Karlof *et al.* [6], conhecida como TinySec, implementa autenticação na camada de enlace, desde que sejam disponibilizadas chaves para uso nos algoritmos criptográficos. O protocolo apresentado no quinto capítulo desta tese, denominado NEKAP, desempenha essa função, permitindo esse controle de acesso e conseqüente defesa dos ataques nas camadas superiores.

2.5.3 Camada de rede

A única funcionalidade da camada de rede intensamente explorada na literatura para RSSF é o roteamento. As demais funcionalidades, como endereçamento e controle de congestionamento, não têm sido discutidas intensamente, especialmente pela sua simplicidade nas RSSF, e não terão influência neste trabalho.

A funcionalidade de roteamento será apresentada em seção à parte, devido à sua importância para o trabalho e conteúdo vasto apresentado na literatura.

Os esquemas de endereçamento utilizados hoje em RSSF utilizam-se basicamente de um identificador único por nó, associado antes do lançamento dos nós e que não apresenta qualquer associação com a topologia da rede. Os identificadores são usados na camada de enlace e também para identificação da origem de cada dado recebido. A eliminação de qualquer tipo de endereço visando a economia de energia foi proposta em [21].

2.5.4 Roteamento em redes de sensores sem fio

Para definir o protocolo de roteamento a ser utilizado, é preciso classificar as RSSF de acordo com a peridiocidade de envio das informações pelos nós sensores, em redes orientadas a eventos e redes de disseminação contínua. Nas redes orientadas a eventos, os nós sensores enviam os dados apenas quando eventos ocorrem, como a mudança nos valores coletados. Nas redes de disseminação contínua de dados, os nós enviam os dados coletados periodicamente para a estação base [22]. Protocolos de roteamento normalmente são desenvolvidos para atender apenas um tipo de rede, quase sempre visando à economia de energia. As redes de disseminação contínua contam com atualização também periódica das rotas, enquanto as redes orientadas a eventos normalmente só atualizam as rotas quando existem eventos a serem enviados [22].

Vários algoritmos de roteamento têm sido propostos para redes *ad hoc* e redes de sensores. É importante verificar as várias diferenças existentes entre RSSF e redes *ad hoc*, formadas por computadores portáteis. A primeira diferença é a mobilidade, nem sempre presente em RSSF. A segunda diferença

está nas limitações energéticas das RSSF, muito superiores às limitações das redes formadas por computadores portáteis. Por último, as RSSF são orientadas a dados, ao passo que redes de computadores portáteis são orientadas a usuário.

Os algoritmos de roteamento para RSSF podem ser classificados em dois grupos, algoritmos para redes planas e algoritmos para redes hierárquicas [10]. Em uma rede plana todos os nós desempenham as mesmas funções, incluindo roteamento. Uma rede hierárquica contém nós especiais responsáveis por tarefas diferenciadas dentro do roteamento, especialmente atuando como roteadores. O principal algoritmo de roteamento para RSSF será apresentado a seguir.

Tiny OS beaconing

O protocolo *Tiny OS beaconing* [5], utilizado no sistema operacional Tiny OS, constrói uma árvore geradora com raiz na estação base. Todos os dados coletados são enviados para a estação base, único destino dos pacotes de dados.

O estabelecimento das rotas funciona da seguinte maneira: a estação base envia, periodicamente, pacotes de atualização de rotas em modos de difusão. Esses pacotes de atualização são conhecidos como *beacons*. Cada nó escolhe como sua rota de saída o nó do qual recebeu o *beacon* pela primeira vez. O *beacon* é, então, repassado para todos os vizinhos, em modo de difusão.

Os nós não mantêm tabelas de roteamento. Os nós conhecem apenas qual dentre seus vizinhos é o nó de saída, mais próximo da estação base. A comunicação nó a nó não é prevista no roteamento e só é possível entre nós vizinhos. A estação base só pode enviar mensagens para os nós em modo de difusão.

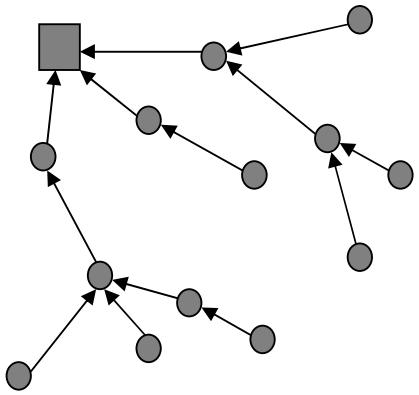

Figura 2.1 - Árvore gerada pelo Tiny OS

O protocolo Tiny OS beaconing é baseado no princípio da propagação da informação, apresentado em [13]. Devido à sua simplicidade, é

usado nos nós Mica2 e demais nós que usam o Tiny OS. O sexto capítulo desta tese apresenta uma técnica de detecção de intrusos baseada no uso de rotas alternativas sobre esse protocolo. O protocolo *Tiny OS beaconing* foi escolhido como modelo para este trabalho, devido ao fato de já ter sido validado para uso nos nós sensores.

2.6 Segurança em RSSF

O uso de mecanismos de segurança em RSSF se justifica na quase totalidade das aplicações, especialmente as aplicações militares, industriais, ou aquelas nas quais informações pessoais são tratadas pelos nós dos sensores, sendo necessário manter a privacidade das pessoas monitoradas.

Os requisitos de segurança de informação para uma rede de sensores sem fio dependem da sua aplicação. Usualmente, os seguintes requisitos podem ser necessários: confidencialidade, autenticidade, integridade e disponibilidade. A determinação da aplicação a ser utilizada indicará quais requisitos serão necessários.

Os requisitos de RSSF impõem restrições sobre os mecanismos de segurança que pode ser usados. Em redes convencionais, ligadas diretamente através de fios ou fibra ótica, os algoritmos de criptografia de chave pública e certificados digitais são usados para garantir vários requisitos de segurança. Entretanto, para RSSF, os recursos limitados dos nós sensores inviabilizam o uso de algumas soluções usadas em outras redes. Algoritmos de chave pública e certificados digitais são inviáveis nas RSSF devido ao baixo poder de processamento dessas.

Outra maneira bastante utilizada para garantir requisitos de segurança em redes convencionais é o uso de um terceiro elemento confiável. Esse elemento deve ser facilmente reconhecido por todos os outros elementos da rede, podendo atuar como uma autoridade certificadora ou como um centro de distribuição de chaves (KDC – *Key Distribution Center)*, enviando, pela rede, as chaves que forem solicitadas. A presença de um terceiro confiável simplifica o processo de distribuição de chaves, uma vez que basta que todo elemento da rede tenha uma relação de confiança com o terceiro confiável, para que este possa estender essa relação a quaisquer dois elementos da rede. Mas a

necessidade de manter baixo o consumo de energia restringe as operações de trocas de chaves pela rede através de um terceiro confiável, como um centro de distribuição de chaves. Com isso, novas soluções de segurança devem ser projetadas para esse tipo de ambiente, observando os requisitos existentes.

Outra restrição é o fato que os nós não podem contar com resistência contra captura e adulteração (*Tampering*), dada a simplicidade de hardware dos nós e a disponibilidade no meio ambiente. Assim, a captura de um nó pode comprometer a segurança da RSSF ao se revelarem informações como chaves, ou mesmo permitindo a reprogramação do nó, o que vai torná-lo uma ferramenta de ataque de um inimigo.

2.6.1 Ataques

Diversos tipos de ataques podem ser observados em RSSF. Os ataques incluem desde escuta e inserção de dados falsos, que têm comportamento similar às demais redes sem fio, até ataques do tipo negação de serviço, que são mais fáceis de serem executados e mais difíceis de serem neutralizados [3].

Algumas características especiais das redes de sensores sem fio permitem classificar os diversos tipos de ataques, especialmente quanto ao poder computacional e participação do invasor na rede.

Quanto ao poder computacional, os seguintes tipos podem ser observados:

- Ataques por nó sensor: neste caso, o invasor tem acesso a um ou mais nós sensores semelhantes aos sensores da RSSF e utiliza-se destes para realizar ataques;
- Ataques por computador portátil: aqui, o invasor utiliza um computador portátil com alto poder de processamento, energia e comunicação, implementando nele os protocolos de comunicação dos nós sensores, de forma a se fazer passar por um nó sensor.

Quanto à participação do invasor na rede, os ataques podem ser assim classificados:

- Ataques internos (*insider attacks*): realizados por um nó sensor membro da rede. Normalmente ocorre quando o inimigo adultera um nó da rede e insere código malicioso que efetua o ataque [23];

- Ataques externos (*outsider attacks*): realizados por um computador portátil ou nó sensor estranho à rede, que não dispõe de informações que são importantes para o funcionamento da rede, como as chaves. Assim, o atacante não consegue provar para os demais nós que é um membro da rede;

Caso o inimigo descubra as informações secretas de um nó, como as chaves, e insira estas informações em outro nó estranho à rede, de forma que este nó seja aceito como membro da rede, é caracterizado um ataque interno.

2.6.2 Arquitetura de segurança

Diversos mecanismos de segurança são encontrados na literatura. Aqueles de maior interesse para este trabalho foram organizados e classificados em componentes de acordo com seus objetivos.

Existem soluções de segurança preventivas e reativas. Algumas delas, como criptografia, são preventivas e impedem a ação de intrusos. Outras, como indicadores de detecção de intrusos a problemas de segurança, permitem ações corretivas, como a revogação de intrusos.

Esse trabalho considera a possibilidade de aplicação de encriptação salto-a-salto e fim-a-fim, a utilização de técnicas de gerenciamento de chaves, a existência de mecanismos de detecção de intrusos, o roteamento seguro, a fusão segura de dados, bem como um esquema de revogação de nós. A seguir, esses mecanismos serão definidos.

2.6.3 Técnicas criptográficas

A criptografia é um mecanismo extremamente importante para viabilizar diversos componentes na arquitetura de segurança de qualquer sistema. Pode ser usada para garantir privacidade, por meio de encriptação dos dados e autenticidade e integridade das informações utilizando assinatura dos dados. A escolha do algoritmo de criptografia a ser usado, bem como o método de distribuição de chaves, deve atender aos requisitos das RSSF consideradas, especialmente o poder computacional limitado e a necessidade de economia de energia. Algoritmos de criptografia de chave pública requerem alto poder

computacional e por isso devem ser evitados. Assim, a escolha do algoritmo deve recair sobre algoritmos de chave simétrica.

A encriptação e a assinatura podem ser processos fim-a-fim, realizados apenas uma vez por mensagem, ou processos salto-a-salto, realizados cada vez que uma mensagem atinge um nó de repasse.

Para os dados coletados, enviados diretamente dos nós para a estação base, a encriptação e assinatura dos dados fim-a-fim podem ser suficientes para garantir os requisitos de segurança. A encriptação e assinatura fim-a-fim exigem apenas o compartilhamento de uma chave por nó com a estação base, que pode ser pré-distribuída [4]. Naquele trabalho, Perrig *et al.* apresentam o uso de encriptação e assinatura fim-a-fim para RSSF no protocolo denominado SNEP. Esse tipo de mecanismo, todavia, limita o processamento interno e inibe o uso de opções como fusão de dados [24], onde os dados devem ser processados internamente na rede. A encriptação e assinatura de dados fim-a-fim também não podem ser usadas para mensagens enviadas em difusão, ou mensagens que devem ser tratadas a cada passo do roteamento, como as mensagens de estabelecimento de rotas.

Encriptação fim-a-fim protege os dados contra ataques de espionagem durante sua transmissão para a estação base. Nessa técnica, para transmitir uma mensagem de um nó para a estação base ou vice-versa, somente o nó e a estação base precisam de encriptar/decriptar a mensagem, o que implica em menor processamento da rede que na encriptação salto-a-salto.

A assinatura também pode ser um processo fim-a-fim, verificada somente pela estação base, ou salto-a-salto. A assinatura de uma mensagem causa processamento extra no momento e local de sua geração, maior utilização da rede na transmissão e processamento extra no momento da verificação. O uso de técnicas de assinatura e codificação pode evitar a inserção de pacotes falsos e a adulteração de mensagens.

Quando os métodos criptográficos fim-a-fim não podem ser usados ou são insuficientes, uma alternativa é usá-los a cada passo do roteamento na modalidade de encriptação e assinatura salto-a-salto. Nesse caso, é necessário um compartilhamento de chaves entre os vários nós que precisam se comunicar

diretamente para a execução do roteamento. Essa abordagem pode evitar a entrada de intrusos no roteamento, caso eles não possuam as chaves necessárias para a realização dos processos criptográficos. Encriptação salto-a-salto implica em mais processamento nos nós intermediários, mas torna possível outros tipos de funções de processamento da rede necessárias, como a fusão de dados. Para usar encriptação salto-a-salto, os nós têm de compartilhar chaves com seus vizinhos.

TinySec

TinySec [6] é uma solução de segurança totalmente implementada, baseada no TinyOS, um sistema operacional para RSSF. O TinySec especifica métodos criptográficos para assinar e encriptar mensagens, provendo propriedades de autenticação e privacidade para uma RSSF, ao custo de poucos bytes adicionais e baixo custo de energia. O TinySec prevê duas modalidades: TinySec-Auth, com a inclusão de uma assinatura para autenticação e integridade e o TinySec-AE, com assinatura e encriptação salto-a-salto.

O TinySec requer o compartilhamento de chaves entre os nós vizinhos e diversas técnicas podem ser usadas, como compartilhamento global de chaves ou compartilhamento probabilístico de chaves. O compartilhamento global de chaves não é seguro porque um único nó capturado pode revelar a chave global. Técnicas probabilísticas, onde dois nós compartilham a chave com certa probabilidade, também não são boas porque canais de comunicação podem não ser estabelecidos quando um ou mais nós do par não tiverem a chave própria.

Sendo uma técnica do nível de enlace, o TinySec executa salto-a-salto e seu custo será considerado quando a criptografia salto-a-salto estiver habilitada.

Em relação à sobrecarga de rede, observa-se que o tamanho do cabeçalho do pacote, normalmente sete bytes no TinyOS, aumenta para oito bytes no TinySec com o modo de autenticação habilitado (TinySec-Auth mode). O campo de dados de um pacote TinyOS é de 0 a 29 bytes. No modo de encriptação, chamado TinySec-AE, o campo de cabeçalho tem tamanho de 12 bytes, incluindo um vetor de inicialização para prover segurança semântica na criptografia.

2.6.4 Gerenciamento de chaves

O uso correto dos algoritmos de criptografia necessita da manutenção do segredo das chaves perante o inimigo. Este segredo deve ser mantido durante a geração, distribuição e armazenamento das chaves. A distribuição de chaves é a principal tarefa de gerenciamento de chaves, que deve controlar o armazenamento, o acesso, o tempo de validade, o envio das chaves e sua revogação, quando expirada sua validade ou quando descoberta por um inimigo.

Na distribuição, as chaves não devem trafegar pela rede sem que seja garantida a confidencialidade. Do contrário, o inimigo poderia escutá-las. Para manter a confidencialidade, são necessárias outras chaves. A solução pode recair então sobre duas possibilidades: ou as chaves são previamente distribuídas ou é utilizado um esquema de chave pública, inviável em redes de sensores, devido à complexidade de seus algoritmos e conseqüente alto custo computacional. A chave usada para garantir a confidencialidade na distribuição das chaves é chamada KEK, *Key Encryption Key*. A chave utilizada durante o processo de criptografia dos dados é chamada chave de sessão. A utilização de uma KEK permite que a chave de sessão seja renovada periodicamente, dificultando a sua descoberta.

A distribuição de chaves então pode ocorrer de três formas:

- Pré-distribuição de todas as chaves utilizadas;
- Pré-distribuição das KEK e posterior distribuição das chaves de sessão;
- Uso de chaves públicas;

A pré-distribuição de todas as chaves de sessão utilizadas é o esquema mais simples e com menor custo de comunicação. Como as chaves não trafegam pela rede, não geram sobrecarga de comunicação. A desvantagem da pré-distribuição é que as chaves de sessão não podem ser alteradas e a descoberta de uma chave compromete a comunicação do canal que utiliza essa chave indefinidamente. Além disso, o número de chaves necessárias em uma rede de larga escala pode ser inviável para a memória escassa dos nós sensores.

Uma técnica muito comum nas redes de computadores é o uso de pré-distribuição de KEK. Essa abordagem permite que a distribuição de chaves possa ser realizada utilizando apenas algoritmos de chave simétrica. Nesta abordagem, é comum a presença de um servidor conhecido por KDC (*Key Distribution Center*) que compartilha uma chave com cada elemento da rede. Assim, esse servidor pode ser usado para geração e distribuição de chaves de sessão, que podem ser usadas em comunicação par-a-par entre dois elementos de rede. A desvantagem dessa técnica é que muitas mensagens são trocadas pela rede para que as chaves de sessão possam ser distribuídas de forma segura. A sobrecarga gerada por essas mensagens inviabiliza seu uso em RSSF.

O uso de algoritmos de chave pública é a abordagem mais elegante e eficaz para a distribuição de chaves. Algoritmos de chave pública permitem a divulgação de uma chave de um par, sendo a outra mantida em segredo. Assim, a chave divulgada, conhecida como chave pública não precisa ser enviada de forma confidencial. Apenas é necessário garantir a autenticidade da chave pública, que pode ser trocada pelo invasor durante sua divulgação. A autenticidade pode ser garantida através do uso de certificados emitidos por autoridades certificadoras. O suporte a distribuição de chaves públicas, incluindo a presença de autoridades certificadoras é conhecido por infra-estrutura de chave pública ou PKI, *Public Key Infrastructure*.

Os nós sensores utilizados neste trabalho não suportam a execução de algoritmos de chave pública, mas somente algoritmos de chave simétrica. Os nós sensores Motes têm processador com estrutura bastante simples, com operações de apenas 8 bits, como visto na seção 2.3.3 - Exemplos de nós sensores. Algoritmos de chave pública demandam alto poder de processamento, operações de multiplicação e divisão, não disponíveis nos microcontroladores utilizados neste trabalho.

Possivelmente, no futuro, os nós sensores serão capazes de executar algoritmos de chave pública, mantendo as mesmas características e funcionalidades dos nós sensores Motes. Nós sensores mais recentes tem sido propostos com base no o microcontrolador MSP430F149 [13] que, por contar com multiplicadores de 16 bits pode ser usado para criptografia de chave

pública, embora o tempo e a energia gastos nesse processo ainda sejam proibitivos. Microcontroladores da família MSP430x33x em [25] já foram utilizados para este fim. Porém, a execução de algoritmos de chave pública ainda não pode ser realizada de forma eficiente. Nós sensores capazes de executar criptografia de chave pública de forma eficiente já foram apresentados na literatura [26]. Esses nós, porém, estão além das características dos nós propostos neste trabalho e suas aplicações e objetivos diferem completamente. É fato também avaliado nessa literatura, que o uso de algoritmos de chave pública pode ter alto custo energético.

Vários trabalhos apresentados até o momento na literatura apresentam chaves pré-distribuídas junto aos sensores, para evitar sobrecargas de rede e de processamento. A forma de pré-distribuição, no entanto, é variável e várias abordagens têm sido propostas [6], entre elas o uso de:

- Chave global: todos os sensores conhecem uma chave mestra global com a qual são geradas todas as outras chaves utilizadas nos algoritmos de criptografia. Assim todos os nós sensores compartilham as mesmas chaves e a descoberta da chave mestra permite ao inimigo quebrar todos os algoritmos de criptografia utilizados;
- Chave por nó sensor compartilhada entre estação base e o nó sensor: cada sensor possui uma chave K, conhecida apenas pela estação base. A estação base terá que armazenar e processar tantas chaves quantos são os nós sensores. Dessa forma, a descoberta de uma chave inutiliza apenas a comunicação entre a estação base e o nó cuja chave foi descoberta;
- Chave por conjunto de nós: é uma alternativa intermediária na qual a descoberta de uma chave compromete apenas um conjunto de sensores. Porém, esta solução tem algumas dificuldades, como distribuir a chave para o conjunto de nós vizinhos de forma segura. As chaves não podem ser previamente distribuídas porque não é conhecida, a priori, a localização dos nós sensores e seu conjunto de vizinhos;

- Chave par-a-par: uma chave para cada par de nós que necessitam se comunicar. Como não é conhecida nenhuma informação prévia sobre a localização e a topologia da rede, essa solução só é aplicável para redes muito pequenas, uma vez que para redes grandes, o número de chaves a serem armazenadas extrapola a capacidade dos nós.

Chaves par-a-par são mais difíceis de estabelecer porque a topologia da rede não é previamente conhecida e restrições de memória impedem total compartilhamento de pares de chaves.

Diversos protocolos de estabelecimento de chaves par-a-par são encontrados na literatura. Técnicas probabilísticas [26][28][29] são boas propostas, mas não asseguram estabelecimento de chaves entre todos os nós vizinhos. Sem chaves, alguns canais de comunicação não poderiam ser usados, aumentando o caminho da rede e, consequentemente, o consumo de energia. Técnicas determinísticas [30][7] assumem um início confiável de chaves globais compartilhadas para estabelecer todas as chaves necessárias para os pares.

Esta tese apresenta um protocolo de estabelecimento de chaves no quinto capítulo, denominado NEKAP. Esse protocolo permite o estabelecimento de chaves entre quaisquer dois nós vizinhos para que eles possam utilizar encriptação e assinatura salto-a-salto, visando, principalmente, à proteção contra os ataques de negação de serviço no roteamento.

2.6.5 Sistemas de detecção de intrusos e revogação de nós

A detecção de intrusos em RSSF deve lançar mão de mais técnicas diferentes que nas redes convencionais, devido às diferenças nos modelos, ataques e recursos. Dois tipos de técnicas podem ser utilizados para detecção de intrusos em RSSF: centralizadas ou descentralizadas. Na técnica centralizada, a estação base é responsável pela execução da detecção de intrusos, iniciando o processo pela coleta de informações da rede, especialmente a produção dos nós sensores (mapa de produção). A estação base possui um grande conjunto de informações à sua disposição, o que facilita o processo de detecção. Na técnica descentralizada, alguns ou todos os nós executam operações simples para detectar intrusos [8] [31]. A grande vantagem dessa técnica é a disponibilidade

instantânea da informação, visto que os nós podem detectar os ataques exatamente no momento em que eles ocorrem.

A detecção de intrusos é normalmente seguida da revogação dos nós intrusos. A revogação é a exclusão do nó da rede, tornando impossível para ele a comunicação com seus vizinhos. Esse processo deveria ser autenticado para evitar a revogação de nós autênticos por intrusos. Como os nós não são protegidos contra violação física no modelo utilizado nesse trabalho, é mais seguro permitir somente à estação base promover a revogação de nós. De outra forma, um nó intruso autenticado pela rede, provavelmente originado de uma violação física, poderia isolar nós autênticos, promovendo outros tipos de ataques de negação de serviço. O protocolo µTesla [4] pode ser utilizado para autenticar mensagens de revogação.

2.6.6 Roteamento seguro

Três mecanismos podem ser considerados para proteger o roteamento: autenticação em difusão, fim-a-fim ou salto-a-salto, durante o estabelecimento de rotas [4][30], detecção de intrusos no roteamento [8][31] e rotas alternativas ou redundantes para aumentar a resiliência contra intrusão [32].

A autenticação do *beacon* enviado para a criação de rotas pode impedir que um inimigo se faça passar pela estação base, no ataque conhecido como *Synkhole*. Para tanto, o beacon deve ser autenticado fim-a-fim. Ou seja, a estação base assina o beacon e todos os nós que o recebem sabem que ele foi enviado pela estação base autêntica. Isso pode ser feito usando o protocolo µTesla, proposto em [4].

Entretanto, a autenticação fim-a-fim não impede que um nó intruso receba o *beacon* autêntico e o repasse, com o objetivo de se inserir no roteamento, em ataques como *Black hole*. Nesse ataque, o inimigo se insere no roteamento, mas depois não repassa nenhum pacote recebido. Para restringir esses ataques, o *beacon* deve ser autenticado também a cada salto, garantindo, assim, que o nó que está sendo assumido como nó de repasse para a estação base é um nó autêntico. A autenticação par-a-par é um dos objetivos desta tese, implementado no protocolo NEKAP, que será visto no quinto capítulo.

2.6.7 Fusão segura dos dados

Eventualmente as leituras dos sensores podem ser imprecisas ou até mesmo inúteis. Mesmo sob condições ambientais perfeitas, os sensores podem não prover leituras absolutamente perfeitas. As RSSF freqüentemente possuem um grande número de nós sensores, trazendo um novo desafio de escalabilidade relacionado ao consumo desnecessário de energia provocado por transmissões de dados redundantes e por colisões. A fusão de dados, técnica em que diferentes dados coletados por vários nós são transformados na rede antes de serem enviados ao usuário, possui pelo menos dois fatores que tornam importante a sua utilização em RSSF. O primeiro consiste na obtenção de leituras de maior precisão, tornando a rede mais robusta e menos vulnerável a falhas e imprecisões de um único nó sensor. O segundo fator é a economia de energia pela da redução da quantidade de mensagens e de dados que são transmitidos pelos nós sensores [33].

Soluções de segurança podem interferir na fusão de dados. A encriptação fim-a-fim inviabiliza fusão de dados porque impede que qualquer nó intermediário tenha acesso à informação para executar a operação. Além disso, a fusão de dados pode confundir os mecanismos de detecção de intrusos, como o *watchdog* [34]. Nesse mecanismo, o nó mantém seu rádio em modo de recepção para verificar se o nó responsável pelo repasse de suas mensagens está cumprindo sua função, repassando os pacotes corretamente. Com a fusão de dados, como esse repasse não acontece diretamente, o *watchdog* não funciona. Assim, em aplicações de segurança crítica, a fusão de dados tem de ser desabilitada para se utilizar encriptação fim-a-fim e sistemas de detecção de intrusos.

2.7 Modelo de rede adotado

Esta seção apresenta o modelo de RSSF a ser usado nesse trabalho. RSSF incluem uma variedade muito grande de configurações. A ampla variedade de hardware para nós sensores, diferentes características da estação base, possibilidade de mobilidade e a diversidade de aplicações impedem que a caracterização do problema seja feita de forma única. O problema de segurança

em redes de sensores sem fio, considerado neste trabalho, será tratado considerando algumas premissas básicas:

- A estação base é confiável, ou seja, não está sujeita a ataques, e não apresenta restrições de energia e processamento como os nós sensores;
- A estação base é ou origem ou destino de todas as mensagens de dados da rede;
- Os nós não são confiáveis, pois não são resistentes à adulteração física (*tamper resistance*);
- A energia do nó sensor é um recurso escasso. Assim, toda funcionalidade presente deve ter relação benefício/energia maximizada;
- Os nós sensores contam com recursos computacionais muito limitados, o que torna proibitivo o uso de algoritmos pesados, como criptografia de chave pública;
- O ambiente é aberto e hostil, que favorece a captura de nós e inserção de nós adulterados;
- A comunicação é sem fio, que favorece a presença de escutas e propagação de interferência;
- Não existe conhecimento prévio de topologia de rede, incluindo a localização, devido à distribuição aleatória dos nós.

Dois modelos de distribuição de nós foram utilizados: o modelo conhecido como colméia, determinístico, no qual cada nó tem exatamente seis vizinhos eqüidistantes e o modelo aleatório, probabilístico, no qual cada nó pode ser lançado em qualquer posição da rede.

No modelo aleatório, os nós foram distribuídos em uma área de 100x100 unidades de distância. As duas coordenadas x e y são geradas aleatoriamente com valores entre 0 a 100. O alcance do rádio era modificado para variar o número médio de vizinhos de forma empírica. O modelo colméia será apresentado com mais detalhes a seguir.

Para ficar condizente com nós sensores reais, o trabalho considera os nós sensores Berkeley Mica2 Motes, limitados para somente 4 Kbytes de memória

RAM e 128 Kbytes de memória de programa [35]. A energia gasta para transmissão sem fio é maior que a energia consumida para recepção e processamento. Nesse nó, a transmissão com o alcance máximo, teoricamente até 100 m, consome 27 mA e a recepção consome 8 mA para uma taxa de transmissão de 38,4 Kbaud. O consumo do processador é de 8 mA para uma freqüência de CPU de 4 MHz. Todavia, é possível utilizar um modo de transmissão com alcance mais curto, reduzindo o consumo de transmissão. Por exemplo, para um alcance de cerca de 40 metros, um nó pode utilizar um modo de transmissão que consome apenas 12 mA.

Os mecanismos apresentados nesta tese, contudo, não estão limitados a esse nó. Vários outros nós, com características de hardware semelhantes, poderiam ser usados neste trabalho, obtendo resultados similares.

A evolução dos nós sensores pode apresentar nós com maior poder de processamento, maior disponibilidade de energia e mais baratos, permitindo que outros mecanismos de segurança pudessem ser propostos para essas redes. Entretanto, os microcontroladores de baixo poder computacional com comunicação sem fio de largura de banda e alcance reduzidos sempre serão aplicáveis, pois seu custo, cada vez menor, vai permitir o desenvolvimento de outras aplicações para os mesmos, que podem, ainda, utilizar os mecanismos desenvolvidos nesta tese.

2.7.1 Modelos de distribuição espacial

Diversas simulações e análises são feitas neste trabalho, nas quais é necessário estabelecer relações entre vizinhos. A distribuição geográfica dos nós pela região de sensoriamento é um fator extremamente importante para definição dos métodos e protocolos de funcionamento da rede. Diversas distribuições podem ser assumidas e serão usadas ao longo deste trabalho, desde o modelo aleatório, onde a probabilidade de existir um nó num determinado ponto é a mesma para qualquer outro ponto, até um modelo determinístico, onde a localização dos nós é previamente definida.

Como distância entre nós é uma dimensão extremamente importante nesse cenário, elegemos o modelo colméia para usar em parte das simulações e análises. No modelo colméia, muito utilizado para cobertura de regiões através

de redes sem fio, cada nó tem os seus seis vizinhos mais próximos eqüidistantes. Esse modelo tem o objetivo de simular uma distribuição determinística de nós pela área monitorada, de forma a cobrir toda a área de forma homogênea.

A vizinhança de cada nó, entretanto, não é conhecida previamente. Os nós só descobrem quais são seus vizinhos após a inicialização da rede. Assim, é mais fácil distribuir os nós pela rede, evitando que cada nó tenha que assumir uma posição exata na rede.

A Figura 2.2 ilustra a distribuição de nós no modelo colméia. Nesse modelo, o número de vizinhos de cada nó é fixo e depende apenas do alcance do rádio r e da distância entre dois nós próximos. A Tabela 2.4 mostra o número de vizinhos de acordo com a relação entre o alcance e a distância de dois nós próximos. Os valores são mostrados até o alcance igual a 4 vezes a distância entre dois nós próximos. Nesse caso, o número de vizinhos de cada nó é 54. Nesta tese não trataremos casos com números de vizinhos acima deste valor.

Tabela 2.4 - Número de vizinhos em função do alcance no modelo colméia

Alcance r	Número de vizinhos
$r < d$	0
$d \leq r < 1.73\ d$	6
$1.73d \leq r < 2d$	12
$2d \leq r < 2.78d$	18
$2.78d \leq r < 3d$	30
$3d \leq r < 3.46$	36
$3.46d \leq r < 3.61d$	42
$3.61d \leq r < 4d$	54

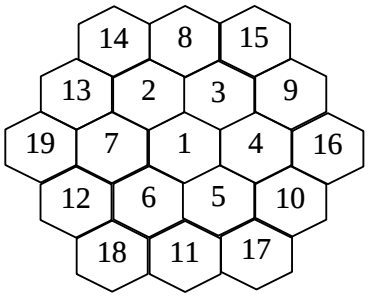

Figura 2.2 - Modelo Colméia

O modelo colméia representa uma distribuição geométrica precisa, que dificilmente será encontrada na prática. Porém, a maioria das distribuições de nós tem como objetivo distribuir os nós de forma homogênea, aproximando-se ao máximo do modelo colméia. O estudo desse modelo, em comparação com o modelo aleatório, pode indicar as vantagens de se realizar um maior esforço para homogeneizar a distribuição dos nós pela rede.

Como nem sempre é possível obter uma distribuição tão determinística, alguns experimentos desta tese são feitos também com outros modelos de distribuição de nós, como o modelo aleatório e modelos intermediários. Eles serão apresentados oportunamente.

2.7.2 Modelo de energia

Manter o baixo consumo de energia é fundamental em qualquer solução para redes de sensores. Este trabalho terá esse objetivo primordial em todas as suas partes. A maioria das decisões foi tomada com o objetivo de reduzir ou manter o consumo de energia. A descrição completa do consumo de energia com processamento, sensoriamento, transmissão e recepção será aqui apresentada.

Considera-se o Mica2 Motes, fabricado pela Crossbow [35], com os seguintes consumos:

- Rádio: 12 mA no modo de transmissão com alcance médio de 40 metros, 8 mA no modo de recepção, e 2 µA no modo *sleep;*
- Processamento: 6 mA no modo normal e 8 µA no modo *sleep*;
- Sensoriamento: 5 mA no modo normal e 5 µA no modo *sleep*.

Como todo nó deve escutar as mensagens enviadas pelos seus vizinhos, o tempo necessário para escutar as mensagens depende do número de vizinhos, do tamanho das mensagens, da largura de banda utilizada e do protocolo de controle do acesso ao meio. Comparando o tempo de recepção com o tempo de transmissão, o primeiro é, pelo menos, tantas vezes maior que o segundo quanto são seus vizinhos. Logo, é inviável utilizar um grande número de vizinhos, pois o maior gasto de energia estaria na recepção. Segundo o modelo colméia, apresentado na seção anterior, nosso modelo terá apenas 6 vizinhos eqüidistantes de cada nó, para que o consumo de energia de recepção não se torne inviável.

Para garantir a longevidade da rede sem a substituição da bateria, os nós devem entrar no modo *sleep* constantemente, permanecendo até a próxima fase de sensoriamento. Isso deve ser bem sincronizado e as mensagens podem ser usadas para manter o sincronismo. No modelo adotado, os nós vão permanecer 99,5% do tempo no modo *sleep*, ficando ativo apenas 0,5% do tempo.

Algumas considerações adicionais foram feitas para se calcular o tempo em que o nó deve permanecer ativo, necessário para que o nó realize todas as suas tarefas em cada ciclo de sensoriamento, antes de voltar a dormir. Considera-se: a largura de banda de 38400 bps; o pacote com 17 bytes; e o preâmbulo de 16 bytes. E ainda, cada nó deve permanecer ativo o tempo necessário para o envio de 10 pacotes, que devem incluir um pacote enviado por cada nó na vizinhança, além do tempo necessário para a negociação do acesso ao meio. Assim, é necessário o funcionamento do nó por 68,8 ms. Para que esse tempo corresponda a 0,5 % do tempo de funcionamento da rede, é necessário um intervalo de 13,68 s em modo *sleep*, o que corresponderá a um ciclo de sensoriamento a cada 13,75 s.

Embora seja uma aproximação grosseira, esse tempo atende à necessidade deste trabalho, que é propor um modelo que permita o uso de uma bateria por um tempo da ordem de um ano.

Considerando ainda que o rádio permaneça quatro vezes mais tempo no modo de recepção que no modo de transmissão, pois deve escutar pacotes de todos os seus vizinhos, além de participar do protocolo de acesso ao meio, assume-se que o rádio permanecerá 0,4% do tempo no modo de recepção e 0,1% do tempo no modo de transmissão, além dos 99,5% no modo *sleep*. Assim, o processador e o sensor permanecerão no modo ativo durante 0,5% do tempo.

A Tabela 2.5 foi extraída de [36]. Criada pelo fabricante, esse documento permite calcular a energia gasta por um nó sensor e o tempo de vida esperado para uma bateria a partir de alguns parâmetros. Os parâmetros descritos acima resultam nos dados da Tabela 2.5.

Tabela 2.5 - Energia gasta por hora de funcionamento de um nó sensor

Energia gasta por hora em mAh	
Processador	0,038
Rádio	0,046
Memória Flash	0,002
Sensor	0,030
Total	0,116

A Tabela 2.6, extraída do mesmo documento, mostra o tempo de vida de várias baterias, em função de sua capacidade. Foi considerado, ainda, um decréscimo natural de 3% ao ano na carga da bateria.

Tabela 2.6 - Tempo de vida de um nó versus capacidade de sua bateria

Capacidade da Bateria (mAh)	Tempo de vida (meses)
250	2,9
500	5,8
1000	11,5
1500	16,9
2000	22,2
3000	32,3

A partir desses dados, é possível perceber que uma rede de sensores sem fio pode permanecer operacional por meses, sem troca ou recarga da bateria, se os requisitos de intervalo de sensoriamento e número de vizinhos forem devidamente dimensionados para esse propósito, assim como é feito no modelo desta tese.

2.7.3 Notação

Os seguintes símbolos serão usados ao longo do texto:

- Id: identificador único do nó sensor, correspondente ao endereço da camada de acesso ao meio;
- K_{MA}: chave mestra do nó A, usada para gerar as demais chaves;
- K_G: chave global, conhecida por todos os nós antes de seu lançamento;
- K_A: chave de difusão, usada pelo nó A para enviar mensagens em difusão;
- K_{AB}: chave par-a-par, usada nas comunicações par-a-par entre os nós A e B;
- K_{An}: n-ésima chave de uma cadeia de chaves, usada nas mensagens enviadas em difusão a partir de A;
- $\{m\}_K$: cifragem da mensagem m com a chave K;
- HMAC(K, m): resultado da computação da função resumo HMAC, *Hashed Message Autenticated Code*, ou resumo criptográfico da mensagem, aplicada à mensagem m com a chave K;

- $A \rightarrow B$: envio de uma mensagem a partir do nó A para o nó B diretamente, ou seja, em salto único;
- $A \Rightarrow *$: envio de uma mensagem em difusão a partir do nó A;
- K_{In}: n-ésima chave da cadeia de chave usada para inserção de novos nós;
- EB $\Rightarrow\Rightarrow$ * : envio de mensagem em difusão multipassos a partir da estação base;
- EB $\Rightarrow\Rightarrow * : \{m\}_*$: envio de mensagem em difusão multipassos a partir da estação base encriptada par-a-par;

Capítulo 3

Trabalhos Relacionados

Os trabalhos apresentados na literatura relacionados à segurança em RSSF procuram adaptar soluções existentes para redes convencionais e redes *ad hoc* para as restrições de RSSF ou propor novas soluções específicas para RSSF.

Alguns trabalhos adicionam alguns requisitos de segurança como confidencialidade e integridade, pela criptografia. Nesse sentido, vários trabalhos relacionados à criptografia e ao gerenciamento de chaves foram propostos para RSSF. Esses trabalhos serão referenciados mais especificamente no decurso do texto.

O aumento da disponibilidade da rede é o foco de vários outros trabalhos. Com esse objetivo, alguns trabalhos propõem mecanismos de segurança a serem aplicados na função de roteamento para se proteger de ataques do tipo negação de serviço. Mecanismos de detecção e isolamento de intrusos também estão presentes na literatura.

Outros trabalhos ainda avaliam os protocolos e algoritmos de segurança tradicionais no contexto de RSSF.

Embora todos esses trabalhos tenham como foco a segurança em redes de sensores e redes *ad hoc*, nenhum deles é capaz de propor mecanismos de segurança eficazes contra alguns ataques de negação de serviço no roteamento, como *Wormhole* e *Hello Flood*. Os mecanismos mais utilizados, como o controle de acesso na camada de enlace, apresentam vulnerabilidades na distribuição de chaves que podem comprometer a solução em alguns ataques. Outros ainda, como mecanismos de detecção de intrusos propostos só funcionam bem quando apenas um nós executa o ataque.

3.1 Criptografia

As restrições de RSSF limitam as soluções criptográficas a serem usadas nesse tipo de ambiente. O custo de processamento dos algoritmos de chave pública limita seu uso. Além disso, o gerenciamento de chaves não deve usar um

número muito alto de mensagens de rede, caso em que o custo do processo pode inviabilizar seu uso.

Guajardo *et al.* apresentam um estudo sobre a implementação de um algoritmo de chave pública para um microcontrolador com poder de processamento similar à algumas implementações de nós sensores [25]. Mallan *et al.* implementaram o mesmo algoritmo para o Tiny OS e avaliaram seu uso no nó sensor Mica 2 [37]. Embora ambos os trabalhos mostrem que seja possível utilizar esses métodos, seu uso ainda demanda algumas dezenas de segundos de processamento dedicado e conseqüente consumo elevado de energia.

A capacidade de processamento dos nós é explorada em [26]. Os algoritmos de criptografia de chave pública e os protocolos de distribuição de chaves são testados visando a verificar sobrecarga de processamento. Os processadores podem também variar seu consumo em função da velocidade de processamento. Os artigos [38] e [39] apresentam testes de microcontroladores executando algoritmos de criptografia de chave pública com diferentes freqüências e resultados mostrando diferentes consumos. Porém, ambos estão acima do esperado para RSSF.

3.1.1 Tiny Sec

TinySec [6] é uma proposta para uma arquitetura de segurança para a camada de enlace do TinyOS, para garantir confidencialidade, controle de acesso, integridade. Os requisitos de segurança providos pelo TinySec são: controle de acesso, integridade, confidencialidade, frescor e segurança semântica.

O TinySec não define a forma como as chaves são distribuídas. Uma abordagem de distribuição de chaves deve ser utilizada em conjunto, que pode ser, desde o uso de uma chave global, até o uso de chaves individuais por nós.

O algoritmo de criptografia usado é conhecido como Skipjack, escolhido por apresentar performance melhor que os algoritmos Triple DES e AES, ser público e não necessitar de nenhum tipo de pré-processamento de chaves. O mesmo algoritmo é usado no modo CBC (*Cipher Block Chainnng*) para encriptação e CBC-MAC (*Cipher Block Chainnng – Message Authenticated Code*) para assinatura.

O TinySec apresenta duas possibilidades de encriptação e assinatura para os pacotes enviados pela rede: apenas autenticados, em opção conhecida como TinySec-Auth, ou autenticados e encriptados, em opção conhecida como TinySex-AE. A Tabela 3.1 apresenta o formato da mensagem para o Tiny OS, TinySec-Auth e TinySec-AE. A sobrecarga na comunicação, principal responsável pelo consumo de energia, é pequena.

Tabela 3.1 - Formato de quadro do TinySec

Field	*Tiny OS*	*TinySec-Auth*	*Tiny-Sec-AE*
CRC/MAC	2	4	4
Destination ID	2	2	2
Source ID	-	-	2
Ctr	-	-	2
Data length	1	1	1
Group	1	-	-
Active message handler	1	1	1
Data	0..29	0..29	0..29
Total	7..36	8..37	12..41

Os números da tabela mostram que o aumento relativo ao TinySec é de apenas um byte na opção TinySec-Auth e 5 bytes na opção TinySec-AE. Este último tem maior sobrecarga devido à composição do vetor de inicialização (IV), formado pelo cabeçalho que deve incluir também o identificador do nó origem (Source ID) e campo de controle (Ctr).

O trabalho de estabelecimento de chaves apresentado no Capítulo 6 pode ser usada em conjunto com o TinySec, como opção para as chaves. O trabalho LEAP [7], apresentado em detalhes mais à frente, também pode ser usado para essa finalidade.

3.2 Gerenciamento de chaves

Nesta seção serão apresentados com mais detalhes alguns trabalhos diretamente relacionados ao nosso, mostrando soluções para distribuição de chaves. O protocolo LEAP [7], em detalhes na seção 3.2.1, incorpora mecanismos para estabelecimento de quatro tipos de chaves, para as diversas possibilidades de comunicação entre os nós.

Vários outros trabalhos têm foco na distribuição de chaves. Alguns, como será visto na subseção 3.2.2, utilizam pré-distribuição probabilística de chaves. Outros, como mostrado na seção 3.2.3, chaves previamente estabelecidas entre

os nós e a estação base são usadas para a encriptação e assinatura das mensagens fim-a-fim.

A autenticidade em comunicação em modo de difusão a partir da estação base é utilizada para garantir que nenhum nó intruso se faça passar pela estação base. O primeiro trabalho com este objetivo é o μTesla, apresentado junto ao trabalho SPINS [4], com detalhes na seção 3.2.3.

O trabalho apresentado em [40] apresenta uma abordagem para gerenciamento de chaves de forma descentralizada, utilizando uma RSSF hierárquica. Os nós mestres são responsáveis pelo gerenciamento local de chaves. Na ausência desses nós, é usada uma chave por grupo de nós. Essa abordagem desconsidera a estação base no gerenciamento de chaves.

3.2.1 LEAP

Zhu *et al.* apresentaram o protocolo LEAP para estabelecimento de chaves em RSSF [7]. O protocolo LEAP estabelece quatro tipos de chaves para cada rede: chave individual (*individual key*), compartilhada entre cada nó e a estação base, usada para comunicação entre a estação base e o nó e vice-versa; chave par-a-par (*pairwise key*), compartilhada entre um nó e um de seus vizinhos, usada para comunicação nó-a-nó; chave de difusão (*cluster key*), compartilhada entre um nó e todos os seus vizinhos, usada para comunicação em difusão local; e chave global (*global key*), compartilhada por todos os nós da rede, para difusão global multipassos pela estação base para toda a rede. Além destas, para difusão local, junto com a chave de difusão, é necessária uma chave auxiliar, obtida de uma cadeia de via única.

As chaves individuais são geradas e pré-carregadas nos nós antes do seu lançamento. As chaves par-a-par são derivadas de uma chave inicial globalmente compartilhada K_I, também pré-carregada nos nós antes do seu lançamento e dos identificadores dos nós, a partir de uma fase de descoberta da vizinhança. Um nó envia uma mensagem em difusão local para se anunciar. Cada nó que recebe essa mensagem responde ao emissor, notificando seu identificador. Assim, os pares de nós vizinhos são capazes de gerar a chave par-a-par a ser usada entre eles a partir da chave K_I e dos identificadores dos nós.

As chaves difusão são geradas pelos próprios nós e entregues para cada um dos seus vizinhos, encriptadas pelas respectivas chaves par-a-par, já então estabelecidas. Essas chaves são usadas para comunicação em difusão local, ou seja, no envio de mensagens de um nó para todos os seus vizinhos. Cada nó deve ter sua própria chave de difusão, que será conhecida por todos os seus vizinhos. Para evitar que um nó tente se passar por outro usando chaves difusão por ele conhecidas, é usada também uma outra chave, pertencente a uma cadeia de chaves de via única. A cadeia é conhecida apenas pelo nó transmissor. A primeira chave da cadeia é enviada a todos os vizinhos junto à chave de difusão. A cada mensagem enviada, uma nova chave da cadeia é divulgada.

Finalmente, as chaves globais são geradas pela estação base e distribuídas para todos os nós legítimos, usando uma árvore de roteamento e o protocolo μTesla, proposto por [4].

No LEAP, todas as chaves, exceto as chaves individuais, são derivadas de uma chave inicial K_I. Para garantir a segurança de todo o protocolo, a chave K_I é apagada em todos os nós após a geração das chaves par-a-par, limitada a um tempo T_{est}. Um pressuposto crítico no LEAP é que $T_{est} < T_{min}$, onde T_{min} é o tempo mínimo necessário para um inimigo efetuar um ataque. Como T_{est} é normalmente pequeno, esse requisito parece razoável. Porém, durante o lançamento dos nós, alguns nós podem simplesmente não se iniciar, devido a problemas de hardware, preservando a chave global K_I e comprometendo o esquema. Se esses nós forem adulterados e suas chaves descobertas, o atacante terá acesso à chave K_I, e assim poderá obter todas as chaves par-a-par da rede.

Outro problema do LEAP é o número excessivo de mensagens para o estabelecimento das chaves, pois cada nó pode ter que enviar um número mensagens igual a duas vezes o número de vizinhos, sendo uma mensagem para configuração da chave par-a-par e outra para distribuição da chave de difusão. Em redes densas, com média acima de 20 vizinhos, o custo de estabelecimento de chaves pode ser consideravelmente alto.

3.2.2 Pré-distribuições de chaves probabilísticas

Esquemas probabilísticos de estabelecimento de chaves têm sido amplamente apresentados na literatura. Eschenauer and Glicor [29]

apresentaram o primeiro trabalho, no qual um conjunto global de chaves secretas é uniformemente pré-distribuído pela rede de forma que cada nó tenha um subconjunto secreto e quaisquer dois nós vizinhos tenham uma probabilidade de compartilhar uma chave em comum nos seus subconjuntos de chaves. Nesse esquema, os nós podem compartilhar uma chave diretamente ou estabelecer uma chave indiretamente com a ajuda de nós intermediários em um caminho com vários passos. Esses esquemas são vulneráveis ao comprometimento de um nó, o que faz com que as chaves nele armazenadas possam ser usadas pelo inimigo para descobrir chaves compartilhadas por outros pares de nós. Além disso, o número de chaves mantido por cada nó é limitado pelas restrições de memória desses elementos, o que vai implicar em uma baixa probabilidade que dois nós consigam estabelecer uma chave diretamente.

Chan *et al.* [28] propuseram uma variação do esquema de pré-distribuição probabilística, chamado *q-composite*. Nesse esquema, são necessárias *q* chaves compartilhadas por dois nós para gerar uma chave que possa ser usada por um par de nós vizinhos, onde *q* é um número inteiro maior que 1. Essa proposta tem o objetivo de reduzir o impacto do comprometimento de um nó. Porém, aumenta as exigências do número de chaves para que dois nós consigam estabelecer uma chave entre eles. Esse aumento tem efeito direto na necessidade de memória e redução da conectividade da rede.

Um problema adicional desses esquemas probabilísticos é que a chave usada pelos nós não pode garantir autenticação forte, porque, para tanto, a chave utilizada não deveria ser conhecida por nenhum outro nó. Para resolver esse problema, Chan *et al.* [28] propuseram uma associação entre os nós da rede para garantir que cada par de nós utilize uma chave diferente, garantindo seu uso par-a-par.

Esquemas determinísticos, como o NEKAP, desenvolvido como parte deste trabalho, são preferidos em relação aos probabilísticos, por garantirem a conectividade total da rede pelo estabelecimento de chaves entre todos os nós que precisam se comunicar diretamente. A vantagem dos esquemas

probabilísticos apresentados na literatura é que normalmente não fazem uso de pressupostos iniciais vulneráveis, como uma chave globalmente conhecida.

3.2.3 SPINS

Perrig *et al.* [4] propuseram uma solução com os principais requisitos de segurança necessários em rede de sensores, em dois blocos: SNEP e μTESLA. O primeiro garante confidencialidade, integridade, autenticação e frescor na comunicação entre a estação base e um nó sensor. E o segundo tem como propósito promover autenticação em modo difusão.

SNEP está apoiado na existência de uma chave mestra compartilhada entre cada nó sensor e a estação base. É uma abordagem com o uso de encriptação e assinatura fim-a-fim com o algoritmo RC5. Permite também o estabelecimento de uma chave exclusiva para comunicação entre dois nós através da estação base, que funciona como KDC.

Por ser uma abordagem fim-a-fim, os requisitos de segurança atendidos não protegem contra ataques nos protocolos de camadas inferiores. A verificação de assinatura e a decriptação somente são realizadas no destino. Desta forma, uma mensagem falsa, alterada ou repetida só é inutilizada nos extremos da comunicação. Isto pode ser explorado para exaurir a energia da rede com inserção de mensagens inválidas. Além disso, a aplicação de encriptação fim-a-fim impede o uso de abordagens como fusão e agregação de dados, que são usadas para economizar energia, ao levar a uma redução do número de mensagens de rede.

μTESLA é uma abordagem para a difusão de mensagens, como as mensagens de atualização de rotas que são enviadas pela estação base e são críticas para que o roteamento seja realizado de forma segura. O funcionamento do μTESLA é baseado no período de tempo de validade de uma chave. Após este período, a chave expira, sendo então divulgada. Para garantir autenticidade dessa chave, ela deve fazer parte de uma cadeia de chaves, onde a chave n-1 é gerada a partir da chave n, em uma função não reversível. Assim, ao receber a chave n, o nó verifica se ela gera a chave n-1 através da função utilizada. Desta forma, é confirmada a autenticidade. Para tanto, é necessário um sincronismo de tempo entre os nós. O sincronismo não é rigoroso perante

um erro, desde que seja menor que o intervalo de tempo utilizado como validade da chave.

As principais desvantagens do µTESLA são a dificuldade de obtenção de sincronismo de relógio, o atraso inserido pela espera da chave e as sobrecargas com divulgação da chave e renovação da cadeia de chaves.

Nenhuma dessas abordagens, SNEP ou µTESLA, impede ataques do tipo DoS na camada de rede. Um nó inimigo pode facilmente se infiltrar no roteamento e executar diversos ataques. Além disto, não é garantida a segurança no envio de mensagens em modo de difusão a partir do nó, o que torna impeditivo o uso de alguns algoritmos de roteamento propostos.

3.2.4 Criptografia de chave pública

Diversos trabalhos recentes mostraram ser viável o uso de algoritmos de criptografia de chave pública para determinadas situações.

Watro *et al.* [41] propuseram o *TinyPK*, que utiliza o RSA, algoritmo de criptografia de chave pública na comunicação entre nós e dispositivos externos. As operações mais caras são realizadas nos dispositivos externos e apenas as operações mais eficientes são realizadas nos nós.

Oliveira *et al.* [42] propõem TinyPBC, um esquema de distribuição de chaves não interativas com autenticação baseada na identidade através de emparelhamento de nós. Esse esquema utiliza criptografia de chave pública para estabelecer chaves par a par entre quaisquer dois nós vizinhos. A chave pública pode ser confirmada pelo identificador do nó, o que evita a necessidade de certificados. Além disso, uma implementação eficiente é apresentada, capaz de realizar o processo em cerca de cinco segundos. Como o mesmo é realizado apenas na inicialização da rede, o custo pode ser suportado, embora seja significativo, pois ele deve ser necessário para o estabelecimento de cada chave par-a-par. No nosso modelo, isso significaria trinta segundos de execução contínua de processamento ativo para estabelecer as chaves com os seis vizinhos.

3.3 Rotas seguras

O processamento interno nas RSSF para agregação e fusão de dados requer que os nós intermediários tenham acesso aos dados, o que dificulta o projeto dos protocolos de roteamento. O problema de roteamento seguro em RSSF foi discutido primeiramente por Karlof e Wagner [5]. Esse trabalho sumariza os ataques contra os protocolos de roteamento existentes e indica medidas preventivas e considerações de projeto para protocolos de roteamento seguros.

Ataques do tipo *Wormhole* estão entre os mais danosos em RSSF. Estudo e propostas sobre prevenção e detecção de ataques do tipo *Wormhole* em redes *ad hoc* já foram apresentadas em [43] e [44]. O princípio utilizado é o de verificar o tempo e a distância percorridos por um pacote e descartar os pacotes que percorrem grandes distâncias em um tempo mais curto que o devido. Essa abordagem não é viável para todas as RSSF, pois ela necessita de sincronismo de tempo e informações de localização que nem sempre podem ser obtidos no hardware simples dos nós sensores. Normalmente não é disponível um relógio de precisão confiável.

O sexto capítulo desta tese apresenta um esquema de alternância no roteamento, visando a aumentar a resiliência da rede à presença de intrusos e, ainda, permitindo a detecção eficiente de intrusos presentes no roteamento.

Os trabalhos conhecidos como INSENS [23], que será visto com detalhes a seguir, ARRIVE [45] e outro em [46] propõem o uso de rotas múltiplas no algoritmo de roteamento para agregar robustez. O uso de rotas múltiplas de forma redundante tolera a presença de intrusos e falhas nos canais de comunicação, porém há um aumento considerável no consumo de energia.

O trabalho apresentado em detalhes na subseção 3.4.1, Localização e Isolamento de Intrusos [47], apresenta uma abordagem que visa aumentar a disponibilidade da rede, detectando e eliminando a presença dos intrusos, que podem ser nós adulterados.

Finalmente ainda serão apresentados nesta seção dois trabalhos de detecção de intrusos em RSSF que não alteram o funcionamento da rede, indicando a presença de intrusos a partir da observação do funcionamento da rede.

3.3.1 INSENS

Deng *et al.* [23] propuseram um protocolo uso de rotas múltiplas para tolerar a presença de intrusos. Sob a alegação que é impossível detectar intrusos, este protocolo se mantém funcional mesmo com a presença de um número moderado de intrusos. Para que isto seja possível, algumas condições devem ser atendidas nas redes de sensores sem fio:

- Os nós não enviam em modo difusão e somente enviam para a estação base;
- Todas as informações de roteamento devem ser autenticadas;
- A estação base deve centralizar todo o processo de roteamento, definindo quais serão os nós roteadores e as rotas;
- Todos os nós têm rotas redundantes para chegar à estação base;
- A presença de vários nós silenciosos em uma rota indicam a presença de intrusos nessa rota.

Além disto, todos os nós devem ter armazenadas as rotas e identificação dos nós sensores que utilizam esse nó como roteador, não permitindo a passagem de qualquer pacote que não esteja nessas rotas. Esta abordagem garante que não exista a inserção de mensagens por parte do inimigo e, junto com as demais ações, permite a presença do intruso sem causar prejuízos maiores à comunicação. Porém, o custo em termos de energia é muito alto. A centralização do algoritmo de roteamento e a manutenção de rotas redundantes consumem recursos energéticos excessivos da rede.

O algoritmo de roteamento apresentado utiliza a estação base para gerar todas as rotas e enviá-las aos nós sensores, de forma criptografada com uma chave por nó sensor, compartilhada com a estação base.

Esta abordagem tem vários problemas que não foram ainda bem explorados. Entre estes problemas estão o custo energético, a vulnerabilidade a alguns ataques, a escalabilidade da rede e o envio de mensagens em modo de difusão.

INSENS tem alto custo energético. O processamento na estação base aumenta o custo de comunicação, uma vez que as tabelas de rotas têm de ser enviadas para os diversos nós que participam do roteamento. Além disto, os

pacotes são enviados de forma redundante, o que leva a um consumo excessivo de energia.

INSENS está sujeito ainda a ataques. O ataque conhecido como S*ybil* [3], que consiste na simulação de várias identidades por um único nó intruso, pode ser efetuado contra uma RSSF utilizando rotas redundantes com sucesso. Para tanto, basta um invasor presente na rede assumir múltiplas identidades, se fazendo passar por vários nós diferentes, assumindo então presença em todas as rotas redundantes.

O uso desta abordagem compromete ainda a escalabilidade da rede. Cada nó deve armazenar rotas para qualquer nó que o utilize como roteador. Ora, se cada informação de rota ocupa alguns bytes, a memória necessária para todas as rotas pode crescer muito, ultrapassando a disponibilidade de memória dos nós sensores.

O envio de mensagens em modo de difusão pelos nós também não é permitido nessa abordagem. Essa limitação pode impedir o funcionamento de algumas aplicações que utilizam comunicação em modo de difusão de um nó com seus vizinhos.

3.4 Detecção de intrusos

3.4.1 Localização e isolamento de intrusos

Staddon *et al.* [47] propuseram um protocolo para localizar nós com falhas e alterar as rotas de forma a suprir a falta dos nós. A falha pode representar algum defeito ou uma invasão que impede a função de roteamento no nó. O isolamento da falha agrega robustez à rede, protegendo contra ataques de negação de serviço. Para tanto, a estação base precisa conhecer a topologia de toda a rede. Cada nó deve informar para a estação base quem são seus vizinhos. Esta será então responsável por comunicar as rotas a todos os nós. Quando determinada região da rede estiver em silêncio, a estação base pode solicitar aos vizinhos dos nós silenciosos que tentem localizá-los para restabelecer assim a árvore de roteamento.

O trabalho de Staddon *et al.* faz *a*lgumas considerações que nem sempre são aplicáveis à RSSF: os nós contam com proteção física contra adulteração e a

estação base é capaz de transmitir diretamente para todos os nós da rede. A partir dessas considerações é proposta uma chave compartilhada entre todos os nós para garantir controle de acesso e integridade. Os dispositivos nós sensores a que se propõe o trabalho desta tese, não contam, porém com mecanismos eficientes contra a adulteração. Assim, o inimigo poderia facilmente obter a chave global a partir de adulteração de um nó da rede e efetuar diversos ataques como inserção de mensagens incorretas.

A técnica de Staddon *et al.* tem vantagens, principalmente o comprometimento com a manutenção do baixo consumo de energia, mas ainda não apresenta boas técnicas para a detecção do intruso. Entretanto, tem problemas: o intruso que executar ataques como Encaminhamento Seletivo pode enganar facilmente o mecanismo de detecção, uma vez que esse ataque permite a passagem de algumas mensagens.

3.4.2 Detecção de intrusos de forma distribuída

Silva *et al.* propõem mecanismos de detecção de intrusos de forma descentralizada, pelos quais alguns nós executam funções de monitores e detectam a presença de intrusos [8]. Apresentam também o esboço de uma metodologia para construção de IDS's específicos de uma RSSF alvo, com aplicações bem definidas, bem como o desenvolvimento de um simulador simplificado capaz de simular as principais características das RSSF e do IDS proposto.

A solução foi projetada de forma a permitir que o IDS pudesse ser adaptado a uma variedade de RSSF e suas diferentes aplicações. A idéia geral é, a partir do conhecimento das características da RSSF específica, definir regras possíveis e, a partir dos dados disponíveis na rede, escolher as regras que podem ser implantadas com maior ou menor custo.

O algoritmo utilizado tem três fases para a detecção dos intrusos: aquisição de dados, aplicação de regras e detecção de indícios. Para validar essa solução, foram realizadas várias simulações com redes planas estacionárias com 100 nós, sendo apenas um deles realizando ataque.

Um problema dessa abordagem é a possibilidade de um nó intruso se passar por monitor, permitindo a um inimigo gerar falsos positivos para acarretar a revogação de nós autênticos.

3.4.3 Detecção de intrusos de forma centralizada

Teixeira [9] propõe mecanismos de detecção de intrusos de forma centralizada, com base nas informações disponíveis na estação base, sem alteração na rede. O modelo de informação para detecção de intrusão é condizente com as RSSFs e a arquitetura é extensível, podendo ser adaptada para diferentes modelos de comportamento e diferentes estratégias de detecção. O sistema de detecção de intrusos apresentado é adaptável a RSSFs novas ou já projetadas, não requer a alteração do software ou hardware dos nós sensores; por conseqüência, não disputa os recursos dos nós sensores com as aplicações e protocolos utilizados na rede.

Esse sistema de detecção de intrusos funciona por observação, obedecendo a uma série de restrições especialmente ligadas à manutenção das características das RSSF. O modelo de informação utilizado foi inspirado na visão de mapas, proposto por Ruiz *et al*. [10]. São criados mapas de produção, roteamento e mapa de estado operacional.

Os mapas são combinados para indicar se o comportamento observado difere ou não do comportamento esperado. Os dados são analisados em redes bayesianas. Foi construído um protótipo e testado em simulações. O percentual de falsos negativos foi razoável, chegando a até 50% em alguns casos.

Os testes foram realizados com apenas um intruso na rede. A presença de mais intrusos não foi analisada, ao contrário do trabalho apresentado no sexto capítulo desta tese, onde os intrusos chegam a representar 30% da rede.

3.5 Gerenciamento de segurança

Não existem muitos trabalhos sobre gerenciamento de RSSF. Savola e Uusitalo [34] apresentam princípios de gerenciamento de segurança para redes ad hoc, que apresentam desafios diferentes para segurança em relação às RSSF. O principal problema em redes ad hoc é a falta de uma administração central

confiável. RSSF não têm esse problema, mas possuem mais restrições de energia e hardware.

Dimitriou e Krontiris [48] apresentam uma visão geral dos atuais desafios em segurança para RSSF, destacando seus aspectos de comunicação autonômica. Eles apresentam os atuais debates nas pesquisas sobre segurança em RSSF: protocolos de estabelecimento e configuração inicial confiável de chaves, tolerância a ataques de negação de serviço, tolerância a nós alterados, roteamento seguro, segurança sensível a localização, fusão de dados segura e técnicas eficientes de criptografia. Na discussão, mostram como a comunicação autonômica oferece oportunidades para aumentar a segurança em RSSF. O trabalho resume estas características autonômicas: auto-configuração, auto-reconhecimento, auto-adaptação, auto-organização e auto-otimização, bem como discute o que é necessário para prover uma solução integrada e completa para segurança em RSSF.

Alguns modelos de gerenciamento de redes foram propostos na literatura. Na arquitetura Manna [49] políticas descrevem o comportamento desejado dos componentes de gerenciamento, como agentes e gerentes. Manna será a base deste trabalho no que diz respeito a gerenciamento. Seu modelo é adotado e as definições aqui inseridas são uma extensão do Manna.

Song *et al.* [50] propõem um elemento que age como um mediador entre redes Universal Plug and Play (UPnP) e RSSF. Esse elemento, chamado BOSS (Bridge of the sensors), é um agente UPnP implementado na estação base e posicionado entre os controladores UPnP e os nós sensores não-UPnP a serem gerenciados. O objetivo é realizar a implantação de serviços em RSSF sem prover configuração para a rede.

Zheng *et al.* [51] propõem uma abordagem de gerenciamento de segurança baseada em políticas para redes *ad hoc*. Os elementos do gerenciamento de políticas PDP (Policy Decision Point), PEP (Policy Enforcement Point) e LPDP (Local Policy Decision Point) são distribuídos pela rede de forma a permitir a ação de soluções de segurança multicamadas, permitindo uma ação mais eficaz. Entretanto, a arquitetura das redes de sensores é muito distinta da

arquitetura das redes *ad hoc* convencionais, razão pela qual essa abordagem não poderá ser usada diretamente em redes de sensores.

Tanto quanto podemos saber, não existe na literatura uma abordagem de provimento de segurança que seja dinâmica e considere os objetivos ou restrições de tempo de vida e consumo de energia das redes de sensores.

3.6 Conclusões

Os diversos trabalhos presentes na literatura visam garantir requisitos de segurança às RSSF. Requisitos como confidencialidade, integridade e autenticidade par-a-par e fim-a-fim são obtidos pelos processos criptográficos de chave simétrica disponíveis. Para tanto as propostas de distribuição de chaves representam as maiores contribuições.

A garantia dos requisitos fim-a-fim, entre um nó e a estação base, são facilmente resolvidos pelo compartilhamento de uma chave única entre a estação base e cada nó, como apresentado por Perrig *et al*. [4]. Porém, essa abordagem não pode ser usada para evitar ataques efetuados nos serviços presentes nas camadas inferiores à aplicação, como é o caso da função de roteamento.

A confidencialidade e autenticidade par-a-par, por sua vez, podem ser usadas para controlar o acesso ao protocolo de roteamento, eliminando, assim, a presença do inimigo e restringindo o efeito dos ataques de negação de serviço. O problema, nesse caso, é estabelecer uma chave entre quaisquer pares de nós. As soluções presentes na literatura ou limitam a conectividade da rede, o que ocasiona o aumento dos caminhos e, conseqüentemente, do consumo de energia, ou usam pressupostos nem sempre válidos, como a segurança de uma chave global, a proteção contra *Tampering*, ou a disponibilidade de algoritmos de chave pública.

Esta tese apresenta, no Capítulo 6, um protocolo de estabelecimento de chaves para criptografia par-a-par, que não apresenta as mesmas limitações dos protocolos então existentes. Esse protocolo, chamado NEKAP, vai cobrir as lacunas deixadas pelos protocolos existentes, com pressuposições iniciais mais condizentes com as RSSF.

Outra grande frente de trabalho de pesquisa está no desenvolvimento de protocolos de roteamento que sejam menos vulneráveis aos ataques de negação de serviço. Abordagens como INSENS aumentam a disponibilidade da rede tolerando a presença do intruso. Porém, seu custo energético é muito alto em relação às soluções de roteamento já apresentadas.

Existem outros trabalhos de detecção de intrusos na literatura, que consideram sempre a presença de um único intruso. Entretanto, os ataques tendem a ser em larga escala, pois um inimigo pode ter tantos recursos quando o proprietário da rede. A presença de vários intrusos pode dificultar a eficácia das abordagens existentes, inviabilizando seu uso pelo excesso de falsos positivos e falsos negativos.

Assim, nenhum dos mecanismos propostos é eficaz para inibir ataques de negação de serviço na função de roteamento, como *Wormhole*, *Hello Flood* ou *Black Hole*. Os mecanismos até então propostos para tanto, apresentam vulnerabilidades que podem comprometê-los seriamente, ou então não são viáveis devidos a restrições de energia e processamento no modelo apresentado. Assim, esta tese visa apresentar mecanismos mais eficientes e menos vulneráveis para garantir a disponibilidade dos dados coletados nas redes de sensores pela proteção da função de roteamento contra ataques de negação de serviço.

Esta tese apresenta, no Capítulo 7, uma proposta de modificação do algoritmo de roteamento das RSSF incluindo alternância no roteamento, visando aumentar a resiliência da rede à presença de intrusos. Essa proposta possibilita, ainda, o uso de um algoritmo de detecção de intrusos atuando de forma centralizada na estação base. O algoritmo também é apresentado. Essa abordagem é uma solução eficiente contra os ataques de negação de serviço conhecidos, pois pode revelar os nós causadores de ataques que suprimem o serviço de roteamento, permitindo sua revogação.

Esta tese apresenta, também uma proposta de arquitetura de gerenciamento de segurança para RSSF, encapsulando os diversos mecanismos de segurança em componentes, que podem ser configurados de acordo com a demanda. A presença de intrusos na rede, detectados pelos devidos mecanismos

pode gerar a necessidade de inclusão de outros componentes, de forma automática, para garantir o correto funcionamento da rede. Essa arquitetura será apresentada no próximo capítulo e seus efeitos serão avaliados ao final desta tese.

Capítulo 4

Um Estudo do Impacto dos Ataques de Negação de Serviço em RSSF

4.1 Introdução

Agentes externos podem interferir no funcionamento de uma rede tornando-a total ou parcialmente indisponível. Esses agentes podem ser maliciosos, considerados como intrusos, ou causados por elementos não conscientes da presença da rede, como agentes naturais, chuvas, ventos ou terremotos, ou ainda ação humana que desconheça a presença da rede. A ação de inimigos se destaca por ter o objetivo de deixar inoperante a rede toda ou parte dela, nos ataques conhecidos como negação de serviço, ou DoS, do inglês *Denial of Service*.

Devido à diversidade de ataques de negação de serviço, seus efeitos na rede variam muito. Da mesma forma, alguns mecanismos de segurança podem ser muito eficazes contra um tipo de ataque, mas sem efeito protetor para outros tipos de ataque.

Este capítulo apresenta um estudo dos ataques de negação de serviço no roteamento, indicando o alcance do ataque, executado de forma isolada ou distribuída, os mecanismos de proteção conhecidos contra o ataque e o custo de execução do ataque. São analisados os ataques conhecidos com Buraco Negro (*Black Hole*), Encaminhamento Seletivo (*Selective Forward*), *Wormhole*, *Sinkhole* e *Hello Flood*. Embora outros ataques do tipo DoS contra o roteamento sejam conhecidos, eles não são expressivos e são pouco citados na literatura.

O objetivo deste estudo é mostrar que tipo de ataque pode ser mais danoso para rede, indicando também estratégias para que a rede possa se defender. Alguns ataques efetuados por alguns poucos nós podem silenciar boa parte da rede. Outros ataques devem ser executados em massa para obter o efeito esperado. Este capítulo apresentará a abrangência de cada ataque e o

número de elementos intrusos necessários para a realização de cada ataque de forma a prejudicar a maior parte da rede.

Como o objetivo é verificar a extensão do ataque, as métricas utilizadas vão indicar qual o percentual da rede será afetado em função do número de nós intrusos. Em alguns casos, quando o ataque tiver abrangência mais restrita, será necessário um grande número de nós intrusos para silenciar uma parte significativa da rede. Nesse caso, como no ataque *Black Hole*, os experimentos foram realizados considerando 10 ou 30 % dos nós da rede efetuando o ataque. Em outros ataques, como o ataque *Wormhole*, a presença de uns poucos nós intrusos pode silenciar a maior parte da rede. Nesses ataques, foi considerada, então, a presença de poucos nós intrusos.

4.2 Ataques de negação de serviço em RSSF

Ataques do tipo Negação de Serviço (*DoS - Denial of Service*) são ataques contra a disponibilidade da rede. Seu objetivo é tornar a rede, ou parte dela, indisponível durante um determinado período de tempo ou indefinidamente. Esses ataques bloqueiam o serviço da rede ou de parte dela, impedindo sua operação normal.

RSSF são muito vulneráveis a ataques de DoS. Devido à diversidade de ataques do tipo negação de serviço, esses ataques são muito mais difíceis de serem defendidos do que ataques passivos ou dos demais ataques ativos. Os protocolos e serviços oferecidos em redes de sensores devem ser projetados considerando a segurança para evitar os ataques DoS. Do contrário, redes de sensores sem fio irão permanecer muito vulneráveis a ataques do tipo negação de serviço [3].

Ataques do tipo DoS podem ser classificados de acordo com a camada da pilha de protocolos de rede onde atuam. Assim, os seguintes ataques DoS foram catalogados em RSSF [3] [5]:

- Camada física:
    - Interferência (*Jamming*): geração de sinal de rádio que causa interferência eletromagnética na mesma freqüência de transmissão dos sensores;

- Adulteração (*Tampering*): adulteração física do sensor para retirar ou substituir informações, possivelmente substituindo o código do nós, alterando seu funcionamento;

- Camada de enlace:
  - Colisão (*Collision*): envio de pacotes falsos simultaneamente a outros autênticos, dentro do alcance do transmissor, com objetivo de forçar uma colisão e conseqüente reenvio, com gasto adicional de energia;
  - Exaustão (*Exhaustion*): qualquer tentativa que visa exaurir os recursos da rede, especialmente fonte de energia. Por exemplo, colisões sucessivas, causando retransmissão e conseqüente exaustão da bateria;
- Camada de Rede:
  - Buraco Negro (*Black Hole*): supressão total do serviço de roteamento num nó inimigo inserido em uma rota;
  - Encaminhamento seletivo (*Selective Forward*): supressão parcial do serviço de encaminhamento de pacotes, simulando falhas intermitentes e dificultando a identificação do ataque;
  - *Worm Hole*: tunelamento de mensagens recebidas em parte da rede para outra parte, gerando problemas de roteamento;
  - *Hello Flood*: amplificação, por um inimigo, da potência do sinal de um nó invasor no processo de estabelecimento de rota. Todos os nós passar a identificar o nó cujo sinal foi amplificado como vizinho;
  - *Misdirection*: envio de pacotes por nós invasores por caminhos errados, podendo exaurir a energia da rede e impedir o encaminhamento das informações para a estação base;
  - *Synkhole:* um nó intruso inicia o algoritmo de roteamento, enviando um novo *beacon* para a rede, se fazendo passar pela estação base. Caso não esteja preparada para se defender desse ataque, a rede pode se reconfigurar para rotear todos os pacotes para o nós intruso.

O presente trabalho está particularmente interessado nos ataques de negação de serviço que podem interferir na função de roteamento das mensagens. Entre os ataques estudados neste trabalho está o ataque adulteração (*Tampering*) e os ataques de negação de serviço no roteamento Buraco Negro, Encaminhamento Seletivo , *Wormhole, Hello Flood e Synkhole*. O ataque adulteração é importante para o estudo desse trabalho, embora não seja um ataque função de roteamento, porque pode permitir a inserção de invasores internos, que detém informações privilegiadas da rede, como chaves e conhecimento dos protocolos, e dificultam a defesa contra os ataques no roteamento.

Este capítulo vai apresentar uma avaliação desses ataques baseado no alcance que eles podem ter. Em alguns casos, como o ataque adulteração, somente será apresentado sua forma de realização e proteção. Em outros, como ataque de Buraco Negro, por ser um ataque muito fácil de ser realizado, o interesse deste trabalho é apresentar as conseqüências de um ataque realizado de forma distribuída por vários nós intrusos. Assim, serão considerados 10 ou 30% dos nós como intrusos. Em outros ataques, como *Wormhole* e *Hello Flood*, como são ataques mais elaborados e com resultados mais abrangentes, serão considerados apenas poucas instâncias desses ataques na rede.

As avaliações foram feitas através de simulações, usando o modelo de rede colméia. O simulador utilizado foi aquele apresentado por Martins *et al*. em [52] que inclui implementação de diversos tipos de ataque e é mais adequado a este trabalho.

4.2.1 Adulteração *(Tampering)*

Um nó sensor pode ser fisicamente adulterado pelo inimigo, num ataque conhecido como *Tampering*. Esse ataque consiste na captura física de um nó, leitura de suas informações e possível substituição do seu código por um código malicioso. Técnicas de leitura de memórias possibilitam a obtenção de todas as informações presentes no nó [52][53] e assim o inimigo terá posse de todas as informações do nó, incluindo as chaves.

O nó adulterado pode então ter seu código substituído e ser inserido novamente na rede. Como o nó pertence à rede, o ataque é considerado um

ataque interno. O inimigo pode ainda repassar as informações obtidas no nó para outros nós intrusos de forma a efetuar o ataque em vários pontos da rede.

Os intrusos adulterados re-inseridos pelo inimigo têm a finalidade de efetuar outros ataques, como inserção de mensagens incorretas, alteração de mensagens que circulam pela rede e ataques de negação de serviço, como Buraco Negro e Encaminhamento Seletivo.

Proteção

Várias proteções físicas podem ser utilizadas no sentido de impedir a adulteração de um nó. Nenhuma delas, porém, consegue ser imune a todos os tipos de mecanismos de adulteração presentes na literatura [1] quando o nó é acessível fisicamente ao inimigo. Além disso, as proteções físicas contra adulteração encarecem os circuitos eletrônicos e conflitam com os requisitos das redes de sensores sem fio.

Avaliação

Considerando o ataque de adulteração como um ataque direto contra a disponibilidade, seu alcance é bem limitado, uma vez que apenas um nó é afetado por ataque. Entretanto, este ataque é normalmente usado para dar acesso a um nó da rede, possibilitando que outros nós ataques possam ser disparados, uma vez que o nó adulterado é considerado pela rede como um nó interno, conhecendo suas chaves e protocolos.

4.2.2 Buraco Negro

O ataque conhecido como Buraco Negro é realizado por um nó intruso que se introduz na árvore de roteamento da rede, mas não realiza as funções de repasse de pacotes. Assim, todo pacote que chega até esse nó para ser repassado é descartado. O ataque só tem efeito se o nó que realiza o Buraco Negro está presente na árvore de roteamento. E o efeito é tanto maior quanto mais nós dependem desse nó para o roteamento. Para reforçar o efeito desse ataque, ele pode ser realizado em conjunto com outro ataque, como o *Wormhole* ou *Hello Flood*, com o objetivo de aumentar o número de nós que dependem daquela rota.

O ataque Buraco Negro é realizado pela inserção de um nó intruso na rede, o qual vai participar do protocolo de roteamento, e fazer com que as rotas se estabeleçam através desse nó. No protocolo de roteamento *Tiny OS beaconing*, a inserção do nó intruso no roteamento se dá simplesmente pelo repasse do *beacon*. Todos os nós que escutam o *beacon* pela primeira vez, com origem no nó intruso, o colocam como pai na árvore de roteamento.

Proteção

O ataque Buraco negro exige a participação do nó intruso no roteamento para que ele tenha efeito. Considerando o algoritmo de roteamento *Tiny OS beaconing*, a entrada do Buraco Negro depende do repasse do *beacon* pelos nós intrusos antes dos demais. A proteção contra esse ataque deve consistir no impedimento do repasse desse *beacon* por nós externos à rede.

A autenticação do *beacon* de atualização de rotas pela estação base não impede o ataque Buraco Negro, pois o nó intruso apenas repassa o *beacon* para entrar no algoritmo de roteamento, sem alterar as informações contidas, mantendo suas informações de autenticação.

O estabelecimento do Buraco Negro pode ser evitado pela autenticação salto a salto, no enlace, a cada salto de repasse de informações. Assim, o nó intruso, sem conhecimento das chaves necessárias, não seria capaz de autenticar o *beacon* no repasse para os seus vizinhos, e teria seu pacote descartado. Nesse caso, cada nó só deve comunicar-se com os nós na sua vizinhança para os quais tenha uma chave segura compartilhada. Este trabalho de doutoramento apresenta uma solução baseada nesse princípio, no Capítulo 6.

A proteção contra o Buraco Negro também pode ser realizada através da detecção e revogação de intrusos. A detecção pode ser feita de forma centralizada na estação base [9] ou distribuída pelos nós[8]. O Capítulo 7 desta tese também apresenta uma abordagem que permite a detecção de intrusos realizada de forma centralizada. A vantagem da detecção centralizada sobre a detecção distribuída é que o comando de revogação deve partir da estação base e ser autenticado e enviado para todos os nós da rede. A revogação de nós não pode ser solicitada por nós da rede; do contrário, os nós intrusos poderiam

utilizar desse mecanismo para desativar nós autênticos, o que poderia causar outro ataque do tipo negação de serviço.

O uso de vários caminhos no roteamento, como apresentado no Capítulo 7, aumenta a resiliência da rede à presença do Buraco Negro, pois permite que os pacotes cheguem até a estação base por outros caminhos, que não aquele onde está presente o Buraco Negro.

Avaliação

Ao ser submetida a um ataque Buraco Negro, a rede se torna parcialmente sem produção. Os nós silenciados são aqueles que dependem do nó que executa o ataque para entregar seus pacotes para a estação base.

A existência de apenas um nó intruso na rede vai acarretar a interrupção da produção de todos os nós presentes na subárvore de roteamento que se origina no nó intruso e vai ao sentido das folhas. Se o nó intruso é um nó folha, o ataque não terá qualquer efeito. O número de nós silenciados, nesse ataque de Buraco Negro, corresponde ao número de nós que dependem do nó intruso no roteamento.

Para aumentar a abrangência do ataque, um inimigo pode utilizar vários nós intrusos, ao invés de apenas um. Nesse caso, o prejuízo causado é muito maior e pode ser estimado através de simulação. Várias simulações foram realizadas neste trabalho para indicar o número de nós silenciados em casos que 10% e 30% dos nós da rede são intrusos. Os resultados são apresentados na Tabela 4.1 e na Figura 4.1.

Tabela 4.1 - Impacto do ataque Buraco Negro

Número de nós	Nós silenciados com 10 % de intrusos	Nós silenciados com 30 % de intrusos
50	21%	49%
100	24%	63%
200	41%	79%
400	50%	87%
600	51%	89%
800	58%	91%
1000	74%	97%

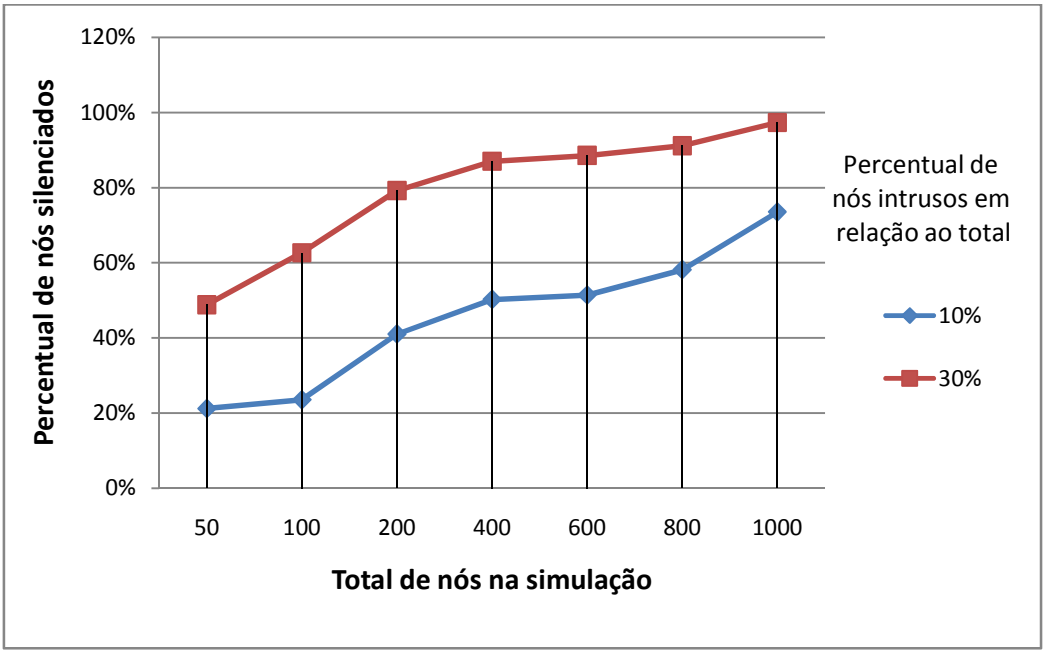

Figura 4.1 - Percentual de nós silenciados em função do total de nós na simulação

Os resultados indicam que, com um número representativo de nós intrusos, 10%, é possível silenciar boa parte dos nós autênticos, cerca de 30%. Aumentando-se o número de intrusos para cerca de 30%, a abrangência do ataque é bem maior, chegando a 80% da rede. Na média, é possível afirmar que para cada intruso presente na rede, três nós autênticos são silenciados. E que um inimigo precisa de apenas um terço do esforço realizado pelo proprietário da rede para conseguir inutilizá-la através desse ataque.

Para as redes com maior número de nós, o número médio de saltos necessários para a entrega de pacotes aumenta, aumentando, conseqüentemente, a probabilidade de existência de um intruso nesse caminho e silenciando um maior número de nós, como pode ser visto no gráfico da Figura 4.1.

4.2.3 Encaminhamento Seletivo

O ataque conhecido como Encaminhamento Seletivo (*Selective Forward)* é muito similar ao Buraco Negro. A diferença é que o ataque Buraco Negro descarta todas as mensagens que deveriam ser repassadas pelo algoritmo de roteamento. Já o ataque Encaminhamento Seletivo repassa algumas

mensagens. O objetivo é confundir os mecanismos de detecção de intrusos e falhas, os quais podem identificar mais facilmente o ataque Buraco Negro.

A decisão sobre qual mensagem deve ser repassada e qual mensagem deve ser descartada pode ser baseada em algumas informações como origem da mensagem, conteúdo da mensagem, número de seqüência, ou ainda, descartar de forma aleatória.

Proteção

A proteção contra o ataque de Encaminhamento Seletivo é realizada da mesma forma que para o ataque Buraco Negro. A diferença se dá apenas na detecção de intrusos. Métodos muito simples podem confundir esse ataque com uma falha intermitente.

Avaliação

Para fins de avaliação por parte desse trabalho, foi considerado o repasse de forma aleatória, onde 80% das mensagens serão descartadas, simulando, assim, uma falha intermitente.

A avaliação do Encaminhamento Seletivo é semelhante ao Buraco Negro, uma vez que esses ataques são muito parecidos. A diferença é que, com a ocorrência do Encaminhamento Seletivo, ao invés de nós silenciados, alguns nós têm resposta reduzida. Dada a tamanha semelhança, os testes executados tiveram resultados idênticos, sem a necessidade de reproduzir seus resultados aqui.

4.2.4 Sinkhole

No ataque conhecido como *Sinkhole*, um nó intruso se faz passar pela estação base e dá início ao processo de estabelecimento de rotas. Ele envia um *beacon* informando ser a estação base. Como a mensagem é repassada a todos os nós da rede, novas rotas são estabelecidas, tendo como destino final o nó que se proclamou estação base. O ataque *Sinkhole* é bastante abrangente e apenas um nó intruso pode desviar toda a produção da rede. Contudo, ele pode ser facilmente eliminado usando apenas a autenticação no *beacon* de atualização de rota, garantindo sua origem na estação base.

Proteção

O protocolo μTesla, proposto por Perrig *et al.* [4], pode ser usado para autenticar o *beacon* enviado pela estação base, evitando que outro nó se passe por ela.

Além de ser facilmente evitado, o Sinkhole também é facilmente detectado, pois o falso *beacon* também alcança a estação base, revelando a existência do *Sinkhole*.

Avaliação

A execução do ataque *Sinkhole* é bastante simples, uma vez que é necessário apenas um nó executando o ataque *Sinkhole* para que toda a rede deixe de enviar seus pacotes para a estação base, inutilizando, assim, toda a rede. A avaliação de seu uso indica que apenas um intruso silencia toda a rede.

4.2.5 Wormhole

O ataque *Wormhole* consiste em criar um túnel entre dois pontos da rede. O inimigo repassa mensagens de uma parte da rede para outra parte em um canal de baixa latência. Uma versão simples deste ataque é o posicionamento de um nó inimigo entre dois nós da rede. O nó inimigo repassa todas as mensagens de um para outro nó. A forma mais comum, porém, é executada por dois nós distantes repassando mensagens entre si, utilizando um canal exclusivo do inimigo de latência menor [5].

Os pacotes repassados podem manter as características originais do ponto onde foi recebido, como endereço de origem e destino, ou então ser repassado como seria por qualquer outro nó, utilizando os endereços de origem e destino dos nós intrusos, que repassaram a mensagem. O uso de uma ou outra abordagem pode ter um efeito diferente sobre a rede e exigir técnicas diferentes para proteção e detecção.

O ataque de *Wormhole* pode ser usado em conjunto com outros ataques como Buraco Negro ou Encaminhamento Seletivo, para descartar os pacotes durante seu roteamento, reduzindo, assim, a produção da rede. Como um dos nós responsável pelo *Wormhole* deve estar mais próximo da estação base, o outro nó terá uma posição muito privilegiada no algoritmo de roteamento,

fazendo que todos os nós no outro extremo do túnel utilizem o mesmo como parte da árvore de roteamento. Desta forma, o roteamento fica sujeito ao túnel que pode suprimir mensagens e até alterá-las, caso não seja utilizado nenhum tipo de proteção.

O ataque *Wormhole* pode ser usado também para aumentar o caminho do roteamento, repassando os pacotes do nó mais próximo da estação base para aquele mais distante, o que poderia exaurir a energia da rede pelo aumento do caminho dos pacotes até a estação base. Esse cenário, no entanto, não será o objetivo deste trabalho, pois a associação com o Buraco Negro e conseqüente redução da produção ser mais prejudicial à rede.

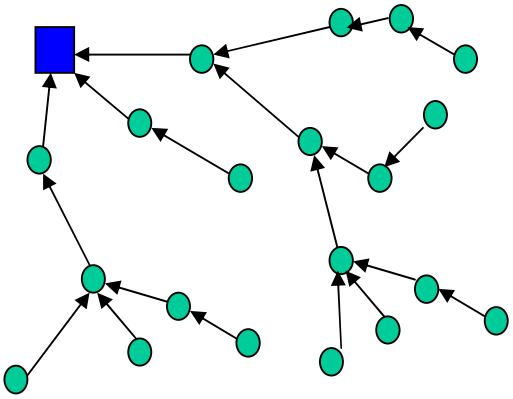

Figura 4.2 - Árvore de roteamento sen a presença de *Wormhole*

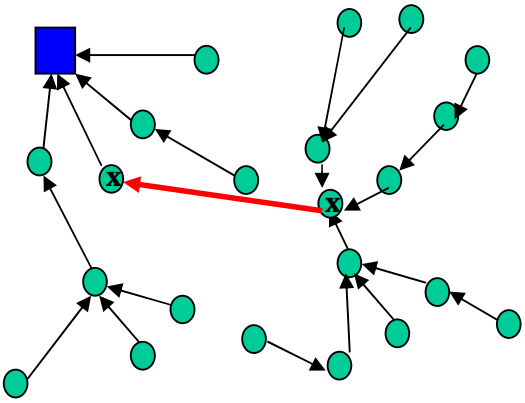

Figura 4.3 - Árvore de roteamento com a presença de *Wormhole*

O principal objetivo do *Wormhole* é ampliar o efeito de outros ataques, como escuta, Buraco Negro, Encaminhamento Seletivo e outros. Como o canal estabelecido pelo ataque consegue privilegiar sua participação no roteamento, esses ataques têm um efeito maior caso estejam associados com o *Wormhole*.

As figuras 4.2 e 4.3 apresentam uma RSSF que utiliza o algoritmo de roteamento *Tiny OS beaconing*. A introdução de dois invasores na figura 4, marcados com “x” e um canal de baixa latência entre eles, deixa o nós mais distante numa posição privilegiada no algoritmo de roteamento, fazendo que as mensagens de muitos nós passem por ele. Essas mensagens podem ser alvo de escuta, adulteração ou negação de serviço.

A execução do ataque *Wormhole* exige do inimigo a inserção de dois nós e a criação de um canal de comunicação de baixa latência entre eles. Não é um ataque tão fácil de ser realizado como os ataques Buraco Negro,

Encaminhamento Seletivo ou *Sinkhole*. O canal de baixa latência pode ser um canal sem fio de alcance maior, ou um canal através de fios de cobre ou fibra ótica.

Proteção

O ataque *Wormhole* é muito difícil de ser evitado e combatido. Ele pode permanecer durante muito tempo na rede sem apresentar efeitos negativos, apenas repassando todas as mensagens recebidas. Abordagens presentes na literatura [44] para combater ataques do tipo *Wormhole* utilizam de informações geográficas, como localização, e temporais, nem sempre disponíveis nas RSSF mais simples, uma vez que recursos como GPS podem encarecer muito o hardware dos nós.

O uso de controle de acesso à rede pode evitar o estabelecimento do *Wormhole*, mas sua aplicação não é trivial. Para que seja evitado, é necessário que os nós consigam reconhecer os vizinhos, impedindo a entrada de um novo vizinho, mesmo que autêntico. Isso porque o *Wormhole* pode repassar pacotes de outro nó, autêntico, já reconhecido pela rede, sem alterar suas características. Caso o controle de acesso da rede permita o acesso de um nó autêntico a qualquer ponto da rede, o *Wormhole* pode ser estabelecer mesmo com o controle de acesso. Em redes móveis com alta mobilidade de nós, não é possível restringir o acesso do nó apenas à sua vizinhança para evitar o *Wormhole*.

A solução de distribuição de chaves apresentada no Capítulo 6 possibilita o controle de acesso na camada de enlace, o que restringe a entrada do ataque *Wormhole* na rede. Embora essa solução tenha sido elaborada para redes estáticas, ela permite uma relativa mobilidade dos nós, desde que o nó percorra as sucessivas vizinhanças para se locomover, restabelecendo chaves através de um vizinho em comum com cada vizinhança. Assim, a possibilidade de ataque *Wormhole* fica muito restrita, pois um intruso teria que percorrer o caminho entre os dois nós, simulando uma movimentação do nó, até atingir o ponto onde será estabelecida a outra ponta do *Wormhole*.

Quando o *Wormhole* é usado para ler o conteúdo das mensagens, seu efeito pode ser eliminado por encriptação das mensagens. Ao ser usado para

promover a negação de serviço, uma vez formado, o efeito do *Wormhole* só pode ser eliminado pela presença de um mecanismo de detecção e revogação de intrusos.

Quando combinado com outro ataque, sua detecção pode ser realizada pelas ações tomadas em função do outro ataque, como Buraco Negro.

Avaliação

A avaliação do ataque *Wormhole* realizada neste trabalho considerou sua existência em conjunto com o Buraco Negro. Assim, todas as mensagens enviadas pelo W*ormhole* são descartadas. Como sua abrangência do ataque é muito grande e apenas um par de nós realizando o *Wormhole* pode afetar a produção de boa parte da rede, esse ataque será avaliado para uma menor quantidade de nós, considerando apenas algumas unidades.

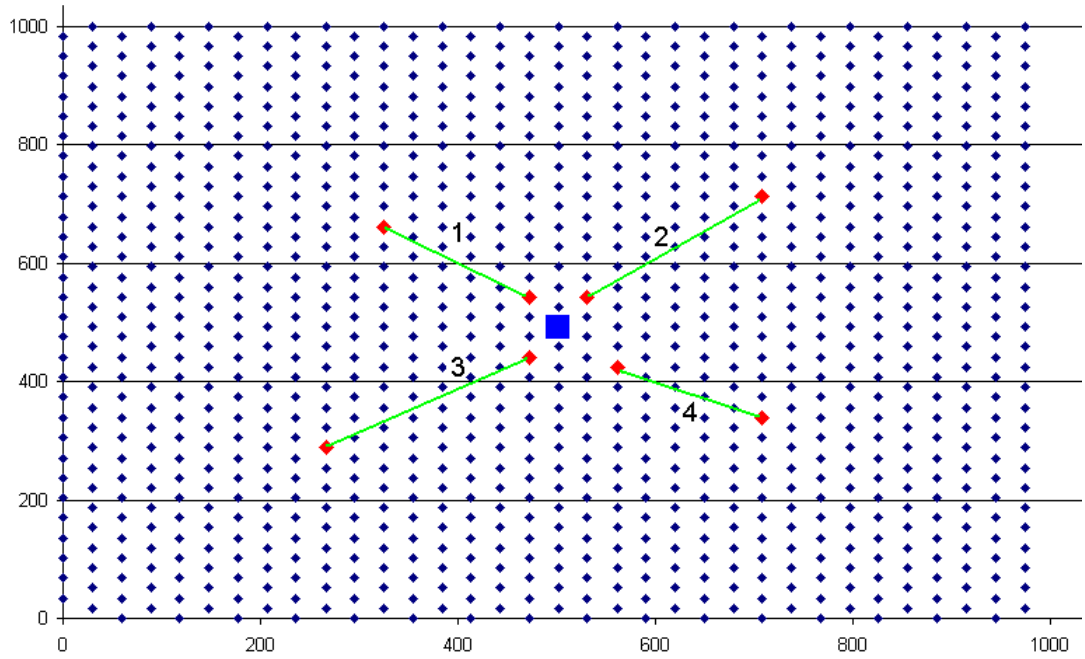

Figura 4.4 - Ataques *Wormhole* simulados

A Figura 4.4 representa os ataques *Wormhole* simulados. Aqui, quatro nós vizinhos da estação base foram escolhidos para iniciar o túnel que tem como fim um outro nó, distante dois saltos da estação base. Assim, esses nós conseguem uma posição privilegiada no algoritmo de roteamento, pois

conseguem repassar o *beacon* antes dos demais nós próximos, se estabelecendo em uma posição próxima à raiz na árvore de roteamento.

Tabela 4.2 - Número de nós silenciados pelo ataque *Wormhole*

Número de Nós	Número de pares de nós intrusos	Nós silenciados
1025	1	231
1025	2	447
1025	3	599
1025	4	828

O ataque Buraco Negro foi realizado em conjunto com o ataque Wormhole, de forma que esse último aumente o alcance do Buraco Negro. Assim, além de se estabelecer no roteamento, os nós intrusos não repassaram as mensagens para a estação base, reduzindo a produção da rede, que foi verificada no final da simulação.

Os números obtidos, apresentados na Tabela 4.2, mostram que alguns poucos nós realizando ataques *Wormhole* com Buraco Negro podem deixar grande parte da rede em silêncio. Apenas um *Wormhole* pode inutilizar 20% da rede. Com um número maior de intrusos, apenas os nós próximos à estação base ficam imunes ao ataque.

A associação entre os dois ataques, *Wormhole* e Buraco Negro, permite a um intruso assumir uma posição privilegiada no roteamento, fazendo que um número maior de nós dependa dela para repassar seus dados para a estação base. Assim, a supressão do serviço de repasse atinge um maior número de nós e tem um impacto maior na produção da rede.

4.2.6 Hello Flood

O ataque conhecido como *Hello Flood* consiste no aumento do alcance de transmissão de um nó muito próximo da estação base, no momento do estabelecimento de rotas. O nó intruso, ao receber o *beacon* de estabelecimento de rotas, o envia em um sinal de longo alcance. Grande parte dos nós da rede escuta o *beacon* e adiciona o nó intruso como pai na árvore de roteamento.

Durante a execução normal da rede, ao produzir um dado e enviar para a estação base através do seu pai, as informações não chegam até o pai, que se encontra distante. Assim, os pacotes são perdidos.

A maior vantagem desse método de ataque é que ele pode ser realizado sem a necessidade de inserção de um nó intruso. O invasor pode somente inserir um repetidor de rádio que amplifica o sinal de um nó algumas vezes. Dessa forma, o ataque é bastante simples de ser efetuado.

A autenticação do *beacon* pela estação base também não é suficiente para evitar esse ataque. Assim como no *Wormhole*, é necessário um controle de acesso que restrinja os movimentos dos nós autênticos na rede, para que o *Hello Flood* não seja confundido com a movimentação de um nó. A solução de distribuição de chaves apresentada no Capítulo 6 também é capaz de restringir ataques do tipo *Hello Flood*.

Outra proteção eficaz contra o *Hello Flood* é detecção e revogação de intrusos. Como os pacotes repassados pelo *Hello Flood* são escutados por grande parte da rede e podem ser gerados próximos à estação base, é possível para a estação base identificar a presença desse ataque e avisar à rede para ignorar o nó em ataque. Caso os pacotes do *Hello Flood* não estejam no alcance da estação base, ainda sim é possível detectá-lo através dos nós mais próximos que escutam o envio de pacotes com uma potência acima do normal. O problema, nesse caso, é que os nós não devem ter permissão para causar a revogação de um nó, pois isso poderia ser usado pelo intruso para revogar os nós autênticos.

Avaliação

A abrangência do ataque *Hello Flood* depende muito do alcance do rádio utilizado. É possível dimensionar ataques que alcancem toda a rede, inutilizando todos os nós, exceto aqueles tão próximos à estação base quanto o nó intruso. Também é possível dimensionar ataques onde o alcance do rádio é direcionado de forma a evitar a região onde se encontra a estação base. O objetivo de evitar a estação base é evitar a detecção imediata do intruso.

Alguns testes foram realizados com o objetivo de verificar a abrangência de ataques de *Hello Flood* executados de forma isolada e de forma distribuída.

A distância da estação base também foi um fator considerado nos testes. Em alguns casos, os nós intrusos estão presentes na área de alcance da estação base, de forma a escutar seus pacotes diretamente. Nesse caso, todavia, a estação base também é capaz de escutar os pacotes enviados pelo nó intruso, facilitando sua detecção. Nos testes 5 e 6 da Tabela 4.3, os nós intrusos se encontram próximos à estação base. Nos demais, a distância da estação base é suficiente para que eles não sejam detectados. O alcance dos nós autênticos em todos os ataques é de 80 metros. Os nós estão distribuídos de acordo com a distribuição colméia.

Tabela 4.3 - Resultados das simulações de Hello Flood

Ref.	Número de Nós	Posição dos intrusos em relação à EB	Alcance do nó intruso (m)	Número de nós intrusos	Nós silenciados	Percentual de nós silenciados
1	1025	Distante	160	1	240	23,41%
2	1025	Distante	160	2	443	43,22%
3	1025	Distante	160	3	557	54,34%
4	1025	Distante	160	4	702	68,49%
5	1025	Próximo	160	1	838	81,76%
6	1025	Próximo	400	1	945	92,20%

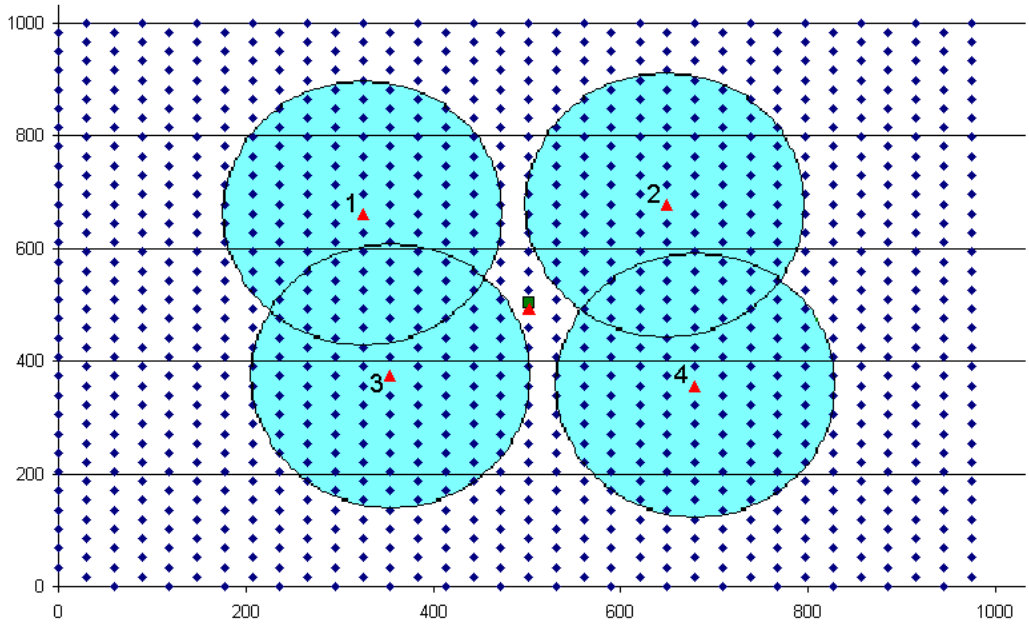

Figura 4.5 - Ataques *Hello Flood* simulados

A Figura 4.5 mostra a posição dos ataques *Hello Flood* na rede simulada. A estação base é representada por um quadrado no centro da rede. Os nós indicados por triângulos e numerados são aqueles usados nos ataques 1 a 5 Tabela 4.3. Os círculos, que tem o centro nesses nós, representam o alcance do seu rádio. Nesses ataques, a estação base não é capaz de escutar os pacotes enviados pelos nós intrusos, dificultando sua detecção. O nó indicado por triângulo no centro da rede é o nó intruso usado nos ataques 5 e 6 da Tabela 4.3.

É possível verificar, diante dos resultados, que o poder de paralisação do ataque *Hello Flood*. Apenas um nó intruso pode paralisar grande parte da rede, acima de 80%, caso esteja próximo da estação base. Certamente, nesse caso, é fácil para a estação base identificar a presença desse intruso e indicar à rede sua revogação, orientando os demais nós para que o ignorem. Caso os nós intrusos estejam distantes da estação base, a ação de cada um pode paralisar a produção de uma parte representativa da rede. Em um ataque distribuído, como na simulação, a abrangência do ataque também pode paralisar a maior parte da rede, chegando também a 80% ou mais, usando apenas quatro nós intrusos devidamente posicionados.

4.3 Sumário de ataques

A partir dos resultados obtidos neste capítulo, é possível sumarizar as informações sobre os ataques. A Tabela 4.4 apresenta uma síntese dos ataques. Estão informados, além do nome do ataque, sua abrangência, a facilidade de detecção direta pela estação base, a vantagem que pode encorajar seu uso e os métodos de defesa.

Tabela 4.4 - Síntese dos ataques

Ataque	Abrangência	Percepção pela estação base	Vantagem	Defesa
Adulteração (Tampering)	Afeta apenas o nó adulterado, mas pode ser usado para outros ataques	Não é possível	Dá acesso a todas as informações sigilosas do nó	Hardware protegido contra *tampering*
Blackhole	Afeta os nós que dependem do intruso para o roteamento.	Apenas pelo mapa de produção, mas pode ser confundido com falha.	Facilidade de execução Dificuldade de detecção	Autenticação salto a salto do beacon Detecção e revogação de Intrusos
Selective Forwarding	Nós que dependem do intruso para o roteamento	Apenas pelo mapa de produção, mas pode ser confundido com falha	Facilidade de execução Dificuldade de detecção	Autenticação salto a salto do beacon Detecção e revogação de Intrusos
Wormhole	Aumenta o efeito do Buraco Negro, aumentando o número de nós cuja rota para a estação base passa pelo intruso.	Apenas pelo mapa de produção, mas pode ser confundido com falha.	Grande abrangência do ataque Dificuldade de detecção	Controle de acesso no enlace, restringindo a mobilidade dos nós Detecção e revogação de Intrusos

Hello Flood	Aumenta o efeito do Buraco Negro, pois o alcance maior atrai mais nós para a rotas onde está presente o intruso. Pode abranger toda a rede.	O ataque pode ser detectado se a estação base escutar o pacote enviado pelo intruso. Os nós vizinhos também podem detectar pelo nível do sinal superior aos demais	Grande abrangência do ataque Facilidade de execução	Controle de acesso no enlace, restringindo a mobilidade dos nós Detecção e revogação de Intrusos
Synkhole	Pode abranger toda a rede	Os beacons do intruso chegam à estação base, denunciando o intruso que pode ser revogado	Grande abrangência do ataque Facilidade de execução	Autenticação do beacon pela estação base Detecção e revogação de Intrusos

4.4 Conclusões

Vários ataques de negação de serviço são apresentados na literatura. Cada ataque apresenta características que podem torná-lo mais fácil de ser executado, mais eficaz, ou ainda mais difícil de ser identificado e combatido.

Os mecanismos mais eficientes de proteção do roteamento são o controle de acesso no enlace e a detecção, seguida da revogação, de intrusos. O controle de aceso no enlace tem por objetivo impedir a participação dos nós intrusos nos algoritmos de roteamento. A detecção e revogação dos intrusos têm por objetivo eliminar os nós que tenham conseguido entrar no roteamento e estejam

prejudicando a produção da rede. Essas soluções podem ser usadas em conjunto, ou alternadas, de acordo com os objetivos e a vulnerabilidade de cada aplicação. Os próximos capítulos apresentam soluções que tornam possíveis a utilização desses mecanismos nas RSSF.

O quinto capítulo apresenta uma arquitetura de gerenciamento de segurança, que possibilita ligar e desligar as soluções de segurança apresentadas, de acordo com a demanda da rede. Essa demanda é registrada pela ocorrência de intrusão, detectada em algum dos mecanismos de detecção de intrusos.

O Capítulo 6 capítulo apresenta um protocolo de estabelecimento de chaves entre nós sensores vizinhos que permite a realização de um controle de acesso efetivo entre a vizinhança dos nós sensores, eliminando a possibilidade de inserção de nós intrusos para a realização de ataques de negação de serviço no roteamento.

O sétimo capítulo apresenta uma abordagem com o uso de rotas alternativas no roteamento para aumentar a resiliência da rede à presença de intrusos e, ainda, permitir a detecção eficiente de intrusos, permitindo, a seguir, sua revogação.

Capítulo 5

Arquitetura de Gerenciamento de Segurança para Redes de Sensores Sem Fio

5.1 Introdução

Diversos trabalhos apresentam propostas de segurança em RSSF para evitar os efeitos da presença de intrusos na rede. Esta tese apresenta dois mecanismos: gerenciamento de chaves para controle de acesso no enlace, no Capítulo 6, e rotas alternativas para aumento da resiliência e detecção de intrusos, no Capítulo 7. E vários outros podem ser usados para garantir requisitos de segurança para as redes de sensores. Os dois mecanismos aqui apresentados preenchem lacunas ainda sem boas soluções.

Cada solução tem seu custo, visto que recursos como processamento e comunicação gastam energia e tempo. As RSSF têm que economizar energia para estender seu tempo de vida, pois as redes consideradas têm grandes restrições no consumo de energia.

Sistemas de gerenciamento de redes podem agir numa rede com o objetivo de aumentar o tempo de vida da mesma. Controle de densidade de nós sensores é um exemplo de serviço de gerenciamento de rede usado com esse objetivo na arquitetura Manna de gerenciamento de RSSF [49].

Um sistema de gerenciamento de segurança pode agir em uma rede da mesma forma, por exemplo, para ativar ou desativar serviços e funções de segurança conforme necessário em resposta a alterações na rede. Mas a rede pode economizar energia quando não há indicação ou suspeita de presença de intrusos. Sistemas de detecção de intrusos podem alertar a rede sobre intrusos e em resposta o sistema de gerenciamento pode ativar ou desativar funções ou serviços de segurança.

Este capítulo propõe uma arquitetura de gerenciamento de segurança para RSSF, incluindo seleção de componentes de segurança, descrição de informação de gerenciamento, descrição de mensagens e definição de eventos de

segurança. De modo autonômico, componentes de segurança podem ser agrupados em níveis, que podem ser alterados em resposta a eventos de detecção de intrusos. O objetivo é estender o tempo de vida da rede pela redução do efeito de ataques e pela economia de energia com a ativação dos serviços de segurança somente quando for necessário.

5.2 Gerenciamento de Redes de Sensores Sem Fio

Em RSSF, o gerenciamento da rede é essencial para garantir o uso racional de todos os recursos. Funções de gerenciamento da rede podem ser configuradas para ligar, desligar ou parametrizar os componentes para conseguir um melhor consumo de energia.

O modelo Manna considera uma arquitetura de gerenciamento tri-dimensional, que consiste de áreas funcionais, níveis de gerenciamento e funcionalidades de RSSF. Essas dimensões são especificadas para o gerenciamento de uma RSSF e são a base para uma lista de funções de gerenciamento.

Em uma aplicação demonstrativa, o protocolo de gerenciamento MannaNMP foi utilizado para controlar a densidade dos nós de RSSFs [55]. Com um serviço de controle de densidade próprio, é possível estender o tempo de vida da rede; setores com concentração excessiva de nós podem ter parte dos seus nós desligados, colocados em estado de espera para serem ligados depois de algum tempo, quando a bateria de um ou mais nós ativos da vizinhança se esgotar. Esses nós reservas substituem os ativos propiciando o aumento do tempo de vida de rede. Outro benefício do serviço de gerenciamento é a possibilidade de redução de tráfego na rede, que tem como conseqüência positiva a diminuição do consumo de energia necessário para repassar mensagens para a estação base e a redução das colisões de pacotes. Estudos sobre a adoção de técnicas de gerenciamento para RSSF têm mostrado outros benefícios, como uma baixa geração de mensagens adicionais de gerenciamento da rede e aumento do tempo de vida [49].

5.3 Arquitetura de gerenciamento

A arquitetura de gerenciamento proposta neste trabalho segue as linhas gerais do modelo Manna, onde várias possibilidades são propostas para organizar o relacionamento entre gerente e agentes. O modelo inclui um protocolo de gerenciamento chamado MannaNMP, que descreve os serviços providos e o formato das mensagens, assim como a base de informações de gerenciamento. Neste artigo, são propostas extensões ao MannaNMP para incluir segurança. A arquitetura de gerenciamento, apresentada a seguir, é composta de uma base de informações de gerenciamento, mensagens trocadas e eventos. O modelo considera que os componentes de segurança descritos acima podem ser parte de situações de gerenciamento. Nesse sentido, a configuração dos componentes de segurança é dinâmica, o que significa que eles podem ser incluídos, excluídos, ativados, e desativados em tempo de operação. Eventos fornecem informações para a rede para tornar possível a configuração e a re-configuração dos componentes de segurança de uma maneira autonômica.

Na arquitetura proposta, a estação base centraliza as informações sobre toda a rede, cabendo a ela a centralização das funções de gerenciamento. A estação base executa, então, o gerente. Os nós, que recebem solicitações e processam as ações de gerenciamento, o fazem através da execução de agentes. Os agentes podem habilitar e desligar diversos componentes de segurança, entre eles os algoritmos criptográficos, sistemas de detecção de intrusos, fusão de dados, alternância no roteamento e revogação de nós. A Figura 5.1 apresenta os elementos da arquitetura de gerenciamento proposta.

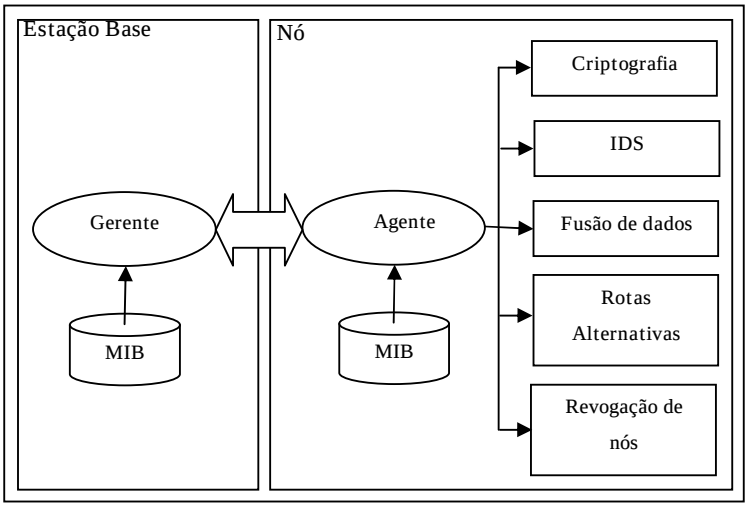

Figura 5.1 - Arquitetura de Gerenciamento de Segurança

5.4 Componentes de segurança

Para este trabalho, foram escolhidos os seguintes componentes de segurança, apresentados na seção 2.6.2:

- Encriptação fim-a-fim;
- Encriptação saltos-salto;
- Assinatura fim-a-fim;
- Assinatura salto-a-salto;
- Detecção de intrusos e revogação de nós;
- Roteamento com alternância de rotas;
- Fusão de dados segura.

Esses componentes garantem uma proteção multicamadas, permitindo a efetiva eliminação dos efeitos da intrusão nos diversos ataques em que ela pode estar presente.

5.5 Decisões autonômicas

RSSF devem ser auto-gerenciáveis e configurar seus componentes para estender seu tempo de vida e assegurar a produção dos dados. Neste trabalho, componentes de segurança são configurados baseados em eventos de segurança gerados por sistemas de detecção de intrusos.

Os eventos de detecção de intrusos configuram os componentes de segurança. Os intrusos detectados pela estação base são revogados usando mensagens autenticadas da estação base. Os intrusos detectados de maneira descentralizada não podem ser revogados pela estação base porque eles não são confiáveis, mas, mesmo assim, um evento de detecção de intruso é gerado, de forma a ativar componentes de segurança.

No trabalho relatado nesta tese, foram definidos níveis de segurança para facilitar decisões autonômicas baseadas em eventos recebidos. Em cada nível de segurança alguns componentes de segurança são ligados para proteger a rede dos intrusos. O nível de segurança da rede aumenta com a evidência de intrusos. O nível de segurança pode ser decrescido, também, devido a níveis baixos de reserva de energia. Para economizar energia, componentes de segurança, como a detecção de intrusos, podem ser desligados.

A Tabela 5.1 exibe eventos e ações autonômicas geradas nesses eventos. Em geral, um intruso é suficiente para alterar o nível de segurança, porque indica que o atual nível de segurança permitiu a entrada de intrusos; todavia, em algumas situações, o nível de segurança pode ser alterado após a detecção de mais de um intruso.

Tabela 5.1- Eventos de detecção de intrusos e ações

Evento	Ação
Estação base detecta um novo intruso	- Intruso é revogado - Nível de segurança aumenta
Um nó sensor detecta um intruso	- Nível de segurança aumenta

A Tabela 5.2 mostra os níveis de segurança. O serviço de detecção de intrusos centralizado, que executa na estação base, está sempre habilitado e não aparece na tabela. Quando a estação base detecta um nó intruso, ele é revogado. No nível Baixo, nenhum componente de segurança é habilitado. A fusão de dados é usada, o que pode reduzir significativamente o consumo da rede.

A detecção de um intruso provoca a mudança de nível. No nível Médio, 10% dos nós da rede atuam como monitores e executam a detecção de intrusos. Esse número foi escolhido baseado no modelo colméia. Como cada nó tem exatamente seis vizinhos, se o número de nós monitores ultrapassasse 16%, a probabilidade de ter um vizinho monitor seria alta para qualquer nó da rede, algo caro e desnecessário para esse nível. Nesse nível, as rotas são autenticadas fim-a-fim usando o µTesla. A criptografia dos dados coletados é feita salto-a-salto e o algoritmo de roteamento utiliza rotas alternativas. As soluções escolhidas no nível Médio são aquelas que consomem menos energia. Para a maioria das invasões e aplicações, esse nível de segurança deve barrar a presença dos intrusos.

No nível Alto, a detecção de intrusos é estendida para 20% dos nós. A probabilidade de existir um nó monitor na vizinhança de qualquer nó é razoável com 20% dos nós intrusos. A grande maioria dos nós terá um monitor em sua vizinhança com 20% dos nós monitores, uma vez que cada nó tem seis vizinhos. Neste nível, a criptografia fim-a-fim desabilita o processamento na rede. Assim,

a fusão de dados não pode ser utilizada. Na presença de intrusos, a fusão de dados não é um método confiável porque um intruso no processo de fusão de dados pode adulterar dados de vários nós. Esse modo somente deve ser usado se nós intrusos ainda são detectados quando a criptografia salto-a-salto está ativa.

Tabela 5.2 - Níveis de segurança autonômicos

Nível	Componentes de segurança utilizados
Baixo	- Sem detecção de intrusos nos nós sensores - Sem utilização de criptografia - Fusão de dados habilitada
Médio	- 10% dos nós executam detecção de intrusos - Atualização de rotas autenticada fim-a-fim - Criptografia salto-a-salto habilitada - Fusão de dados habilitada - Rotas alternativas
Alto	- 20% dos nós executam detecção de intrusos - Criptografia fim-a-fim habilitada - Atualização de rotas autenticada salto-a-salto - Rotas alternativas - Sem fusão de dados
Crítico	- 30% dos nós executam detecção de intrusos - Sem fusão de dados - Criptografia fim-a-fim e salto-a-salto habilitadas - Atualização de rotas autenticada salto-a-salto e fim-a-fim - Rotas alternativas

Se com o nível Alto ativo, intrusos ainda forem detectados, o nível Crítico é iniciado. Nesse nível, todos os componentes de segurança apresentados são utilizados, incluindo criptografia salto-a-salto e fim-a-fim. Nesse nível, considera-se que nós intrusos conhecem algumas chaves da rede. Assim, utiliza-se criptografia redundante, fim-a-fim e salto-a-salto. Dessa forma, um intruso terá de conhecer várias chaves para ter acesso às mensagens da rede.

No nível Crítico 30% dos nós são monitores e executam a detecção de intrusos. Com isso, espera-se dois nós monitores por vizinhança, atuando de forma redundante, para que nenhum intruso deixe de ser detectado. Para que esse nível seja atingido, a rede deve estar sob forte e intenso ataque de intrusos muito bem preparados, com acesso às diversas chaves da rede.

Quando recursos de energia indicarem exaustão da bateria, os nós podem reduzir o nível de segurança para aumentar o tempo de vida. Nesse caso, os componentes de segurança têm um custo de energia maior que a rede pode

gastar. Como os nós estão no fim dos seus tempos de vida, é melhor tentar trabalhar sem segurança do que gastar a energia restante com segurança.

A criptografia pode incluir encriptação e assinatura e os objetivos da rede devem determinar qual técnica tem de ser usada. Se os dados da rede são confidenciais, a encriptação tem de ser usada. De outro lado, somente assinatura pode ser utilizada para evitar adulterações e enganos.

5.5.1 Base de informações de gerenciamento (MIB)

Embora as decisões autonômicas de mudança de nível exijam um conjunto reduzido de mensagens, a MIB foi estendida para configurar e parametrizar todos os componentes de segurança, pois as mensagens podem ser úteis em outros tipos de aplicações a serem desenvolvidas.

Para configurar os componentes de segurança, um número de objetos de gerenciamento foi definido para a MIB. Os objetos são organizados de acordo com o tipo de componente de segurança que os utilizam: criptografia, chaves, dados e administração.

Criptografia

Objetos booleanos indicam se o sistema usa uma função específica de segurança; seus nomes são auto-explicativos: Encriptação fim-a-fim, Encriptação salto-a-salto, Assinatura fim-a-fim, Assinatura salto-a-salto, e Comunicação por difusão.

Gerenciamento de chaves

O próximo conjunto de objetos mantém informações sobre chaves e tem de ser armazenado nos nós sensores para os objetivos da criptografia. Os dados contidos nos objetos não podem circular pela rede por razões de segurança. Cinco objetos são definidos: Chaves de difusão (lista de chaves usada para difusão); Chaves par-a-par (lista de chaves usada para cada vizinho de um nó); Última chave (última chave revelada da cadeia de um nó vizinho); Chave fim-a-fim (chave para ser usada na encriptação fim-a-fim); Chave global (chave para ser usada por todos os nós vizinhos).

Dados

Vários tipos de controle de dados são enviados pela rede pelos nós ou pela estação base. Uma parte deles foi definida pela MIB:

- Nível de segurança (*Choice*) ⇒ Baixo (0), Médio (1), Alto(2), Crítico(3);
- Fusão de dados? (*Boolean*) ⇒ Indica se a fusão de dados é utilizada nos nós;
- Intruso detectado? (*Boolean*) ⇒ Indica se um intruso foi detectado na rede;
- Identificador do intruso (*ID*) ⇒ Identificador do intruso detectado;
- Identificador de nó revogado (*ID*) ⇒ Identifica o nó intruso para ser revogado;
- Lista de nós revogados (*List*) ⇒ Lista de nós suspeitos e revogados;
- Lista de chaves revogadas (*List*) ⇒ Lista de chaves revogadas por um nó.

Administração

A sub-árvore Administração conta com contadores estatísticos de número de pacotes e enviados e recebidos, além do estado adminitrativo do nó narede.

- Estado administrativo (*Choice*) ⇒ Desbloqueado (0), Levemente bloqueado (1), Bloqueado (2);
- Pacotes de gerenciamento enviados (*Integer*) ⇒ mensagens de gerenciamento enviadas pelo nó;
- Pacotes de gerenciamento recebidos (*Integer*) ⇒ mensagens de gerenciamento recebidas pelo nó;

5.5.2 Definição das mensagens

O modelo propõe um gerenciamento de segurança orientado por mensagens, no qual mensagens de controle são usadas para ativar ou desativar componentes, como detecção de intrusos, criptografia, fusão de dados e assinatura digital. Uma mensagem indicando a presença de um intruso colocaria a rede em estado de alerta. Como a identificação precisa do intruso

não é possível, a rede tem de reduzir as possibilidades de comunicação do intruso de forma a anular seus efeitos.

Um número de mensagens de gerenciamento foi definido e é listado a seguir. Em termos do modelo gerente/agente, elas são do tipo *set* e são usadas para estabelecer ou alterar os valores dos objetos, como definido no protocolo MannaMNP.

Mensagens para criptografia

As mensagens são para: ativação de encriptação fim-a-fim; ativação de assinatura fim-a-fim; ativação de assinatura salto-a-salto; utilização de difusão (para um específico período); mudança do protocolo de gerenciamento de chaves.

Mensagens de dados

As mensagens definidas são: *mudança no nível de segurança* (mudança de configuração dos componentes de segurança); *utilização de fusão de dados* (gera mensagem para desativar a encriptação fim-a-fim); *detecção de intruso* (coloca a rede em estado de alerta e envia o identificador do intruso para a estação base); *revogação de nós* (inclui o identificador do nó revogado na lista); *revogação de chave* (inclui a chave revogada na lista de chaves revogadas de nós recebedores).

5.5.3 Eventos

Na ocorrência de eventos, os nós sensores enviam mensagens para informar a estação base. Essas mensagens são usadas pela estação base para alterar a configuração da RSSF, o que pode ser feito imediatamente ou algum tempo depois. Mensagens de *trap* são usadas para informar eventos previamente programados. No caso de redes hierárquicas, as mensagens são encaminhadas para os maiores níveis de hierarquia até alcançar o gerente. Nós intermediários, quando possível, tomam decisões em resposta aos eventos informados, o que torna a rede mais inteligente e pode diminuir o fluxo de mensagens. Para reduzir o consumo de energia, a responsabilidade de

monitoramento de alguns ou todos os eventos é atribuída à estação base ou somente a alguns nós. A comunicação baseia-se no protocolo MannaNMP.

Os eventos definidos são os seguintes: *Detecção de intruso* (nó sensor identificou um nó suspeito); *Revogação de chave* (um nó intruso foi revogado); *Desaparecimento de nó* (nó suspeita que um nó vizinho desapareceu); *Reativação de nó desaparecido* (um nó previamente suspeito de desaparecimento foi identificado e pode ser um intruso); *Nível crítico de energia* (um nível crítico de energia de um nó foi alcançado, as chaves desse nó têm de ser revogadas).

5.6 Contribuições

Dezenas de mecanismos de segurança estão disponíveis na literatura e poderiam ser usados como componentes da arquitetura de gerenciamento de segurança aqui proposta. Alguns deles foram escolhidos para compor este trabalho, levando em consideração o modelo adotado, a eficiência, desempenho e escalabilidade de cada solução.

Dois mecanismos de segurança adicionais são propostos nos capítulos seguintes, para preencher lacunas em aberto na arquitetura de gerenciamento de segurança. Estes mecanismos permitirão reduzir o custo computacional para se obter os requisitos de segurança, ao mesmo tempo em que possibilitam o aumento da proteção desses mecanismos.

No Capítulo 8, uma validação da arquitetura de gerenciamento é realizada, através da avaliação de cada nível e componente de segurança, mostrando sua aplicabilidade nas redes de sensores sem fio especificadas neste trabalho.

Capítulo 6

Estabelecimento de Chaves em RSSF – Protocolo NEKAP

6.1 Introdução

O primeiro ponto a ser abordado na segurança das redes de sensores é a restrição de acessos a nós externos, possivelmente inseridos por um inimigo. Esses nós podem promover ataques do tipo escuta, inserção de mensagens falsas, repetição de mensagens e ataques do tipo negação de serviço no roteamento, uma vez que pode participar dos protocolos de estabelecimento de rotas e executar ataques interferindo no repasse de mensagens.

Este capítulo apresenta um protocolo de estabelecimento de chaves, chamado NEKAP, *Neighborhood-based Key Agreement Protocol*, que estabelece dois tipos de chaves, chaves par-a-par e chaves difusão entre cada nó e seus vizinhos. O principal objetivo dessas chaves é autenticação na camada de enlace, para restringir o acesso ao protocolo de roteamento, eliminado a possibilidade de inserção de intrusos.

O protocolo NEKAP apresenta algumas similaridades com o protocolo LEAP[7], uma solução para distribuição de chaves em RSSF. Todavia, NEKAP é mais resiliente e também mais eficiente no consumo de energia.

O estabelecimento de chaves par-a-par e de difusão entre os nós e seus vizinhos possibilita a autenticação de todos os pacotes enviados diretamente em seu primeiro passo. Assim, todos os pacotes não autenticados podem ser descartados, o que restringe o acesso aos serviços de rede apenas aos nós que participaram do processo de estabelecimento de chaves.

O protocolo de gerenciamento de chaves em RSSF deve atender a alguns requisitos especiais de desempenho e segurança [56]:

- Resiliência contra captura de nós, visto que os nós podem ser facilmente capturados pelo inimigo, que pode ter acesso às suas chaves;

- Resiliência contra replicação de nós, uma vez que os nós capturados podem ser clonados e re-inseridos na rede;
- Suporte a revogação de nós, e conseqüentemente de todas as suas chaves, que não podem ser usadas em outros pontos da rede;
- Escalabilidade, pois o tamanho das RSSF pode variar de algumas unidades até milhares de nós.

No protocolo aqui apresentado, cada nó, carregado inicialmente com uma chave mestra diferente, envia sua chave para os seus vizinhos, em difusão, encriptada com uma chave conhecida globalmente. As chaves de cada nó são geradas a partir da chave mestra, de forma que todos os vizinhos podem gerar essa chave após a divulgação das chaves mestras. As chaves par-a-par são geradas a partir de um conjunto das chaves mestras da vizinhança, tornando mais difícil a descoberta dessas chaves pelo inimigo. O estabelecimento de todas as chaves é feito a partir de três mensagens enviadas em modo difusão pelo nó, tornando o protocolo muito eficiente em termos de energia.

A principal contribuição desta parte do trabalho é o protocolo de distribuição de chaves, pelo qual cada chave tem sua validade restrita à vizinhança onde se encontra. Dessa forma, o comprometimento de uma chave também tem efeito restrito à vizinhança do nó. Se um inimigo capturar um nó e descobrir suas chaves, não poderá realizar ataques que comprometam outras partes da rede. Assim, não é possível para um inimigo realizar um ataque em larga escala apenas a partir da captura de poucos nós. Além disso, o custo energético da solução aqui apresentada é menor que das outras até então apresentadas.

O NEKAP é um protocolo para estabelecimento de chaves par-a-par e difusão em RSSF. Essas chaves podem ser usadas para autenticação, e, possivelmente, encriptação na comunicação entre dois nós vizinhos. Chaves individuais e chaves globais não fazem parte do escopo deste trabalho, por já terem sido exploradas de forma satisfatória na literatura, como em [4][5].

6.2 Protocolo

6.2.1 Estabelecimento das chaves de difusão

Inicialmente, cada nó envia seu identificador em difusão e escuta todas as mensagens enviadas em difusão por seus vizinhos (Figura 6.1, passo 1). Depois, cada nó envia, também em difusão, sua chave mestras e a primeira chave da cadeia de autenticação. Essas mensagens são encriptadas pela chave global (Figura 6.1, passo 2). Caso algum nó perca a chave de difusão de algum de seus vizinhos, ele pode solicitar a repetição da mensagem.

1. $A \Rightarrow *: Id_A$
2. $A \Rightarrow *: \{K_{MA}, K_{A1}\}_{K_G}, HMAC(K_G, K_{MA}, K_{A1})$

Figura 6.1 - Protocolo para troca das chaves mestras

Em seguida, cada nó pode gerar a chave de difusão para cada um de seus vizinhos, utilizando a função não reversível bem conhecida, de modo que $K_A = f(K_{MA})$.

A chave global e as chaves mestras têm um período de validade, depois do qual elas são apagadas do nó. A partir do estabelecimento das chaves de difusão, toda mensagem enviada por difusão será autenticada com a chave de difusão e com a próxima chave da cadeia, ambas do nó transmissor, como mostra a Figura 6.2. Caso seja necessária a encriptação da mensagem de difusão, ela será feita somente com a chave de difusão, como mostra a Figura 6.3.

$$A \Rightarrow *: m, K_{Ai,} HMAC(K_A, K_{Ai}, m)$$

Figura 6.2 - Mensagem em difusão autenticada

$$A \Rightarrow *: \{m\}_{K_A}, K_{Ai,} HMAC(K_A , K_{Ai}, m)$$

Figura 6.3 - Mensagem em difusão encriptada e autenticada

6.2.2 Estabelecimento das chaves par-a-par

Dados dois nós A e B, sua chave par-a-par será uma função de todas as chaves mestras que eles têm em comum. Para gerá-la, A e B necessitam conhecer os vizinhos em comum. Com essa finalidade, todos os nós enviam sua lista de

vizinhos em difusão (Figura 6.4). As mensagens incluem os identificadores dos vizinhos e são enviadas usando os mecanismos de autenticação em difusão.

$$A \Rightarrow *: Id_A, \{V_1, V_2, \ldots, V_n\}_{K_A}, K_{A2}, HMAC(Id_A, K_A, K_{A2}, V_1, V_2, \ldots, V_n)$$

Figura 6.4 - Envio de informações sobre a vizinhança

Uma vez que o nó tenha recebido a lista de vizinhos dos seus próprios vizinhos, ele pode gerar a chave par-a-par que será usada na comunicação com cada um desses vizinhos. Essa chave será formada da seguinte forma: Sejam A um nó e B um de seus vizinhos, e sejam V_A e V_B seus respectivos conjuntos de vizinhos. Então, a chave par-a-par entre A e B, K_{AB} será:

$$K_{AB} = f(Id_A, Id_B, K_i \mid i \in V_A \cap V_B),$$

onde f é uma função não reversível. Os identificadores de A e B também devem ser usados como parâmetros da função, para evitar que sejam geradas chaves idênticas em uma vizinhança próxima.

Nesse momento, a fase de estabelecimento de chaves pode ser encerrada, e as chaves mestras e globais são apagadas. O custo do estabelecimento de mensagens é de apenas três mensagens.

6.2.3 Inserção de Novos Nós

O protocolo prevê também momentos de inserção de novos nós, feita para manter a densidade da rede depois da interrupção do funcionamento de alguns nós, seja por problemas de hardware ou exaustão de bateria. No momento de inserção de nós, os novos nós devem ser capazes de reconhecer e autenticar os antigos, e vice-versa. Além disso, devem ter um mecanismo para estabelecer as chaves de forma segura.

Um requisito para a inserção de novos nós é o conhecimento prévio, por parte dos nós antigos, de informações que podem autenticá-los junto aos novos. Estes, por sua vez, podem conhecer uma chave global que seja divulgada para os nós antigos.

A inserção de novos nós é um problema em esquemas de autenticação par-a-par [57]. Um inimigo pode utilizar a inserção de nós para inserir também os seus nós maliciosos. Para evitar a inserção de nós inimigos, deve ser garantido o controle de acesso à rede durante a inserção de novos nós. Para

tanto, os nós antigos devem ser capazes de reconhecer novos nós autênticos e também os nós novos devem reconhecer os nós antigos autênticos. E devem também ser capazes de estabelecer chaves entre eles.

Três requisitos devem ser garantidos durante a inserção dos novos nós: a autenticidade dos novos nós junto aos antigos, a autenticidade dos nós antigos junto aos novos e a confidencialidade do processo de troca de chaves entre esses nós. Todos esses requisitos poderiam ser alcançados apenas pelo compartilhamento de uma chave global entre os nós antigos e os novos, caso não existissem nós intrusos na rede. Mas, como a possibilidade de existência de intrusos existe, é necessário o uso de outros mecanismos para minimizar o efeito da presença de um nó malicioso na rede.

Para evitar que um inimigo insira seus nós na rede durante o processo de inserção de novos nós, será introduzido o conceito de rótulo de inserção. Um rótulo de inserção é conjunto de informações secretas, pré-carregadas nos nós antes do seu lançamento, que devem ser utilizadas durante o processo de inserção de novos nós. Um rótulo de inserção é composto do resumo HMAC das seguintes informações: identificador único do nó; chave mestra do nó; primeira chave da cadeia de chaves para autenticação em difusão; identificador do processo de inserção de nós para o qual esse rótulo deve ser usado; identificador do processo de inserção de nós no qual o nó foi inserido.

O rótulo de inserção é autenticado com uma chave global, que faz parte de uma cadeia de chaves, conhecida apenas pela estação base. A cada fase de inserção, uma chave dessa cadeia é usada. O protocolo se inicia com o conhecimento da vizinhança. Cada nó deve enviar seu identificador em difusão. Esse processo é iniciado pelos novos nós e seguido pelos nós antigos (Figura 6.5, passo 1). Em seguida, informações necessárias para o estabelecimento das chaves são enviadas junto aos rótulos de inserção (Figura 6.5, passo 2), também em difusão. Essas informações serão encriptadas com a mesma chave global usada para gerar o rótulo de inserção. Essa chave é previamente conhecida apenas pelos novos nós. Assim, eles conseguem decifrar as informações dos nós antigos. O próximo passo é o encerramento da fase de inserção de nós (Figura 6.5, passo 3). A mensagem é enviada pela estação base em difusão *multi-ponto*.

A seguir, a estação base divulga a chave global de inserção da fase (Figura 6.5, passo 4). Essa chave é encriptada e enviada par-a-par, para evitar sua escuta pelo inimigo.

1. $A \Rightarrow *: Id_A$
2. $A \Rightarrow *: \{Id_A, K_{MA}, K_{A1}, i, n\}_{K_{In}}, HMAC(K_{In}, Id_A, K_{MA}, K_{A1}, i, n)$
3. $EB \Rightarrow\Rightarrow *: ENCERRA(n), HMAC(K_{In}, ENCERRA(n),n)$
4. $EB \Rightarrow\Rightarrow *: \{K_{In}\}_*$

Figura 6.5 - Autenticação durante a inserção de nós

Como essa chave faz parte de uma cadeia de chaves, e os nós antigos têm a chave anterior, eles reconhecem a autenticidade dessa chave. A primeira chave dessa cadeia é a primeira chave global, usada no primeiro lançamento de nós. Como essa cadeia é armazenada na estação base, ela pode ser criada tão longa de modo a não ser necessário gerar uma nova cadeia durante a vida útil da rede.

Assim, os nós poderão usar essa chave para verificar que o rótulo de inserção está correto, garantindo assim a autenticidade dos nós. Os rótulos de inserção para as fases posteriores à inserção do nó devem ser pré-distribuídos, pois o desconhecimento da chave global impede os nós de gerá-los.

Após trocarem as informações autenticadas e garantirem sua autenticidade através da chave divulgada pela estação base, os novos e antigos nós já conhecem quem são seus vizinhos e trocam as informações necessárias para o estabelecimento das chaves par-a-par, através das chaves mestras e do conhecimento da vizinhança. O conhecimento da vizinhança pode ser feito através do anúncio dos identificadores, assim como é feito no estabelecimento inicial de chaves par-a-par (Figura 6.4). Assim, as chaves par-a-par podem ser estabelecidas da mesma forma que são estabelecidas durante a inicialização.

6.3 Descrição do protocolo

Neste esquema, uma chave global K_G e uma mestra K_{Mi} são carregadas no nó antes de seu lançamento. Após o lançamento, a chave mestra K_{Mi} é encriptada com a chave K_G e enviada localmente em difusão para todos os

vizinhos. Assim como no LEAP, K_G tem um período curto de validade, suficiente para a troca das chaves mestras (Figura 6.1) .

A chave de difusão é a primeira chave a ser estabelecida e é gerada a partir da chave mestra enviada. Cada nó, ao receber a chave mestra de seus vizinhos, gera a chave de difusão para cada um deles. Para tanto, usa uma função não reversível conhecida. A Figura 6.6 apresenta o conjunto dos vizinhos de um nó sensor A, delimitados pelo alcance do seu rádio. Depois do estabelecimento da chave de difusão, todo nó deve ter uma chave de difusão para cada um de seus vizinhos.

Como no LEAP, além da chave de difusão, será necessário o uso de uma chave de uma cadeia de chaves para garantir a autenticação das mensagens. A cada mensagem enviada, uma das chaves da cadeia também será enviada. Assim, o nó que recebe a mensagem pode verificar a validade da chave. A primeira chave da cadeia deve ser enviada junto à chave mestra, durante a fase de estabelecimento de chaves.

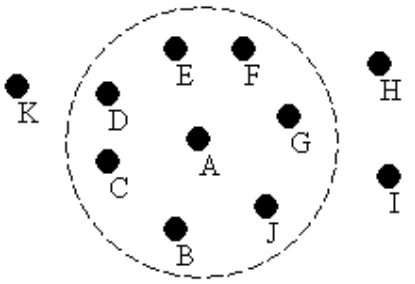

Figura 6.6 - Vizinhos do nó sensor A

Ao término da cadeia de chaves, que é finita e pequena, devido a restrições de memória, faz-se necessário a geração de uma nova cadeia. O nó gera sua nova cadeia de chaves e envia a primeira chave para seus vizinhos. No LEAP, essa chave é enviada para cada vizinho em separado, autenticada com a chave par-a-par, com o custo de uma mensagem por vizinho. Para evitar esse custo alto, o protocolo NEKAP envia a primeira chave da nova cadeia em difusão, autenticando essa chave com a última chave da cadeia anterior.

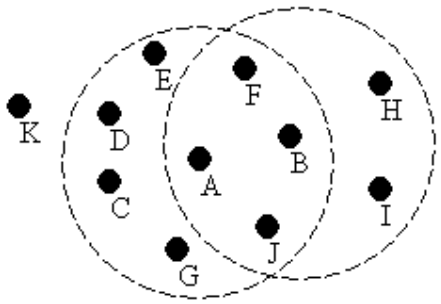

Figura 6.7 - Intersecção dos vizinhos de A e vizinhos de B

Depois da troca das chaves mestras, quaisquer dois nós vizinhos A e B terão um conjunto de chaves em comum, K_{MA}, K_{MB}, e ainda $K_{MX,}$ para todo X que é vizinho de ambos, A e B. Na Figura 6.7, por exemplo, a intersecção dos dois

círculos correspondentes ao alcance do rádio dos nós A e B determina todos os nós que são vizinhos, ao mesmo tempo, de A e B, no caso os nós F e J. Tanto o nó A quanto o nó B conhecem as chaves mestras desses nós e podem usar essas informações para gerar a chave par-a-par que será usada na comunicação entre esses dois nós A e B. Para aumentar a segurança desse protocolo, são usadas todas as informações em comum, além dos identificadores dos nós. No exemplo da Figura 6.7, a chave par-a-par a ser usada pelos nós A e B seria dada por: $K_{AB} = f(Id_A, Id_B, K_A, K_B, K_F, K_J)$.

6.4 Mobilidade

Embora tenha sido concebido para redes estáticas, o protocolo NEKAP pode também ser usado em redes de baixa mobilidade. Uma restrição que se faz, nesse caso, é que o movimento do nó permita o intercâmbio de chaves entre os nós no caminho a ser percorrido. Assim, um nó em movimento pode usar seus vizinhos confiáveis para estabelecer uma relação de confiança com os novos vizinhos.

A relação de confiança será determinada pelo compartilhamento de chaves de comunicação par-a-par e difusão. O processo de estabelecimento de chaves do NEKAP instaura uma relação de confiança entre todos os pares de nós vizinhos. Durante o movimento do nó, é possível estender essa relação aos vizinhos de seus vizinhos.

Após iniciar o movimento, um nó deve indicar, periodicamente, sua presença. Os novos vizinhos que perceberem o movimento do nó devem argüir os seus vizinhos se o nó em movimento tem relação de confiança com algum deles. Caso seja identificado um vizinho em comum, este pode atuar como intermediário e repassar as informações necessárias para a geração das chaves entre os dois novos vizinhos, assim estabelecendo a relação de confiança entre eles. Esse processo deve ser totalmente encriptado, de forma a manter as chaves confidencias, para garantir a segurança do processo contra eventuais escutas inseridas por um inimigo. A Figura 6.8 apresenta as mensagens a serem trocadas para o estabelecimento das novas chaves entre os novos vizinhos. O processo pode ser otimizado para que sejam estabelecidas chaves com vários novos vizinhos ao mesmo tempo.

1. $A \Rightarrow *: Id_A$
2. $B \Rightarrow *$: Who know A?
3. $C \Rightarrow *$: I know A
4. $A \Rightarrow C: \{K_{MA}, K_{A1}, V_1, V_{2,} \ldots, V_n\}_{K_{AC}}, HMAC(K_{AC}, \{K_{MA}, K_{A1}, V_1, V_{2,} \ldots, V_n\}_{K_{AC}})$
5. $C \Rightarrow B: \{K_{MA}, K_{A1}, V_1, V_{2,} \ldots, V_n\}_{K_{CB}}, HMAC(K_{CB}, \{K_{MA}, K_{A1}, V_1, V_{2,} \ldots, V_n\}_{K_{CB}})$
6. $B \Rightarrow C: \{K_{MA}, K_{A1}, V_1, V_{2,} \ldots, V_n\}_{K_{AC}}, HMAC(K_{AC}, \{K_{MA}, K_{A1}, V_1, V_{2,} \ldots, V_n\}_{K_{AC}})$
7. $C \Rightarrow A: \{K_{MA}, K_{A1}, V_1, V_{2,} \ldots, V_n\}_{K_{CB}}, HMAC(K_{CB}, \{K_{MA}, K_{A1}, V_1, V_{2,} \ldots, V_n\}_{K_{CB}})$

Figura 6.8 – Estabelecimento de chaves durante a movimentação de nós

O estabelecimento de chaves para dar suporte à mobilidade é mais caro que aquele realizado na inicialização. Além disso, pode permitir que um nó intruso, que já tenha obtido acesso à rede, obtenha mais chaves e possa expandir seu ataque. Assim, a mobilidade de nós só deve ser usada em casos especiais, onde seja fundamental para o funcionamento da rede.

6.5 Análise de segurança

O controle de acesso com autenticação par-a-par, conforme proposto neste trabalho e também apresentado no LEAP, elimina a possibilidade de ocorrência de diversos tipos de ataques promovidos por nós externos, como escuta, inserção de dados incorretos, adulteração dos dados, alteração da origem, e também os ataques de negação de serviço no roteamento, como *Black Hole*, *selective forwarding*, *Wormhole*, entre outros. A possibilidade de ataques internos, porém, deve ser verificada.

Uma vez estabelecidas as chaves para assinatura e encriptação par-a-par, o controle de acesso é feito pelo descarte de todas as mensagens enviadas que não tenham assinatura emitida por um dos vizinhos conhecidos.

Para efetuar um ataque interno, promovido por nós maliciosos reconhecidos como legítimos pelos nós da própria rede, um inimigo começa pela captura e adulteração de um nó ou pela descoberta das chaves através de escutas ou criptoanálise. Dessa forma, um inimigo pode descobrir todas as chaves presentes em um nó, chaves que tenham sido enviadas em algum

momento, ou ainda chaves que já estiveram de posse dos nós. Por exemplo, a chave global poderia ser descoberta, mesmo se for apagada durante o processo de estabelecimento de chaves, por meio de um nó que não tenha se inicializado corretamente.

No protocolo NEKAP, as chaves são distribuídas sem qualquer relação com a localização dos nós na rede. Assim, a descoberta de uma chave em particular, ou de um conjunto de chaves de um determinado nó, não permite ao inimigo a obtenção de qualquer vantagem sobre a rede. A única chave que poderia trazer um efeito maior, se descoberta, é a chave global, que poderia levar ao conhecimento das chaves trocadas durante a inicialização. Para tanto, seria necessária a existência de uma ou várias escutas durante a inicialização. Com isso, o inimigo poderia obter as chaves par-a-par e as chaves de difusão trocadas no ponto de localização da escuta. As chaves de difusão obtidas nesse processo não são de grande valia para o inimigo, pois ele não pode inserir nem adulterar mensagens enviadas em difusão, pois é necessária também a chave correta da cadeia de chaves, desconhecida pelo inimigo. Resta, ao inimigo, utilizar as chaves par-a-par descobertas, para inserir e adulterar mensagens. Para efeito de comparação, é possível verificar que o protocolo LEAP é totalmente vulnerável à descoberta da chave global.

O inimigo pode ainda clonar o nó com suas chaves e lançar essas cópias em outros pontos da rede, realizando um ataque distribuído com a finalidade de ampliar seu efeito sobre toda a rede. Nesse trabalho, o efeito da clonagem de nós é eliminado ou amplamente reduzido. Como as chaves são limitadas à vizinhança, seu uso em outros pontos não permite ao inimigo ser reconhecido como um nó da rede.

6.5.1 Instanciação para distribuição determinística

Para realizar um estudo analítico da vulnerabilidade do NEKAP, será feita uma instanciação do problama, utilizando o modelo colméia, citado na seção 2.7.1. Nesse modelo, a distribuição de nós é uniformizada em seis direções, o que faz com que esteja bastante próxima de uma distribuição ótima. A Tabela 2.4 e a Figura 2.2 são replicadas aqui para facilitar a compreensão desta análise.

Tabela 2.4 - Número de vizinhos em função do alcance no modelo colméia

Alcance r	Número de vizinhos
r < d	0
d ≤ r < 1.73 d	6
1.73d ≤ r < 2d	12
2d ≤ r < 2.78d	18
2.78d ≤ r < 3d	30
3d ≤ r < 3.46	36
3.46d ≤ r < 3.61d	42
3.61d ≤ r < 4d	54

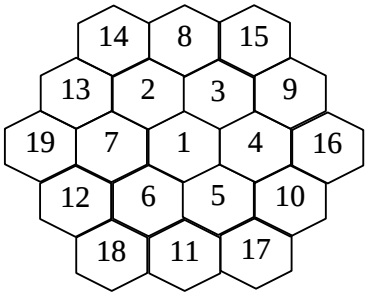

Figura 2.2 - Modelo Colméia

Considerando o caso onde o número de vizinhos é igual a seis, como no estabelecimento de chaves entre os nós marcados como 1 e 4 na Figura 2.2, por exemplo, serão usadas as chaves dos vizinhos comuns 3 e 5, além das chaves dos próprios nós, 1 e 4, e seus identificadores para gerar a chave par-a-par entre esses nós. Nenhum outro nó, além dos nós 1 e 4, será capaz de conhecer essa chave, pois os nós 3 e 5 não são vizinhos, e não conhecem as respectivas chaves. De forma análoga, é possível observar que nenhum nó conhece chaves alheias e assim, suas chaves também estão protegidas. Dessa forma, nesse modelo a adulteração dos nós não revela nenhuma chave adicional.

Considerando o caso onde o número de vizinhos é igual a 12, ainda na Figura 2.2, devem ser considerados dois tipos de vizinhos, de acordo com a distância entre eles: vizinhos com distância d, e vizinhos com distância 1,73 d. No estabelecimento das chaves do primeiro grupo, por exemplo, vizinhos 1 e 2 , eles terão como vizinhos comuns os nós 3, 4, 6, 7, 8, 13. A chave par-a-par vai ser estabelecida com base nessa vizinhança. Nenhum outro nó, além dos nós 1 e 2, é capaz de conhecer toda essa vizinhança. Assim, nenhum nó vai obter essa chave. Para o segundo grupo, como exemplo, nós 1 e 13, os vizinhos em comum são 2, 7, 8 e 12. Nenhum outro vizinho conhece toda essa vizinhança. O nó 2 não é vizinho de 12. O nó 7 não é vizinho do 8, e assim, nenhum vizinho é capaz de obter essa chave. E, de forma análoga, é possível observar que nenhum nó conhece chaves alheias e assim, suas chaves também estão protegidas. Também

dessa forma, nesse modelo a adulteração dos nós não revela nenhuma chave adicional.

Considerando o caso onde o número de vizinhos é igual a 18, devem ser considerados três tipos de vizinhos: os dois tipos analisados no caso anterior e, ainda, os vizinhos que estão à distância igual a 2d. No estabelecimento das chaves dos dois primeiros grupos, não há, também, nenhum nó capaz de descobrir chaves de outros nós. Entretanto, no estabelecimento de chaves entre nós que estão à distância 2d, como os nós 4 e 7 na figura, a vizinhança em comum, formada pelos nós 1, 2, 3, 5, 6, 8 e 11, é totalmente conhecida pelo nó 1, que é capaz de computar também a chave par-a-par estabelecida entre os nós 4 e 7. Para tanto, precisa armazenar as chaves mestras trocadas durante a inicialização, o que só vai acontecer se o nó tiver, desde a inicialização, tendências maliciosas, introduzido por um intruso, ou adulterado logo após seu lançamento. O nó 1 também poderia conhecer as chaves estabelecidas entre os nós 2 e 5, e 3 e 6. Logo, cada nó com intenções maliciosas poderia descobrir três chaves par-a-par estabelecidas entre seus vizinhos.

6.5.2 Modelo de simulação

O modelo analítico mostrou que, para o modelo colméia, adotado neste trabalho, nenhum nó é capaz de descobrir chaves utilizadas por outros nós, a menos que a densidade da rede seja tal que cada nó tenha 18 vizinhos ou mais. Como, entretanto, na prática, a distribuição de nós nem sempre segue o modelo colméia, será apresentado um modelo de simulação, no qual as distribuições de nós se distinguem do modelo colméia, incluindo até mesmo uma distribuição totalmente aleatória. As distribuições adotadas no modelo de simulação vão incluir redes de densidade média próxima de 20 nós vizinhos, pois essas redes podem apresentar um maior número de chaves conhecidas em uma vizinhança próxima. Os parâmetros da simulação serão o tamanho da rede e seu modelo de distribuição. O objetivo da simulação é identificar quantas chaves par-a-par podem ser descobertas indevidamente por um nó qualquer da rede com intenções maliciosas. Um nó com intenções maliciosas pode ser resultado da inserção de uma escuta e obtenção da chave global ou pela adulteração de um nó autêntico. Uma escuta durante o processo de inicialização, com a descoberta

da chave global, tem o mesmo efeito da adulteração de um nó na fase inicial, sob o ponto de vista dessa simulação, pois o objetivo é descobrir as chaves mestras trocadas em um ponto da rede.

As simulações foram baseadas na distribuição de 1024 nós em uma área de dimensões 100 x 100 unidades de distância. Cada simulação utilizou uma forma diferente de distribuição dos nós, a saber:

1. Aleatório: 1024 nós sensores, com coordenadas x e y geradas aleatoriamente, entre 0 e 100;
2. Faixas: A área foi dividida em 10 faixas de 10 x 100. Em cada faixa foram distribuídos 102 nós sensores, de forma aleatória. O objetivo dessa distribuição é simular o lançamento através de um avião, que pode sobrevoar a região diversas vezes para cobrir toda a área alvo;
3. Pertuba30: A partir de uma distribuição uniforme, onde os nós são lançados em 32 linhas de 32 nós cada, igualmente distantes, cada nó é deslocado de sua posição de uma distância igual à distância entre os nós multiplicada um valor aleatório, obtido através de uma distribuição normal entre 0 e 1, multiplicada por um fator de 0,3.
4. Pertuba10: Idêntico ao modelo anterior, mas com a distância de deslocamento dos nós multiplicada por um fator de 0,1;
5. Pertuba5: Idêntico ao modelo anterior, mas com a diferença que a distância de deslocamento dos nós é multiplicada por um fator de 0,05;
6. Uniforme: os nós sensores são distribuídos de forma determinística em 32 linhas com 32 nós cada, com distância fixas entre os nós.

Essas distribuições foram usadas para avaliar o quanto a aleatoriedade na distribuição de nós pode influenciar o protocolo NEKAP.

A Figura 6.9 apresenta graficamente as distribuições de nós sensores. Verifica-se a redução da aleatoriedade a partir da primeira para a última distribuição.

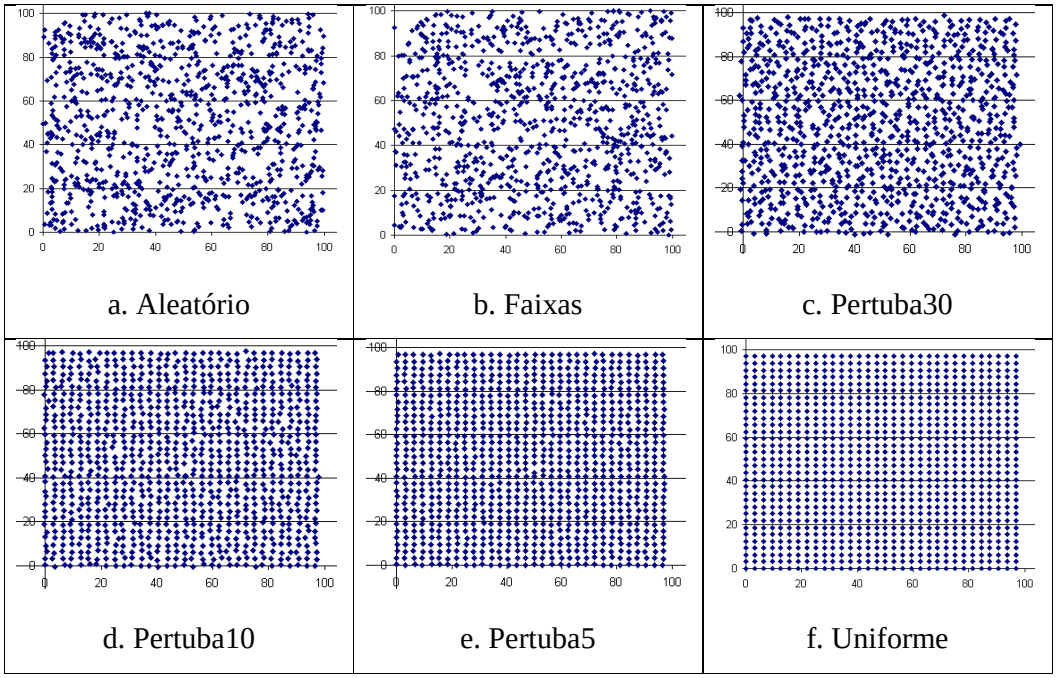

Figura 6.9 - Distribuições de nós simuladas

A intenção de um inimigo é descobrir o maior número possível de chaves. As chaves de difusão não são úteis, uma vez que a autenticação em difusão depende também das chaves da cadeia, que não são divulgadas. Logo, as chaves par-a-par têm maior interesse do inimigo. Foi verificado, então, o número de chaves par-a-par que podem ser conhecidas através da adulteração de um nó na inicialização. Os resultados da simulação podem ser vistos na Tabela 6.1. Foram coletados os seguintes dados:

I. Numero médio de vizinhos;
II. Número total de chaves par-a-par estabelecidas;
III. Número médio de chaves par-a-par adicionais que um inimigo conheceria adulterando um nó na fase de inicialização.

	I	II	III
Aleatório	19,28	9875	30,61
Faixas	19,10	9782	28,11
Pertuba30	18,20	9320	17,33
Pertuba10	18,70	9575	11,44
Pertuba5	18,64	9546	10,73
Uniforme	18,64	9546	10,71

Tabela 6.1 - Resultados da simulação

Essa última informação (III) é representativa para cada nó. Durante a simulação, são verificadas quantas chaves cada nó poderia obter, caso tivesse comportamento malicioso no momento de sua inicialização. Os valores apresentados na coluna III representam a média do número de chaves obtidas em cada nó. Para avaliar o impacto dessa descoberta, seja considerado, por

exemplo, o caso médio, na simulação Pertuba30. Pela coluna I, os nós têm, em média, 18 vizinhos. São estabelecidas, assim, na vizinhança desse nó, 171 chaves par-a-par, que representa a combinação dos 19 nós da vizinhança tomados 2 a 2. A simulação mostrou que um nó adulterado na inicialização poderia revelar, além das 18 chaves par-a-par que ele estabelece com seus vizinhos, outras 17 chaves par-a-par extras. Essas 17 chaves representam a vulnerabilidade do protocolo, pois são chaves desnecessárias para aquele nó. Esse número, porém, é muito baixo se comparado com outras abordagens.

Uma conclusão interessante obtida a partir dessa simulação é que a distribuição aleatória é mais vulnerável que a distribuição uniforme e existe uma tendência clara que, quanto maior a aleatoriedade da distribuição, maior é o número de chaves que podem ser descobertas por um nó malicioso durante a inicialização. Esse fato pode ser explicado pela maior aglutinação de alguns nós, formando grupos maiores e mais próximos. Nesses grupos, a interseção de vizinhanças é maior e o número de chaves descobertas também. Nos modelos de distribuição mais uniformes, os nós se encontram mais eqüidistantes e normalmente o número de vizinhos é mais constante. Já nas distribuições mais aleatórias, o número de vizinhos varia muito de uma região para outra e, regiões com alta densidade favorecem a descoberta de chaves pois mais nós compartilham a mesma região.

Para efeito de comparação, no protocolo LEAP, caso o inimigo adultere um nó durante sua inicialização e obtenha a chave global, ele será capaz de descobrir todas as chaves par-a-par da rede, e não apenas aquelas trocadas na vizinhança. Na simulação Pertuba30, isso representaria 9320 chaves par-a-par.

Assim, o protocolo NEKAP, aqui apresentado, resiste bem melhor às diversas condições de ataques e ainda à quebra da pressuposição inicial de segurança da chave global durante a inicialização, utilizada no LEAP.

6.6 Implementação

A implementação do protocolo de distribuição de chaves aqui apresentado foi realizada para o nó sensor Mica2 Motes [35], com o sistema operacional Tiny OS, utilizando as funções de criptografia já existentes.

O algoritmo de criptografia escolhido para gerar o campo HMAC da mensagem foi o RC5, no modo CBC-MAC. Esse mesmo algoritmo pode ser usado como função irreversível, para gerar a cadeia de chaves e as chaves par-a-par e de difusão.

A memória necessária está dentro dos parâmetros do Mica2 Motes. O código total do protocolo compilado com Tiny OS resulta em memória de programa de 10 Kbytes. Será necessário armazenar as seguintes chaves: chaves de difusão, uma por vizinho; chaves par-a-par, uma por vizinho; uma cadeia de chaves para uso pelo nó; última chave divulgada da cadeia de chaves de cada vizinho. Considerando a média de 20 vizinhos e uma cadeia de chaves com 20 chaves, cada nó necessita armazenar 80 chaves. Considerando, ainda, chaves de 8 bytes, ou 64 bits, teremos um total de 640 bytes reservados para armazenar chaves. O nó Mica2 Motes conta com 128 K de memória de programa e 4 K de memória RAM, de forma que a utilização do protocolo descrito neste artigo é viável para esse nó.

6.7 Trabalhos relacionados

O protocolo LEAP [7] é a proposta para distribuição de chaves que mais se aproxima deste trabalho. O protocolo foi descrito e discutido na seção 3.2.1. A diferença do trabalho proposto neste capítulo para o LEAP é a forma como a chave global inicial é usada. No LEAP, essa chave pode ser usada a qualquer momento para gerar novamente as chaves par-a-par. Logo, essas chaves podem ser descobertas a qualquer momento, bastando a um inimigo descobrir a chave global. E isso pode ser feito através de algum nó que não se inicializou corretamente ou através de algum método de criptoanálise. Neste trabalho, isso não é possível, pois a chave global é usada somente durante a inicialização, para a troca das chaves mestras, que, por sua vez, são usadas para a geração das chaves par-a-par. Assim, a descoberta da chave global não revela nenhuma informação posterior neste trabalho.

Eschenauer & Glicor [29], Chan *et al.* [28] e Liu e Ning [26] apresentam protocolos de pré-distribuição aleatória de chaves. O primeiro problema dessa abordagem é que, caso dois nós não compartilhem chaves, eles devem estabelecer chaves através de outros nós que tenham conseguido estabelecer

comunicação segura entre eles e isso pode aumentar o consumo de energia. Além disso, mensagens em modo de difusão não são aceitas. Logo, o envio de mensagens em modo de difusão pelo nó, amplamente utilizado em diversos algoritmos de roteamento só pode ser realizado pelo envio de uma mensagem para cada vizinho, o que multiplica o consumo de energia pelo número de vizinhos. Outro problema dessa abordagem é a suscetibilidade a nós adulterados. Um nó adulterado pode se comunicar com diversos outros nós em diversos locais da rede, de forma que a clonagem de nós adulterados pode ter um impacto muito alto na rede. NEKAP não tem esses problemas, sendo mais eficiente em termos de consumo de energia e mais seguro por não permitir que nós adulterados sejam clonados e inseridos em vários pontos da rede.

6.8 Conclusões

Este capítulo apresentou o protocolo NEKAP, para estabelecimento de chaves para RSSF. As chaves são usadas para garantir a autenticação par-a-par e conseqüente controle de acesso. Essa abordagem possibilita eliminar diversos tipos de ataques promovidos por inimigos. O principal objetivo deste trabalho é realizar o controle de acesso no roteamento para dificultar a realização de ataques pelo inimigo sem, no entanto causar impacto no consumo de energia.

Para comprometer a comunicação em uma rede que utiliza o NEKAP, um adversário deve estar fisicamente presente na vizinhança e adulterar um nó da rede. Pode, ainda, ouvir passivamente e coletar todas as mensagens usadas na inicialização da rede e descobrir a chave global de inicialização. Mesmo assim, um ataque bem sucedido tem impacto apenas local. Além disso, para gerar as chaves, esta solução requer apenas três mensagens enviadas em difusão a partir de cada nó, em oposição à comunicação par-a-par entre todos os pares de vizinhos da vizinhança, usada pelo LEAP. Assim, a solução aqui apresentada é também mais eficiente em termos de energia e tempo de configuração. As simulações realizadas indicam que as ações do inimigo não trazem efeitos muito grandes para a rede, afetando, no máximo, alguns poucos nós.

A solução aqui apresentada, em conjunto com outras soluções já existentes na literatura, como autenticação fim-a-fim [4], possibilita um

aumento significativo da eficiência da rede na presença de um inimigo e impede a maioria dos ataques conhecidos.

Capítulo 7

Rotas Alternativas para Detecção e Aumento da Resiliência à Intrusão Distribuída

7.1 Introdução

O capítulo anterior apresentou uma proposta de estabelecimento de chaves entre vizinhos para promover o controle de acesso no enlace, evitando a presença de intrusos no roteamento. Essa estratégia, entretanto, não é eficaz contra a ocorrência de *tampering*, quando um nó da rede é adulterado por um inimigo, que pode obter todas as informações do nó e ter acesso a qualquer protocolo da rede. Além disso, algumas aplicações podem não suportar ou simplesmente abdicar de criptografia, exigindo outro tipo de solução. Assim, este capítulo apresenta outra abordagem que pode ser usada em conjunto com a criptografia ou de forma independente.

Este capítulo apresenta uma estratégia para prover as redes de sensores sem fio com habilidades de tolerância à intrusão e, por conseguinte, aumentar a resiliência das mesmas. A estratégia prevê a criação de rotas alternativas no roteamento, o que também contribui com a detecção de intrusos. O algoritmo de roteamento TinyOS beaconing [14] foi modificado para que cada nó repassador utilize dois caminhos para enviar suas informações para a estação base. Caso um intruso esteja presente em um desses caminhos, inutilizando-o, o caminho alternativo pode continuar funcionando, garantindo a entrega de parte das informações. O capítulo apresenta o algoritmo modificado, bem como uma avaliação do seu desempenho em termos de energia e eficácia. O desempenho foi verificado por simulação e os resultados mostraram uma boa eficiência mesmo para um alto número de intrusos.

As rotas múltiplas são caminhos redundantes no roteamento, usadas de forma alternada e sem replicação de informações. A alternância de rotas aumenta a tolerância da rede a intrusos, visto que oferece uma opção a mais de roteamento. Caso exista um intruso em uma das rotas, uma rota alternativa

possibilita o encaminhamento dos pacotes por ela. Além disso, fazendo-se análise dos pacotes recebidos, é possível descobrir rotas que não entregam os pacotes corretamente e que estejam causando problemas no roteamento. A alternância de rotas foi escolhida para manter o consumo de energia próximo daquele verificado com rotas simples.

A utilização de alternância de rotas deste caso contribui para o aumento da resiliência da rede e ainda permite a detecção eficiente de nós intrusos. Nas simulações apresentadas, observa-se que o algoritmo de detecção de intrusos tem grande eficiência na presença de poucos intrusos, mas ainda consegue identificar boa parte destes em simulações com grande número de nós intrusos, sendo capaz de detectar todos em sucessivas fases de detecção e revogação de nós.

Vários trabalhos abordam o uso de rotas múltiplas em RSSF [23], [45] e [46]. As rotas múltiplas podem ser usadas de forma redundante ou de forma alternada. Rotas usadas de forma redundante aumentam a tolerância a falhas, porém aumentam o consumo de energia, pois replicam as informações pela rede em diversos caminhos. O uso de rotas de forma alternada, neste trabalho, foi escolhido para manter o consumo de energia bem próximo daquele verificado sem os mecanismos propostos.

A proposta deste trabalho, utilizando alternância de rotas, contribui para o aumento da resiliência e ainda permite a realização de detecção nós intrusos de forma eficiente. Nas simulações apresentadas, observa-se que o algoritmo de detecção de intrusos tem grande eficiência na presença de poucos intrusos, mas ainda consegue identificar boa parte destes em simulações com grande número de nós intrusos.

7.2 Rotas alternativas

Rotas múltiplas podem ser disjuntas, com todos os nós distintos entre as rotas; ou entrelaçadas, quando contêm nós em comum. Ganesan *et al.* apresentam uma comparação entre rotas múltiplas disjuntas e entrelaçadas. Rotas disjuntas são mais resilientes a falhas e intrusão [46]. Rotas entrelaçadas são mais baratas em termos de consumo de energia para criação e manutenção. Um ponto de falha em um nó comum, porém, pode inutilizar todas as rotas

existentes. Vários algoritmos de roteamento foram propostos para RSSF nos quais os mecanismos utilizados para a criação e manutenção de rotas em cada um destes algoritmos são justificados em função do tipo de rede e do tipo de aplicação. Existem muitas formas de criar rotas múltiplas para cada protocolo. Este trabalho, porém, restringe-se ao protocolo conhecido como PI, ou Propagação da Informação [58], usado no Sistema Operacional *Tiny OS,* onde é conhecido como *Tiny OS beaconing*.

7.2.1 Estabelecimento

As rotas alternativas são estabelecidas utilizando como base o algoritmo de roteamento conhecido como *Tiny OS beaconing*. Esse algoritmo é baseado no uso de um *beacon*, enviado em modo de difusão pela estação base. Cada nó determina qual dos seus nós vizinhos será o próximo salto em direção à estação base a partir da recepção do *beacon*. A ordem de recepção do *beacon* pelos nós é que determina a criação de rotas. Para a criação de rotas múltiplas, o nó deve estabelecer mais rotas além daquela originada no nó que enviou o *beacon* primeiro. Assim, a segunda rota é estabelecida a partir do segundo nó vizinho que repassar o *beacon*. Esse processo não garante a criação de rotas disjuntas, porém, garante que cada nó terá saídas alternativas no primeiro salto, através de dois nós de saída. A rota criada após o recebimento do *beacon* pela primeira vez será tratada como rota padrão e a rota criada após o recebimento do *beacon* pela segunda vez será tratado como rota alternativa.

O algoritmo de roteamento considera as mensagens originadas em um nó de forma diferente daquelas geradas por outros nós e que passam pelo nó em questão. As mensagens geradas por um nó serão enviadas de forma alternada pelas rotas que passam pelo mesmo nó, ou seja, uma vai pela rota padrão e a próxima vai pela rota alternativa. As mensagens que são apenas repassadas serão sempre enviadas pela rota padrão. Dois objetivos motivaram essa estratégia. Caso mensagens repassadas pudessem seguir pela rota alternativa, o caminho desde um nó até a estação base seria alternativo a cada salto, criando inúmeras possibilidades. Porém, como é necessário registrar o caminho percorrido pelo pacote, essa estratégia se tornaria cara. Além disso, caso os pacotes repassados pudessem seguir pela rota alternativa, poderia ocorrer laços

que levariam ao encarecimento do roteamento e até mesmo impedimento de entrega de alguns pacotes.

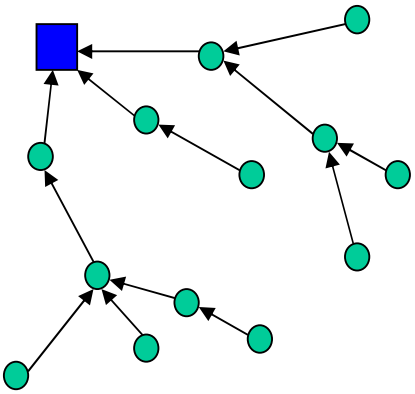

Figura 7.1 - Árvore gerada pelo Tiny OS

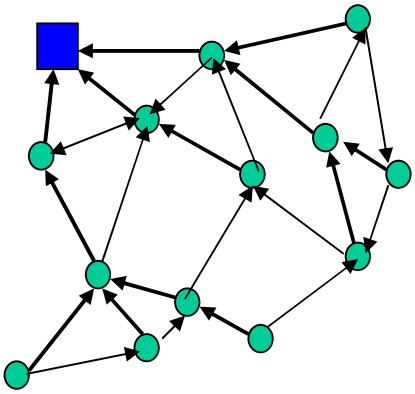

Figura 7.2 - Grafo resultante com rotas múltiplas

A Figura 7.1 e a Figura 7.2 exemplificam a criação de rotas alternativas em uma rede pequena. As setas mais escuras indicam as rotas padrão; as mais claras, as rotas alternativas.

Após o estabelecimento das rotas é necessário um mecanismo para decidir qual rota deve ser usada a cada momento, de forma imprevisível para o inimigo, mas conhecida para a estação base. A decisão da rota deve levar em consideração a detecção e o isolamento dos intrusos: o número de mensagens que passam por cada uma das duas rotas deve ser comparável ao da outra rota e tanto a estação base quanto o nó devem saber *a priori* a rota usada para cada mensagem. Assim, quando uma mensagem não chega, a estação base sabe por onde ela foi roteada.

7.2.2 Conhecimento da topologia

Para identificar intrusos, a estação base deve conhecer a topologia da rede. Mecanismos de conhecimento da topologia já estão disponíveis na literatura. Staddon *et al.* propõe uma forma de conhecimento da topologia pelo envio do identificador de um nó vizinho em cada medida efetuada pelo nó sensor [47]. Com isso o custo de energia é menor, pois não existem mensagens extras. Porém, o tempo necessário para o conhecimento da topologia é maior. A opção

por um mecanismo capaz de descobrir a topologia em um tempo menor permite a detecção de intrusos desde os momentos iniciais de funcionamento da rede.

Neste trabalho, para a estação base conhecer a topologia da rede, cada nó envia uma mensagem indicando quais são os vizinhos responsáveis pelas rotas até a estação base. Após o recebimento das informações dos nós, a estação base é capaz de criar um grafo de conectividade da rede, que será usado no algoritmo de detecção de intrusos. Nós participantes de rotas em silêncio, ou gerando um número baixo de informações, serão marcados no grafo para possibilitar a localização de intrusos.

7.3 Detecção de intrusos

O comportamento da rede diante de nós intrusos é diferente do comportamento diante de falhas. No caso de uma falha, a rede pode contornar o problema identificando outro nó que possa assumir a função do nó com falha. Uma nova execução do algoritmo de estabelecimento de rotas pode resolver o problema ocasionado por uma falha. No caso de uma intrusão, a rede deve, antes, isolar o nó intruso. Somente a execução do algoritmo de estabelecimento de rotas não elimina o intruso, pois ele vai participar desse processo e se inserir novamente na árvore de roteamento.

No algoritmo proposto, a produção de cada nó por cada rota é verificada pela estação base para que ela possa descobrir a existência de intrusos. Esse mapa de produção vai indicar a diferença entre o número de mensagens esperadas pela estação base, provenientes de cada nó, e o número de mensagens efetivamente recebidas, o que mostra a perda em cada rota. Para obter esse número, porém, a estação base deve identificar o caminho percorrido por cada pacote recebido e contabilizar os pacotes por nó e rota de origem.

O algoritmo pode identificar intrusos usando esse mapa de produção. O algoritmo é recursivo. Ele inicia na estação base e segue em direção aos nós folha usando a árvore formada pelas rotas padrão. A cada passo um nó é avaliado. Para esse nó, é verificada a diferença entre as perdas pelas rotas padrão e alternativa de cada nó que dependem desse nó para o roteamento. Caso as perdas apresentem diferenças significativas entre a rota padrão e a rota alternativa, o nó responsável pela rota de maior perda é marcado como intruso.

Caso um nó não apresente resposta em nenhuma das rotas, ele é marcado como nó em silêncio.

O resultado da execução do algoritmo é a marcação de nós em silêncio e de possíveis nós intrusos. À medida que o algoritmo é executado, uma pontuação referente ao grau de intrusão pode ser incrementada. Quanto maior a pontuação, mais evidente é a presença de um intruso naquele nó, pois maior é sua interferência na produção da rede. A pontuação de um possível intruso é incrementada caso seja identificado nós em silêncio no caminho de roteamento que passa por esse nó intruso. Assim, os pontos do intruso aumentarão devido aos nós em silêncio que dependem dele.

Nós com falhas intermitentes ou canais com muitos erros podem ser confundidos com a presença de intrusos realizando ataques do tipo Encaminhamento Seletivo. As falhas intermitentes, porém, tem padrão aleatório. Já esses ataques tendem a seguir uma lógica, restringindo os pacotes com base em alguma informação disponível, como a origem. Mesmos os ataques de Encaminhamento Seletivo, nos quais os pacotes são repassados aleatoriamente, podem ser distinguidos das falhas intermitentes pelo percentual de perda. Especialmente porque este trabalho não está interessado em ataques que tenham baixo percentual de perda.

O Algoritmo 6.1 apresenta apenas as inicializações e a chamada à função Detecta Intruso, mostrada no Algoritmo 6.2. Na inicialização do algoritmo são calculadas as perdas em cada uma das rotas, além da inicialização das variáveis e da chamada da função de detecção de intrusos a partir da estação base.

Pontos do Intruso ← 0;

Nó Intruso ← vazio;

Para todo Nó I pertencente à rede

1. Perda padrão (I) ← Número de mensagens enviadas na rota padrão – número de mensagens recebidas na rota padrão / Número de mensagens esperadas na rota padrão
2. Perda alternativa (I) ← Número de mensagens enviadas na rota alternativa – número de mensagens esperadas na rota alternativa / Número de mensagens enviadas na rota alternativa

Detecta intruso (Estação Base, Intruso, Pontos do Intruso)

Algoritmo 6.1 - Inicializações e chamada da detecção de intrusos

O Algoritmo 6.2 apresenta a detecção de intrusos, que é realizada de forma recursiva a partir da estação base. Cada execução do algoritmo tem um nó em foco, na variável X. Esse algoritmo pode ser dividido em três partes: A verificação das perdas nos nós vizinhos do nó X que o utilizam como rota padrão, a verificação das perdas nos nós vizinhos do nó X que o utilizam como rota alternativa e a chamada da função, de forma recursiva, para os nós vizinhos que usam X como rota padrão.

```
Detecta intruso (Nó X, Nó Intruso, Pontos do Intruso)
    1. Para cada nó I, vizinho de X que utiliza esse nó como rota padrão:
        a. Se Perda padrão (I) >> Perda alternativa (I) então
            i. Se Pontos do Intruso = 0 então
                    Intruso ← X;
            ii. Incrementa Pontos do Intruso;
        b. Se Perda padrão (I) = Perda alternativa (I) = 100 % então
            i. Se Pontos do Intruso ≠ 0
                    Incrementa Pontos do Intruso;
            ii. Senão
                    Marcar I como Falha
    2. Para cada nó I, vizinho de X que utiliza esse nó como rota alternativa
        a. Se Perda alternativa (I) >> Perda padrão (I) então
            i. Se Pontos do Intruso = 0 então
                    Intruso ← X;
            ii. Incrementa Pontos do Intruso;
        b. Se Perda padrão (I) = Perda alternativa (I) = 100 % então
            i. Se Pontos do Intruso ≠ 0
                    Incrementa Pontos do Intruso;
            ii. Senão
                    Marcar I como Falha
    3. Para cada nó I, não marcado como falha, vizinho de X que utiliza esse nó como rota padrão:
        a. Se Pontos do Intruso = 0 então
            i. Detecta intruso(I, Intruso, Pontos do Intruso)
            ii. Se Pontos do Intruso ≠ 0 então
                    Marcar Intruso e Pontos do Intruso
                    Pontos do Intruso ← 0
                    Intruso ← vazio;
        b. Senão
            i. Detecta intruso(I, Intruso, Pontos do Intruso)
Fim Detecta Intruso
```

Algoritmo 2 – Algoritmo recursivo para detecção de intrusos

Caso o nó X seja um nó intruso, os nós vizinhos de X que dependem desse nó devem apresentar taxas altas de perdas na rota que passa por X. O mesmo não deve acontecer na outra rota. A primeira e a segunda parte do

algoritmo 2 verificam a diferença entre as perdas das duas rotas. Caso exista uma diferença grande entre essas perdas, o nó é um provável intruso. Para possibilitar uma maior acurácia na identificação dos intrusos, os prováveis nós intrusos recebem uma pontuação que é incrementada sempre que houver a suspeita que esse nó é um intruso.

A pontuação do intruso representa a extensão de seu ataque. Quanto mais abrangente o ataque maior será a pontuação. Caso a rota que passa por um intruso identificado tenha poucos elementos, a pontuação do intruso também será baixa. O significado dessa baixa pontuação é um ataque que afeta poucos nós. Os nós que não tem nenhum filho na árvore de roteamento, conseqüentemente, não são detectados. Todavia, esses nós não representam problema, uma vez que seu ataque não é efetivo.

A partir do momento que um nó intruso é identificado, todos os indícios de intrusão percebidos nas rotas que dependem desse nó são associados a ele. Essa associação é realizada na terceira parte do algoritmo. Se um nó intruso já foi identificado, o segundo parâmetro da função “Detecta intruso” recebe o nó identificado como intruso. Se esse parâmetro estiver assinalado, então os indícios de intrusos são associados a esse nó.

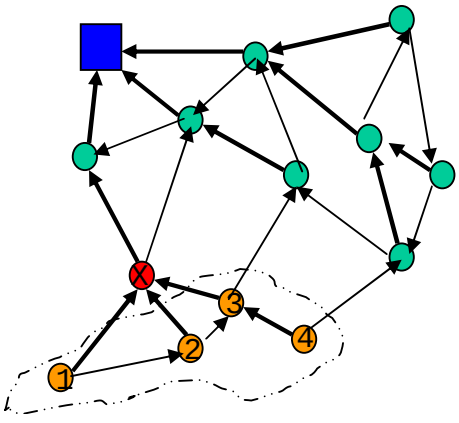

Figura 7.3 - Exemplo de detecção de intrusos

Um exemplo de detecção de intrusos pode ser visto na Figura 7.3. Nesse exemplo, o nó marcado com X realiza um ataque, não repassando as mensagens enviadas pelos nós marcados por 1, 2, 3 e 4. Os nós que dependem do nó marcado com X para seu roteamento estão circulados. O algoritmo de detecção do intruso iniciará a partir da estação base. Quando o nó marcado com X for analisado pelo algoritmo de detecção, será verificado as perdas por ambas as rotas dos nós marcados de 1 a 4. As perdas dos nós 1 e 2 ocorrerão na mesma proporção pelas duas rotas, pois ambas dependem do nó X. Não poderá ser constatada a presença do intruso pela análise desses nós. Os nós 3 e 4, porém, terão perdas muito maiores na rota padrão, que depende do nó X,

que na rota alternativa, que não depende do nó X. Na primeira parte do algoritmo 2, na análise das perdas dos vizinhos que usam o nó X como rota padrão, serão identificadas as perdas excessivas dos nós 3 e 4 e o nó X será identificado como possível intruso, tendo sua pontuação de intruso registrada em duas unidades.

7.4 Revogação de intrusos

Após a detecção de intrusos, é importante o uso de um mecanismo de revogação dos intrusos detectados para impedir sua atuação. A revogação deve ser feita por elemento confiável para que não seja usada para revogar nós autênticos. Após a revogação dos intrusos, é necessária a reconfiguração da rede, para que sejam estabelecidas novas rotas em substituição àquelas que contavam com a participação de intrusos e que deixam de funcionar pela sua revogação.

A revogação de nós deve ser iniciada por um elemento reconhecidamente confiável. Nas RSSF somente a estação base atende a esse requisito e pode solicitar a revogação. O processo deve ser totalmente autenticado. Para tanto, o protocolo μTesla [4] pode ser usado. O μTesla é um protocolo para autenticação em difusão em múltiplos passos que usa uma cadeia de chaves para validar as chaves utilizadas. O μTesla pode ser usado para enviar os comandos de revogação para todos os nós.

O processo de revogação inicia-se após a fase de detecção de intrusos. Os nós identificados como intrusos devem ser revogados. Uma ou mais mensagens podem ser enviadas autenticadas com uma chave da cadeia de chaves do μTesla. Após o anúncio da chave, os nós podem inserir os identificadores dos nós na lista de nós revogados, e descartar todas as mensagens enviadas por eles.

Após o processo de revogação, a estação base deve iniciar uma nova fase de estabelecimento de rotas para eliminar do roteamento os nós intrusos. Em seguida deve dar continuidade ao sensoriamento e coletar os dados da rede. Após determinado tempo, uma nova fase de detecção pode ser iniciada e novos nós podem ser revogados após essa fase.

7.5 Avaliação

Três aspectos foram analisados para a avaliação do presente caso: o desempenho, a funcionalidade e a escalabilidade. O desempenho é baseado no consumo de energia, uma das métricas mais importante das RSSF. A funcionalidade é avaliada em relação à redução do número de nós silenciados, bem como ao resultado da detecção de intrusos. A escalabilidade é avaliada pela execução em diferentes massas de dados com tamanhos que variam desde algumas dezenas até milhares de nós, além de dois tipos diferentes de distribuições. O ataque conhecido como Buraco Negro foi utilizado, embora outros ataques que interferem no roteamento poderiam ter sido escolhidos.

Para avaliar esses aspectos, foram realizados três conjuntos de simulações. Embora alguns simuladores permitam simular as características de RSSF, como o SensorSim [59], TOSSIM[60] e o PowerTOSSIM [61], esses simuladores não têm recursos para simular ataques. Dessa forma, o simulador apresentado em [52] inclui implementação de diversos tipos de ataque e é mais adequado a esse trabalho. Os objetivos específicos de cada conjunto de simulações podem ser assim definidos:

- Consumo de Energia: Simulação de rede com protocolo de roteamento com rotas alternativas com o objetivo de verificar o aumento do consumo de energia causado pelo uso de rotas alternativas;
- Eficácia das Rotas Alternativas: Simulação de rede com protocolo de roteamento com rotas alternativas e com a presença de intrusos. O objetivo é verificar a eficácia do uso de rotas alternativas para reduzir o número de nós silenciados por certos ataques de negação de serviço;
- Eficácia da Detecção de Intrusos: Simulação de rede com protocolo de roteamento com rotas alternativas, com a presença de intrusos e execução do algoritmo de detecção de intrusos. O objetivo é verificar a eficácia do algoritmo de detecção de intrusos;

As redes simuladas utilizam distribuições diversas com quantidades de nós variando entre 40 e 1025, distribuídas de diversas formas, desde a distribuição aleatória até a distribuição uniforme, conforme apresentado na

seção 2.7. O uso de cenários diversos possibilita verificar a escalabilidade da solução aqui apresentada.

7.5.1 Consumo de energia

A medida de desempenho mais importante em RSSF é o consumo de energia, pelo fato das fontes serem usadas de forma descartável. Qualquer mudança deve ser avaliada, então, em relação ao seu consumo de energia. Esta seção apresenta algumas simulações de RSSF típicas sem a presença de rotas alternativas e também com a presença de rotas alternativas. O objetivo é verificar qual o aumento no consumo de energia causado pelo uso dessa abordagem.

As condições de simulação incluem a geração de dados pelos nós em intervalos fixos com seu envio à estação base. A energia gasta com o processamento é uma função das tarefas realizadas pelo nó e pelo tempo de simulação. A energia gasta com transmissão é uma função do número de pacotes transmitidos. O algoritmo de estabelecimento de rotas só é executado uma única vez. Dessa forma, alguns poucos nós, responsáveis por repassar pacotes de muitos outros, têm consumo muito maior que os demais.

Tabela 7.1 - Aumento do consumo pelo uso de Rotas Alternativas

Distribuição	Número de Nós	Consumo Médio sem alternância (mJ)	Consumo Médio com alternância (mJ)	Percentual de aumento
Aleatória	1025	501,54	519,45	3,57%
Aleatória	399	346,17	360,68	4,19%
Aleatória	40	145,03	168,70	16,32%
Faixas	1024	486,28	504,78	3,80%
Faixas	399	330,96	346,46	4,68%
Hexagonal	1020	517,98	532,22	2,75%
Hexagonal	399	336,48	347,77	3,35%
Pertuba 5	1024	519,89	534,05	2,72%
Pertuba 5	399	338,90	350,29	3,36%

O resultado obtido, nesse caso, é a energia média gasta pelos nós. A Tabela 7.1 mostra o consumo médio para o uso de rotas simples e o consumo médio com o uso de rotas alternativas.

O uso de rotas alternativas representou um aumento no consumo da rede entre 2 e 16 %, sendo que esse último só foi registrado para uma vez, para uma rede muito pequena, com apenas 40 nós. Observa-se que o aumento do consumo pelo uso de rotas alternativas é maior para um número menor de nodos. Em redes com um maior número de nós, a energia adicional gasta com a rota alternativa é menos significativa, em função da maior energia gasta para roteamento do dado até a estação base.

A distribuição da energia gasta pelos nós da rede também é interessante para o nosso trabalho. Como o simulador não implementa nenhum esquema de renovação das rotas, alguns nós são sobrecarregados, resultando em consumo excessivo de energia.

7.5.2 Eficácia das rotas alternativas

O segundo grupo de simulações tem objetivo de verificar a eficácia da alternância de rotas para o aumento da resiliência à intrusão. Isso se deve ao fato que a existência de um intruso em uma das rotas de um nó pode não ser suficiente para silencia-lo, uma vez que seus dados podem ser entregues pela outra rota, mesmo que em menor freqüência.

Esta seção apresenta algumas simulações de RSSF com rotas alternativas e a presença de intrusos. Para garantir a escalabilidade da solução aqui apresentada, várias simulações foram realizadas com diferentes quantidades de nós sensores, bem como nós intrusos. As distribuições de nós são subconjunto das apresentadas na seção anterior.

O ataque escolhido para essa análise foi o ataque de Black Hole, devido à abrangência desse ataque, capaz de silenciar todos os nós que dependem do nó intruso para o roteamento dos pacotes até a estação base. Os nós intrusos foram sorteados aleatoriamente, representando 10 ou 30% do total de nós. Ataques abaixo de 10% dos nós da rede podem ter um impacto muito pequeno, e, acima de 30 %, um impacto muito grande, silenciando a maioria dos nós. O número de pacotes entregues, por cada nó, à estação base, foi registrado. Ao final, foram verificados o número de nós silenciados para as simulações das RSSF com rotas alternativas e sem o uso desse mecanismo. Os resultados podem ser vistos na Tabela 7.2.

Tabela 7.2 - Aumento da Resiliência pelas Rotas Alternativas

			Sem Alternância		Com Alternância	
Teste	Número de Nós	Percentual de Nós Intrusos	Nós silenciados	Nós com resposta reduzida	Nós silenciados	Nós com resposta reduzida
Aleatório	1024	10 %	378	0	274	221
Aleatório	1024	30 %	739	0	649	269
Aleatório	399	10 %	71	0	49	81
Aleatório	399	30 %	192	0	155	134
Aleatório	40	10 %	0	0	0	0
Aleatório	40	30 %	14	0	7	19
Faixas	1024	10 %	387	0	144	204
Faixas	1024	30 %	827	0	635	231
Faixas	399	10 %	105	0	78	117
Faixas	399	30 %	213	0	154	141

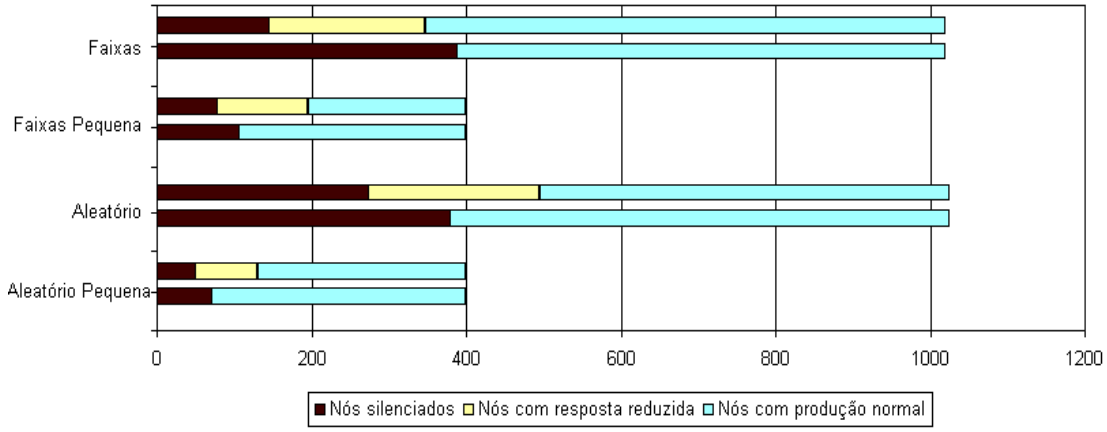

Figura 7.4 - Respostas da Rede na Presença de 10% de Intrusos para o roteamento com alternância e sem alternância

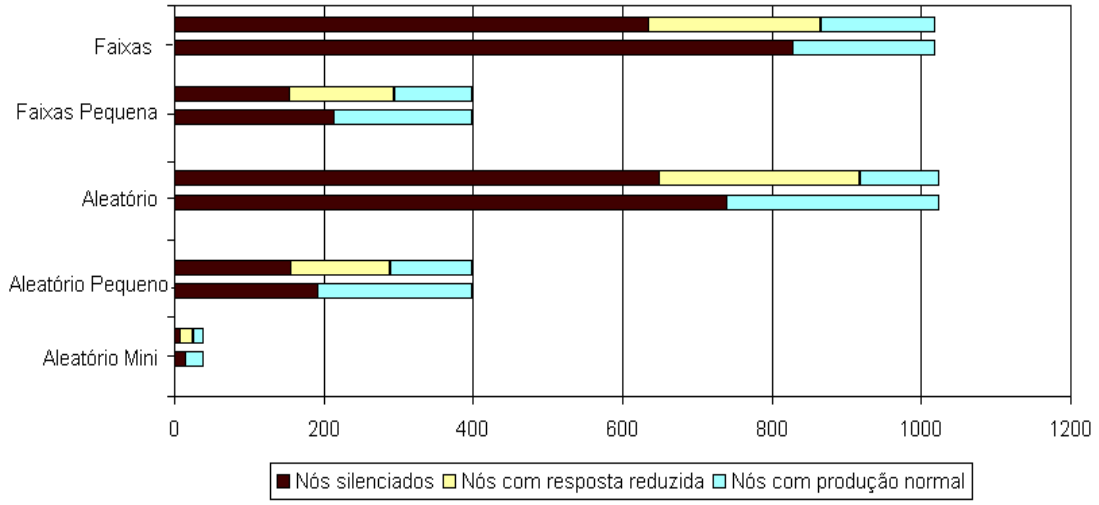

Figura 7.5 - Respostas da Rede na Presença de 30% de Intrusos para o roteamento com alternância e sem alternância

A Figura 7.4 e a Figura 7.5 mostram graficamente a diferença de produção nas diversas situações. Os resultados aqui apresentados mostram um número menor de nós silenciados pelo uso de rotas alternativas.

7.5.3 Eficácia da detecção de intrusos

O último grupo de simulações a ser apresentado neste trabalho tem o objetivo de validar o algoritmo de detecção de intrusos apresentado e mostrar sua eficácia. Para tanto, esse algoritmo foi implementado no simulador utilizado [52].

De acordo com o algoritmo apresentado na seção 7.3, a detecção de intrusos é mais eficiente para os intrusos que apresentam uma hierarquia mais alta na árvore de roteamento, pois eles recebem maior pontuação no algoritmo, uma vez que um maior número de nós que dependem dele são silenciados. Os intrusos que não tem função no algoritmo de roteamento não são detectados, pois sua presença não tem efeito sobre a rede.

Tabela 7.3 - Eficácia da detecção de intrusos para um intruso

			Ação dos intrusos		Resultados da detecção	
Teste	Número de Nós	Número de Simulações	Simulações com intruso efetivo (participa no roteamento)	Simulações com intruso inócuo (não participa do roteamento)	Simulações com Intruso detectado	Simulações com intruso não detectado
Aleatório	1024	40	16	24	16	24
Aleatório	399	40	19	21	19	21
Aleatório	40	40	20	20	20	20
Faixas	1024	40	15	25	15	25
Faixas	399	40	19	21	19	21

A detecção de intrusos tem eficácia total para detectar um pequeno número de intrusos, desde que eles não estejam presentes em rotas comuns, ou seja, interferindo na produção de nós em comum. A Tabela 7.3 apresenta os resultados para o caso de apenas um intruso presente no roteamento.

A detecção de intrusos tem sua eficácia reduzida para um número grande de intrusos. Mas ainda sim, nesse caso, o algoritmo de detecção consegue

localizar parte dos intrusos. A revogação desses intrusos, com o restabelecimento de rotas e nova execução do algoritmo de detecção pode permitir a descoberta de outros intrusos, de forma que a solução converge para a detecção total do número de intrusos. A tabela 4 apresenta os resultados para as mesmas distribuições de nós usadas nas outras simulações. As quantidades de intrusos simulados foram de 10 e 30% do total de nós. Esses números foram escolhidos porque, a partir de 10% de intrusos, a rede tem sua produção reduzida de forma significativa. Acima de 30% de intrusos, a rede apresenta uma perda muita alta da produção, inviabilizando sua recuperação. A Tabela 7.4 apresenta numericamente os resultados obtidos. A Figura 7.6 e a Figura 7.7 apresentam os resultados graficamente.

Tabela 7.4 - Detecção de Intrusos com grande número de intrusos

Teste	Número de Nós	Número de Intrusos	Número de Intrusos presentes no roteamento	Número de Intrusos detectados	Intrusos não detectados
Aleatório	1024	121	49	30	19
Aleatório	1024	355	129	29	100
Aleatório	399	49	23	16	7
Aleatório	399	137	63	24	39
Aleatório	40	5	1	1	0
Aleatório	40	17	7	4	3
Faixas	1024	99	34	25	9
Faixas	1024	302	114	28	86
Faixas	399	49	19	11	8
Faixas	399	125	60	23	37

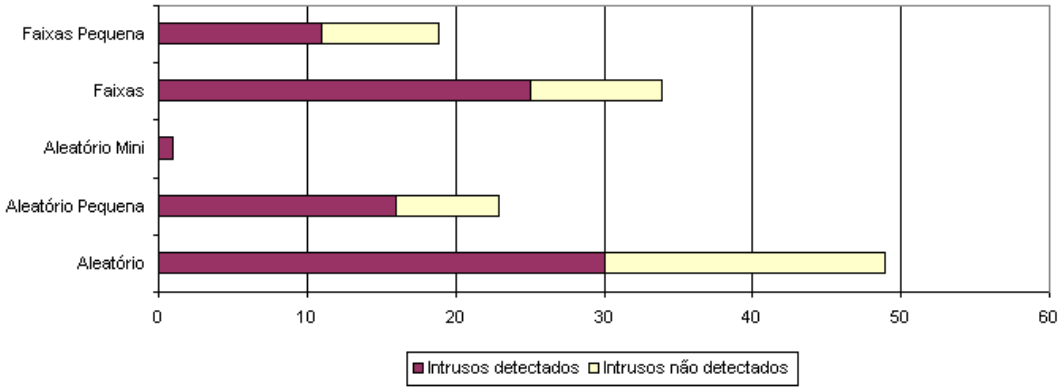

Figura 7.6 - Detecção de intrusos em redes com 10 % de intrusos

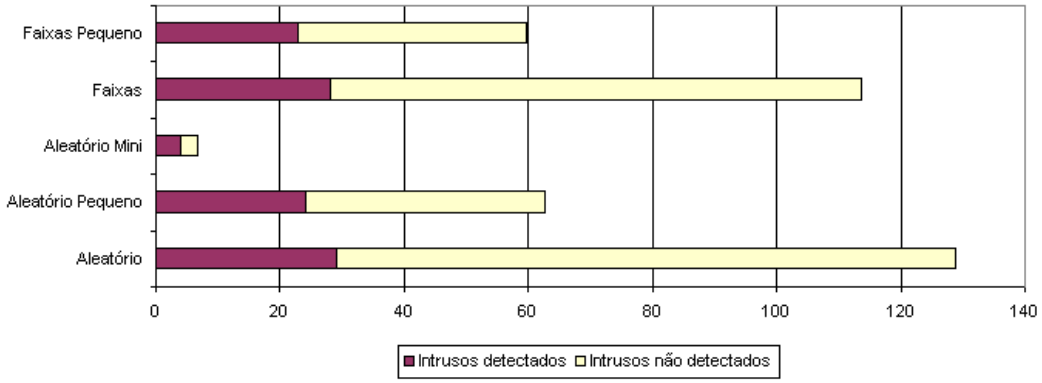

Figura 7.7 - Detecção de intrusos em redes com 30 % de intrusos

Não foi verificado nenhum caso de falso positivo. A presença de falhas intermitentes poderia levar a existência de falsos positivos, mas somente em situações muito específicas que não puderam ser obtidas nas simulações.

7.5.4 Detecção em várias iterações

Após a revogação dos intrusos detectados, e a conseqüente revogação desses intrusos, a rede pode refazer as rotas, eliminando os intrusos já revogados. A partir desse momento, a rede pode reiniciar o processo de sensoriamento, habilitando nova fase de detecção de intrusos. Com a repetição de sucessivas fases da rede, todos os intrusos efetivos, ou seja, aqueles que participam do roteamento, podem ser detectados.

Para avaliar o número de intrusos detectado nas sucessivas iterações da rede, foram realizadas várias simulações, com o objetivo de verificar quantas iterações são necessárias para eliminar 100% dos intrusos em vários tipos de redes.

Tabela 7.5 - Intrusos detectados após várias iterações da rede

	Número de nós	Total de intrusos	Iterações até a detecção total	Intrusos detectados	Intrusos sem efeito
A	1025	121	6	81	40
B	1025	355	17	293	62
C	399	49	4	30	19
D	399	137	13	110	27
E	40	5	2	1	4
F	40	17	7	14	3

Seis simulações foram realizadas, nomeadas de A a F, onde a distribuição de nós foi aleatória e o número de nós e intrusos variam conforme a tabela. Os resultados são apresentados na Tabela 7.5.

Os resultados também podem ser vistos graficamente na Figura 7.8.

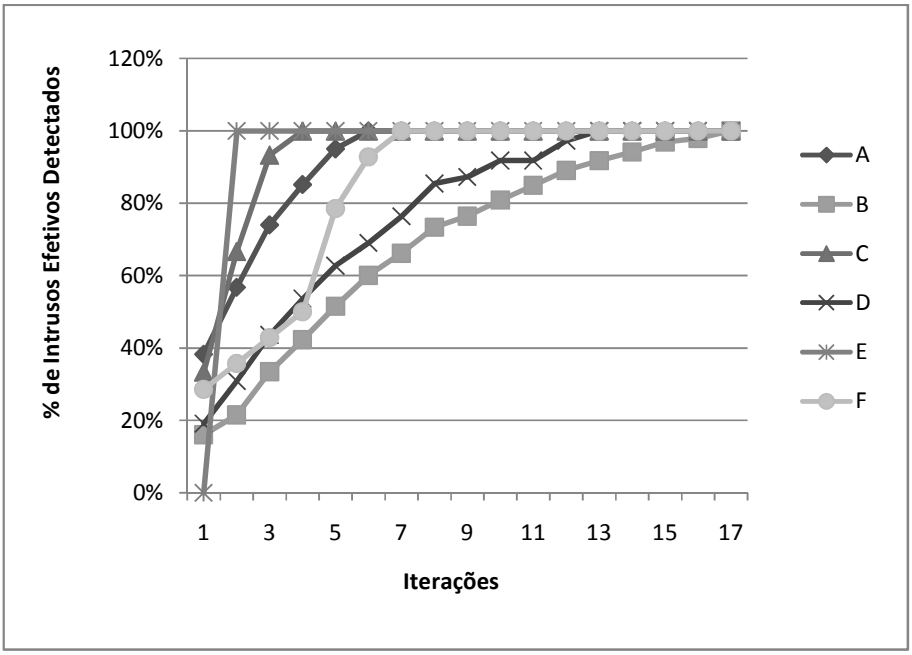

Figura 7.8 - Intrusos detectados após cada iteração

É possível estimar o tempo total para detectar todos os intrusos. Na simulação mais longa, 17 iterações foram realizadas. Considerando a geração de dados dos nós a cada 5 s, por exemplo, e a geração de 20 dados antes do início da fase de detecção de intrusos, serão necessários 100 segundos por iteração e 1700 segundos para as 17 iterações, ou seja, cerca de 28 minutos para detecção total dos intrusos efetivos.

7.6 Conclusões

Este capítulo apresentou a proposta de uso de rotas alternativas para o roteamento em RSSF, com o objetivo de aumentar a resiliência da rede à presença de intrusos, bem como permitir uma detecção eficiente de intrusos. Essa abordagem é necessária em redes onde a presença do inimigo pode significar a inserção de muitos nós intrusos com o objetivo de eliminar as funcionalidades da rede.

As restrições existentes em RSSF, especialmente de consumo de energia, foram consideradas no desenvolvimento desse trabalho. Resultados de simulações mostram que o aumento no consumo de energia é pequeno e pode ser assumido em redes onde a segurança é fator crítico.

A eficácia no aumento da resiliência também foi comprovada. Essa afirmação é possível considerando que é mais interessante para a rede aumentar a área com produção ativa do que a produção total. Os resultados das simulações mostram que o uso de rotas alternativas possibilita aumentar o número de nós em produção, aumentando assim a área monitorada, mesmo que a produção total da rede seja mantida, pois é maior o número de nós que tem a produção reduzida sob o efeito de ataques.

Capítulo 8

Validação da Arquitetura de Gerenciamento de Segurança para Redes de Sensores

8.1 Introdução

A arquitetura de gerenciamento de segurança proposta no Capítulo 5 foi consolidada com a apresentação dos mecanismos de segurança do Capítulo 6, o protocolo NEKAP, e do Capítulo 7, a proposta de alternância de rotas no encaminhamento de pacotes para a estação base. Neste capítulo, essa arquitetura será avaliada, através da avaliação de todos os componentes individualmente e agrupados nos níveis propostos, de forma a mostrar como o consumo dos mecanismos de segurança podem interferir no consumo total da rede.

Para validar a arquitetura de gerenciamento aqui apresentada, um conjunto de simulações foi realizado para verificar o consumo de energia dos diversos componentes de segurança no modelo de rede desta tese.

A metodologia da avaliação inclui a simulação de uma rede sem nenhum componente de segurança habilitado e, em seguida, a simulação da rede incluindo cada componente de segurança para avaliar seu consumo em separado. Foi utilizado um simulador baseado em eventos discretos, desenvolvido no DCC-UFMG [52]. O objetivo é mostrar o consumo de energia de cada componente de segurança e de cada nível, para justificar a manutenção dos níveis inferiores sempre que a presença de intrusos não tiver sido constatada.

Para tanto, a simulação usou uma rede plana e homogênea, com o número total de nós variando entre 50 e 1000 nós. Os nós são distribuídos pela rede de forma uniforme, no modelo conhecido como colméia, onde cada nó tem exatamente seis vizinhos eqüidistantes, a uma distância de 34 metros. Cada simulação opera 1890 segundos, cerca de 30 minutos. Neste período, 135 operações de sensoriamento são realizadas por nó.

8.1.1 Consumo sem componentes de segurança

O primeiro conjunto de simulação tem por objetivo avaliar o consumo da rede no nível inicial, sem qualquer mecanismo de segurança habilitado, exceto uma possível detecção de intrusos na estação base, que não sobrecarrega a rede.

A Tabela 8.1 apresenta os resultados para o caso base. Além de não ter nenhum componente de segurança habilitado, a rede utilizava de fusão de dados no primeiro salto do roteamento. Assim, todo nó que repassa dados coletados só repassa dos dados que estão no segundo salto no caminho para a estação base. Os dados recebidos em primeiro salto são os nós vizinhos do nó repassador e terão seus dados sumarizados em fusão de dados e apenas o resultado é repassado.

Tabela 8.1 - Consumo de energia na rede sem componetes segurança

Número de nós	**Pacotes transmitidos**	**Pacotes recebidos**	**Pacotes transmitidos para atualização de rotas**	**Pacotes recebidos para atualização de rota**	**Energia media gasta por nó (mAh x 10^{-3})**
50	16229	112170	150	909	110,27
100	42379	298276	300	1893	146,33
200	112592	798710	600	3897	195,75
400	308612	2186616	1200	7953	267,99
600	568560	4022888	1800	12045	328,74
800	817687	5802845	2400	16161	355,53
1000	1162417	8238595	3000	20277	403,87

8.1.2 Criptografia

TinySec [6] é a escolha de criptografia deste trabalho. Os processos criptográficos devem ser avaliados em termos de sobrecarga de processamento e de rede. Entretanto, como o tempo efetivo de rede é normalmente maior que o tempo efetivo de processamento e os nós precisam manter o estado ocioso durante as operações de rede, é possível considerar que a sobrecarga de processamento ocorre durante o tempo ocioso do processador e pode ser descartada. Assim, apenas a sobrecarga de rede será considerada.

TinySec tem dois modos de funcionamento: TinySec-Auth e TinySec-AE. O primeiro incrementa um byte no pacote de rede, de 33 para 34 bytes, incrementando o consumo de energia da rede em 3%. O segundo incrementa 5 bytes, de 33 para 38 bytes, incrementando o consumo de energia da rede em 15%.

8.1.3 Gerenciamento de chaves

Os protocolos de gerenciamento de chaves escolhidos, SPINS e NEKAP, estabelecem suas chaves durante a fase de configuração da rede. O SPINS inclui dois protocolos, SNEP e µTesla. O SNEP não gasta energia para se estabelecer, pois utiliza chaves pré-distribuídas. O µTesla divulga periodicamente uma chave que pertence a uma cadeia de chaves de via única. Avaliar µTesla corresponde a avaliar a divulgação dessa chave.

Para avaliar o µTesla é necessário considerar mensagens enviadas em modo de difusão. No modelo adotado nesta tese, a estação base somente envia o *beacon* de atualização de rotas em modo de difusão. Como a mensagem de atualização de rotas é a mais importante a autenticar e é enviada periodicamente, sua ocorrência será considerada na mesma freqüência do que a divulgação das chaves do µTesla.

8.1.4 Sistemas de detecção de intrusos e revogação de nós

Sistemas de detecção de intrusos podem executar na estação base ou em nós especiais, chamados monitores [8]. Na estação base, têm a vantagem de não consumir os recursos limitados da rede e podem solicitar a revogação dos nós, uma vez que são os únicos pontos confiáveis da rede. Nos nós monitores, tem a vantagem de estarem próximos aos intrusos, o que pode facilitar a detecção, uma vez que eles podem escutar a vizinhança local e identificar a ação de nós suspeitos.

Para a avaliação de IDS, vamos considerar apenas um incremento no tempo de recepção dos nós monitores. De acordo com o modelo adotado neste trabalho, os nós permanecem ativos apenas durante o tempo necessário para enviar e receber mensagens relativas ao sensoriamento. Os nós monitores

teriam de ficar mais tempo ativos para perceberem o comportamento dos outros nós e verificar se eles estão repassando as mensagens devidamente.

Como no modelo proposto são considerados seis vizinhos por nó, com apenas um deles com funções de roteamento, o nó monitor teria de escutar, além de suas próprias mensagens, as mensagens enviadas pelo nó repassador. Como parte dessas mensagens já é normalmente recebida pelos nós monitores, o acréscimo no tempo de recepção será considerado como 1/6 do tempo, ou 16,7%.

Nos níveis de segurança propostos, o número de nós monitores varia entre 10 e 30%. Para avaliar o custo do IDS descentralizado, nos nós monitores, será considerado um incremento de 16,7% no número de mensagens recebidas para o percentual de nós monitores. A Tabela 8.2 mostra o resultado da energia extra consumida quando 10% dos nós estão executando a detecção de intrusos. Para 30%, o incremento é três vezes superior.

Tabela 8.2 - Energia extra consumida com IDS em 10% dos nós

Número de nós	Pacotes transmitidos	Pacotes recebidos	Average extra spent energy per node with IDS (mAh x 10^{-3})
50	16229	112170	1.84
100	42379	298276	2.44
200	112592	798710	3.27
400	308612	2186616	4.48
600	568560	4022888	5.49
800	817687	5802845	5.94
1000	1162417	8238595	6.74

8.1.5 Roteamento seguro

Rotas alternativas foram escolhidas para incrementar a segurança do roteamento neste trabalho. Para avaliar rotas alternativas, foram realizadas simulações usando o modelo apresentado neste trabalho.

Outra possível solução para incrementar a segurança do roteamento é autenticar o *beacon* de atualização das rotas. Essa opção é muito barata, em relação às demais, pois menos de 1% dos pacotes considerados são de atualização de rotas. Autenticá-los incrementa 3% do seu custo, ou seja, 0,03% do total, algo desprezível para a precisão desta avaliação.

Tabela 8.3 - Energia consumida com rotas alternativas

			Enegia media gasta por nó		
Número de nós	**Total de pacotes transmitidos**	**Total de pacotes recebidos**	**Sem rotas alternativas (mAh x 10^{-3})**	**Com rotas alternativas (mAh x 10^{-3})**	**Aumento**
50	17419	119932	110.27	117.95	6.96%
100	45083	317323	146.33	155.67	6.38%
200	117236	831173	195.75	203.72	4.07%
400	328676	2328338	267.99	283.36	5.74%
600	581786	4119388	328.74	348.61	6.04%
800	876762	6217556	355.53	380.97	7.16%
1000	1224154	8678537	403.87	425.42	5.34%

8.1.6 Fusão de dados

Ao contrário dos outros componentes avaliados até aqui, fusão de dados reduz o consumo de energia da rede. Entretanto, a fusão de dados deve ser desligada nos níveis de segurança superiores, devido às vulnerabilidades do processo.

Nessa avaliação não são consideradas as rotas alternativas, pois quando afusão de dados está ativa, as rotas alternativas não o estão. A Tabela 8.4 apresenta os resultados. Fusão de dados economiza até 40% do consumo da rede.

Tabela 8.4 - Aumento do consumo de energia sem fusão de dados

			Energia media consumida por nó		
Número de nós	**Total de pacotes transmitidos**	**Total de pacotes recebidos**	**Sem rotas alternativas (mAh x 10^{-3})**	**Com rotas alternativas (mAh x 10^{-3})**	**Aumento**
50	17419	119932	117,95	146,66	24,34%
100	45083	317323	155,67	200,95	29,09%
200	117236	831173	203,72	265,88	30,51%
400	328676	2328338	283,36	388,11	36,97%
600	581786	4119388	348,61	472,68	35,59%
800	876762	6217556	380,97	526,62	38,23%
1000	1224154	8678537	425,42	595,82	40,05%

8.1.7 Gerenciamento de segurança

Dois tipos de mensagens de gerenciamento são enviados pela rede neste trabalho. Mensagens enviadas em modo de difusão para configurar componentes de segurança e mensagens de detecção de intrusos, enviados pelos nós para a estação base.

Mensagens enviadas pela estação base são repassadas por todos os nós em inundação (*flooding)*. O custo total é correspondente a uma mensagem por

nó da rede. Esse é o custo para informar à rede que o nível de segurança aumentou ou diminuiu.

Mensagens de detecção de intrusos são enviadas pelos nós monitores para a estação base e o custo depende da distância do nó até a estação base. Uma referência para esse custo é o custo de uma mensagem de sensoriamento. Ambas percorrem o mesmo caminho até a estação base.

As mensagens de gerenciamento só trafegam pela rede na presença de intrusos. Essa característica garante que não será um custo desnecessário em hipótese alguma. Essas mensagens sempre serão justificadas, pois poderão deixar a rede em situação de alerta.

O custo das mensagens de gerenciamento é muito baixo, pois são próximos aos custos das mensagens de atualização de rotas, perto de 1% e de um único sensoriamento de um dado, que ocorrem 135 vezes por cada nó para a simulação que opera cerca de 30 minutos.

8.2 Resultados finais

Para encerrar essa seção de avaliação, os níveis de segurança foram simulados em conjunto, para mostrar a diferença efetiva de mudança de nível no consumo de energia da rede. O aumento do consumo de energia foi considerado para criptografia, fusão de dados, rotas alternativas e detecção de intrusos.

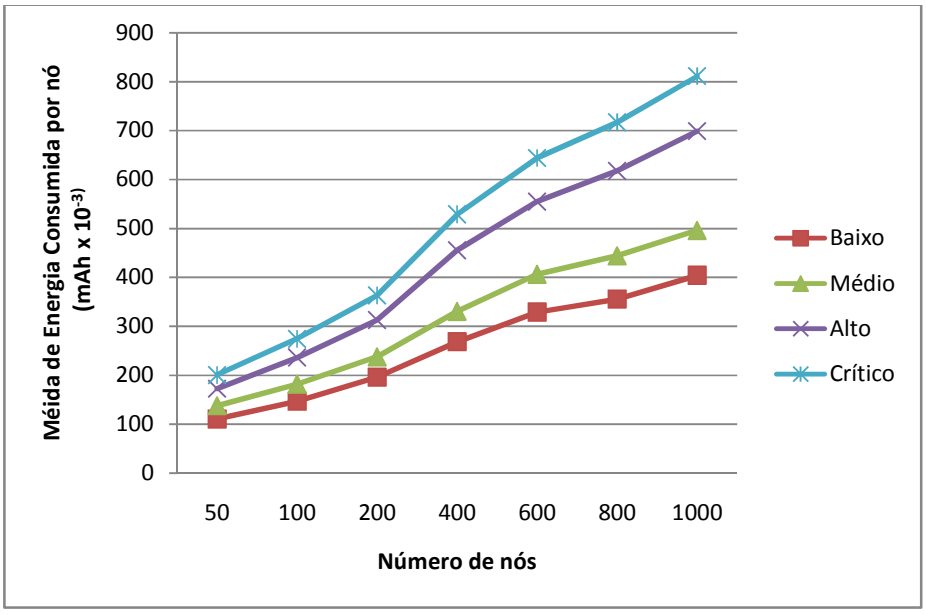

Figura 8.1 - Consumo de energia nos diferentes níveis de segurança

Tabela 8.5 - Consumo de energia nos diferentes níveis de segurança

Número de Nós	Consumo de energia por nó nos níveis (mAh x 10^{-3})			
	Baixo	Médio	Alto	Crítico
50	110,27	137,48	172,33	200,53
100	146,33	181,46	235,97	274,34
200	195,75	237,54	312,30	363,20
400	267,99	330,34	455,28	528,63
600	328,74	406,39	554,56	644,00
800	355,53	444,05	617,49	716,56
1000	403,87	495,97	698,67	810,82

A Tabela 8.5 apresenta os dados finais da simulação, plotados na Figura 8.1. As diferenças entre os diversos níveis podem chegar a um aumento de 100% do nível baixo para o nível crítico. Isso representaria uma queda de 50% no tempo de vida da rede, caso todos os componentes de segurança fossem usados durante todo o funcionamento da rede.

8.3 Discussão

Um requisito comum da maior parte das soluções propostas para redes de sensores é aumentar a disponibilidade do serviço oferecido pela rede. Para tanto, o prolongamento do tempo de vida da rede pelo baixo consumo de bateria tem sido o alvo principal dos trabalhos. Porém, problemas de segurança podem ser encontrados nesse ambiente, especialmente pelos ataques de negação que serviço que podem reduzir os serviços da rede antes do término da capacidade das baterias.

Diversas soluções de segurança podem ser utilizadas para bloquear ataques de negação de serviço e aumentar a disponibilidade da rede. Cada solução aumenta o consumo de energia de 10 a 20%. Se essas soluções são sempre utilizadas, o tempo de vida da rede diminuirá por causa da exaustão das baterias.

Uma arquitetura de gerenciamento de segurança, como a apresentada neste trabalho, possibilita equilibrar a disponibilidade da rede e o consumo de energia, ligando e desligando as soluções de segurança quando necessário. Entretanto, o gerenciamento também tem um custo de energia.

Considerando o modelo apresentado, é possível avaliar três cenários distintos:

1 - Rede sem segurança: Nesse caso, a disponibilidade da rede pode ser comprometida pela presença de intrusos, o que pode reduzir a produtividade e o tempo de vida da rede;

2 - Rede com uso constante de algumas soluções de segurança: Nesse caso, a presença de intrusos é evitada ou reduzida, o que pode aumentar a disponibilidade da rede, mas o consumo de energia aumenta para executar essas soluções de segurança. Com algumas soluções de segurança, o consumo de energia pode aumentar em 40%, conforme a simulação, e reduzir o tempo de vida pelo término das baterias;

3 - Rede com gerenciamento de segurança para ativar soluções de segurança somente quando necessário. Se nenhum intruso é detectado, a rede pode trabalhar sem componentes de segurança, e manter o consumo de energia mínimo para prolongar o tempo de vida da rede. Quando a rede detecta um intruso, a solução de gerenciamento aumenta o nível de segurança para evitar o efeito do intruso.

O consumo adicional de energia no terceiro cenário com mecanismos de segurança dependerá da presença de intrusos. No melhor caso, somente sistemas centralizados de detecção de intrusos executam, sem execução nos nós. Quando um primeiro intruso é detectado, o gerenciamento de segurança começa a ativar soluções de segurança pelo envio de mensagens. As soluções de segurança serão ligadas gradualmente, aumentando o consumo da rede, mas evitando o efeito dos intrusos. Em grandes redes, a rede pode ser dividida em setores e as soluções de segurança podem ser ativadas somente onde intrusos são detectados.

8.4 Conclusões

Este capítulo apresenta uma arquitetura de gerenciamento de segurança para RSSF. O objetivo é estender a disponibilidade da rede e o tempo de vida pela configuração de soluções de segurança e sua ativação somente quando for necessário.

A arquitetura de gerenciamento de segurança é uma extensão do Manna[49], um modelo de gerenciamento para RSSF e herda dele o protocolo, formato das mensagens e estende sua MIB.

De maneira autonômica, um gerente na estação base pode configurar níveis de segurança nos nós e ativar componentes de segurança para evitar o efeito dos intrusos. Um evento de detecção de intrusos gera decisão autonômica alterando os níveis de segurança.

É possível propor o uso de políticas para flexibilizar a configuração das ferramentas de segurança. As políticas podem ser escritas de acordo com as necessidades de cada aplicação, indicando os eventos que devem ser responsáveis pela configuração de cada componente de segurança. Fica aqui como sugestão para trabalhos futuros.

Capítulo 9

Conclusões

Este trabalho apresentou a tese de que é possível proteger as funções de roteamento das redes de sensores sem fio sem que o custo associado inviabilize seu uso nesse tipo de ambiente de grandes restrições de energia e processamento.

As contribuições apresentadas neste trabalho, especialmente o protocolo de distribuição de chaves e o uso de rotas alternativas, em conjunto com outras abordagens existentes na literatura, conseguem anular o efeito de diversos tipos de ataques de negação de serviço que atuam no roteamento. Assim, esses mecanismos aumentam a disponibilidade da rede. A arquitetura de gerenciamento de segurança permite racionalizar o uso dos mecanismos de segurança propostos nesta tese, de forma a estender o tempo de vida da rede pela economia de energia e aplicação dos mecanismos de segurança assim que intrusos forem detectados.

O protocolo de distribuição de chaves chamado de NEKAP foi apresentado no quinto capítulo. Esse protocolo possibilita o estabelecimento de chaves entre quaisquer dois nós vizinhos na rede, com custo relativamente baixo. As chaves podem ser usadas para assinar todas as mensagens trocadas durante o estabelecimento das rotas de forma a impedir a inserção de nós intrusos externos no roteamento.

Embora o uso de uma abordagem de controle de acesso utilizando o NEKAP possibilite a eliminação dos ataques externos, ainda é possível a existência de um nó intruso interno, resultado da adulteração física (*tampering*) de um nó autêntico. Além disso, algumas aplicações podem não possibilitar o controle de acesso proposto. Por essa razão, outra abordagem também foi proposta. O Capítulo 7 apresentou uma abordagem para uso de rotas alternativas no roteamento para aumentar a resiliência da rede à presença de intrusos e ainda permitir a detecção eficiente dos intrusos. Assim, até os ataques internos podem ser descobertos e eliminados.

O uso das duas abordagens em conjunto permite a proteção do roteamento com custo relativamente baixo até mesmo para essas redes. Essas abordagens podem, ainda, ser usadas separadamente ou ocasionalmente, acionadas por um mecanismo de gerenciamento de redes.

Uma abordagem de gerenciamento de segurança, como apresentado no sétimo capítulo, permite a configuração dos mecanismos de segurança, permitindo seu uso quando necessário, evitando, assim, o desperdício de energia. O arcabouço proposto neste trabalho permite utilizar não somente as soluções aqui descritas, mas também outras, presentes na literatura, de forma racional. Esta última parte do trabalho permite manter o consumo de energia em parâmetros aceitáveis enquanto a rede não for alvo de ataques. As redes de sensores podem permanecer nessa situação durante todo o período de vida de seus nós, de forma que nenhum mecanismo de segurança venha a sobrecarregar indevidamente seu funcionamento.

Os objetivos desta tese foram atingidos com sucesso. O protocolo de gerenciamento de chaves proposto é capaz de possibilitar o controle de acesso no enlace, e o algoritmo de roteamento com rotas alternativas possibilita a detecção de intrusão, mesmo que seja efetuada de forma distribuída e em grande parte da rede. Por fim, para conter o consumo de energia a níveis plausíveis para esse tipo de rede foi proposta a arquitetura de gerenciamento de segurança, que atende bem a essa finalidade.

Contar com mecanismos de segurança é sempre útil para qualquer tipo de aplicação. No cenário de RSSF isso se torna mais evidente. Certamente as restrições de energia poderiam inviabilizar essa proteção. Entretanto, as soluções apresentadas nesta tese garantem que o consumo de energia adicional das ferramentas de segurança é suportável e pode ser despendido apenas quando a presença de intrusos for observada.

9.1 Trabalhos futuros

Esta tese abre horizontes para o desenvolvimento de vários trabalhos seguindo os mesmos objetivos: proteger a rede contra ataques de negação de serviço e manter o consumo de energia em níveis plausíveis com o tipo de rede e duração da aplicação.

O protocolo de estabelecimento de chaves apresentado nesta tese pode ser melhorado para redes de sensores sem fio que suporta mobilidade. Nesse cenário, seu uso ainda é muito restrito. Podem, ainda, ser estendido para redes com outras características, como redes hierárquicas, ou redes heterogêneas.

O uso de nós com maior poder de processamento e maior capacidade energética pode, ainda, possibilitar a execução de algoritmos de chave pública em alguns nós, podendo usar esse recurso para melhorar a distribuição e renovação de chaves.

A alternância no roteamento apresentada nesta tese pode, ainda, ser implementada em outros protocolos de roteamento propostos na literatura. O mesmo pode ser feito a outros modelos de rede, onde a mobilidade e hierarquia estejam presentes.

A arquitetura de gerenciamento de segurança também pode ser aprimorada. Um primeiro trabalho pode ser a inclusão de uma ferramenta de definição de políticas, que permita especificar as condições e eventos que devem ocorrer para que os componentes de segurança sejam configurados e parametrizados de acordo com cada aplicação, tornando a arquitetura mais flexível. Outro ponto a ser trabalhado pode ser a inclusão de outros componentes de segurança, específicos para o mesmo modelo de rede tratado nesta tese e também outros que podem ser aplicáveis a outros modelos de rede, como hierárquica e heterogênea.

Referências Bibliográficas

[1] FOUNDATION, N. S. – *Report of the National Science Foundation Workshop on Fundamental Research in Networking* – Abril de 2003, Airlie House, http://www.cs.virginia.edu/~jorg/workshop1 - Acessado em janeiro de 2007

[2] RUIZ, L. B.; CORREIA, L. H. A.; VIEIRA, L. F. M.; MACEDO, D. F.; NAKAMURA, E. F.; FIGUEIREDO, C. M. S.; VIEIRA, M.A. M.; MAIA, E. H. B.; CAMARA, D.; LOUREIRO, A. A. F.; NOGUEIRA, J. M. S.; DA SILVA JR, D. C.; AND FERNANDES, A. O. – Arquiteturas para Redes de Sensores Sem Fio – *Anais do 22º Simpósio Brasileiro de Redes de Computadores*, pages 167 – 218, Porto Alegre – RS – Brazil, 2004

[3] WOOD, A. D.; STANKOVIC, J. A. – Denial of Service in Sensor Networks – *IEEE Computer*, October 2002

[4] PERRIG, A.; SZEWCZYK, R.; TYGAR, J. D.; WEN, V.; CULLER, D. E.– SPINS: Security Protocols for Sensor Networks – *Wireless Networks 8*, 2002, Kluwer Academic Publishers, Netherlands

[5] KARLOF, Chris and WAGNER, David - Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures - *First IEEE International Workshop on Sensor Network Protocols and Applications*, Anchorage, Alaska, May 2003.

[6] KARLOF, C.; SASTRY, N.; SHANKAR, U.; WAGNER, D. - "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks", *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems* (SenSys 2004). November 2004.

[7] ZHU, S.; SETIA, S; JAJODIA, S.; - LEAP: efficient security mechanisms for large-scale distributed sensor networks – *Proceedings of 10th ACM Conference on Computer and Communications Security*, Washington D.C., October, 2003

[8] SILVA, A. P. R.; LOUREIRO, A. A. F.; MARTINS, M. H. T.; RUIZ, L. B.; ROCHA, B. P. S., WONG, H. C. (2005) - Decentralized intrusion detection in wireless sensor networks - *Proceedings of the First ACM*

International Workshop on Quality of Service & Security in Wireless and Mobile Networks - October 13, 2005, Montreal, Quebec, Canada Q2SWinet 2005

[9] TEIXEIRA, F. A. (2005) - *Detecção de Intrusos por Observação em Redes de Sensores Sem Fio* – Dissertação defendida junto ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Minas Gerais, novembro de 2005

[10] RUIZ, L. B. - *MANNÁ: Uma Arquitetura para Gerenciamento de Redes de Sensores Sem Fio* – Tese de doutorado defendida no Programa de Pós-Graduação em Ciência da Computação, UFMG, 2004

[11] ATMEL CORPORATION - *Microcontroler ATMega128* – disponível em http://www.atmel.com/dyn/products/product_card.asp?part_id=2019, acessado em outubro de 2008

[12] ATMEL CORPORATION - *Microcontroler ATMega8535* - disponível em http://www.atmel.com/dyn/products/product_card.asp?part_id=2018, acessado em outubro de 2008

[13] TEXAS INSTRUMENTS – *Microcontroler MSP430F149* http://focus.ti.com/docs/prod/productfolder.jhtml?genericPartNumber=MSP430F149, acessado em 20/08/2003

[14] HILL, J.; SZEWCZYK, R.; WOO, A., HOLLAR, S.; CULLER, D. AND PISTER, K.. System Architecture Directions For Networked Sensors. - *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, Cambridge*, USA, November 2000.

[15] PISTER, K. S. J.; KAHN, J. M., AND BOSER, B. E. - Smart dust: Wireless networks of millimeter-scale sensor nodes - *Electronics Research Laboratory Research Summary*, University of California, Berkeley, USA (1998)

[16] EYES Project – http://eyes.eu.org - acessado em 07 de setembro de 2003

[17] TMOTE SKY – *Ultra low power IEEE 802.15.4 complient wireless sensor module* – Datasheet - Disponível em

http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf , acessado em 04 de abril de 2007

[18] TEXAS INSTRUMENTS – *CC1000, Single Chip Very Low Power RF Transceiver* - disponível em http://focus.ti.com/lit/ds/symlink/cc1000.pdf, acessado em outubro de 2008, Oslo, Noruega

[19] MALAFAYA, H.; TOMÁS L.; SOUSA, J. P. - *Sensorização sem fios sobre ZigBee e IEEE 802.15.4 – III Jornada de Engenharia de Electrónica e Telecomunicações e de Computadores*, novembro de 2005, Lisboa, Portugal

[20] BLUETOOTH SIG. 2002. *Specification of the Bluetooth System*. Specification Version 1.1 (Feb., 2002). Bluetooth SIG.

[21] ELSON, J. AND ESTRIN, D - *An Address-Free Architecture For Dynamic Sensor Networks*, Technical. Report 00-724, Computer Science Department USC, January 2000.

[22] MACEDO, D. F.; CORREIA, L. H. A.; SANTOS, A. L.; LOUREIRO, A. A. F.; NOGUEIRA, J. M. S. - A Pro-Active Routing Protocol for Continuous Data Dissemination in Wireless Sensor Networks - *Proceedings of the 10th IEEE Symposium on Computers and Communications* (ISCC 2005), Cartagena, Spain

[23] DENG, J.; HAN R.; MISHRA, S. - INSENS: Intrusion-Tolerant Routing in Wireless Sensor Networks – *Proceedings of 23rd IEEE International Conference on Distributed Computing Systems* (ICDCS 2003), Providence, RI (May 2003).

[24] HU, L.; EVANS, D.; Secure aggregation for wireless networks, *Procedings of Workshop on Security and Assurance in ad Hoc Networks*, Orlando, Flórida, January, 2003

[25] GUAJARDO, J.; BLÜMEL, R.; KRIEGER, U. AND PAAR, C. - *Efficient Implementation of Elliptic Curve Cryptosystems on the TI MSP430x33x Family of Microcontrollers*, disponível em http://www.crypto.ruhr-uni-bochum.de/Publikationen/texte/guajardopkc2001_msp430.pdf, acessado em 07 de setembro de 2003

[26] CARMAN, D. W.; KRUUS, P. S.; MATT, B. J. - *Constraints And Approaches For Distributed Sensor Network Security* – NAI Labs Technical Report - September 1, 2000

[27] LIU, D.; NING, P.; LI, R. - Establishing Pairwise Keys in Distribucted Sensor Networks - *ACM Transactions on Information and System Security*, Vol. 8, No. 1, February 2005

[28] CHAN, H.; PERRIG, A; SONG, D. – Random Key Predistribution Schemes for Sensor Networks, *Procedings of IEEE Symposium on Security and Privacy* - May 11 - 14, 2003 Berkeley, CA, p. 197

[29] ESCHENAUER, L.; GLICOR, V. D. – A Key-Management Scheme for Distributed Sensor Network – *Proceedings of the 9th ACM conference on Computer and Communication Security*, Washington, DC, USA, November 2002

[30] OLIVEIRA, S.; WONG, H. C.; NOGUEIRA, J. M. – "NEKAP: Intruder Resilient and Energy Efficient Key Establishment in Sensor Networks" – *Proceedings of IEEE International Conference On Computer Communications And Networks, ICCCN'07 Workshop on Advanced Networking and Communications* – Honolulu, Hawai, 2007

[31] FREILING, F; KRONTIRIS, I.; DIMITRIOU, T. – Towards Intrusion Detection in Wireless Sensor Networks - *13th European Wireless Conference*, Paris, France, 2007.

[32] OLIVEIRA, S.; WONG, H. C.; NOGUEIRA, J. M.; PAULA, W. P. – Alternate Routes for Detection and Increase of Resilience to the Distributed Intrusion in WSN – *Proceedings of IFIP NETWORKING 2006 Workshop on Security and Privacy in Mobile and Wireless Networking* (SecPri_MobiWi 2006), Coimbra, Portugal

[33] LUO, H.; LUO, J.; LIU, Y. AND DAS, S. - Adaptive Data Fusion for Energy Efficient Routing in Wireless Sensor Networks - *IEEE Transactions on Computers*, vol. 55, no. 10, Oct. 2006.

[34] SAVOLA, R; UUSITALO, I. – Towards Node-Level Security Management in Self-Organizing Mobile Ad Hoc Networks – *Proceedings of International Conference on Internet and Web Applications and*

Services/Advanced International Conference on Telecommunications, 2006. AICT-ICIW - Volume , Issue , 19-25 Feb. 2006, Guadeloupe, French Caribbean

[35] CROSSBOW TECHNOLOGY INC - *Mica 2 Wireless Measurement System* - disponível em http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0042-04_B_MICA2.pdf, acessado em outubro de 2008, San Jose, CA, USA, February 2004

[36] CROSSBOW TECHNOLOGY INC - *Power Management - Expected battery life vs system current usage and duty cycle* – Disponível em: http://www.xbow.com/Support/Support_pdf_files/PowerManagement.xls, acessado em outubro de 2008

[37] MALAN, D. J.; WELSH, M. AND SMITH, M. D. - A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography – *Proceedings of First IEEE International Conference on Sensor and Ad Hoc Communications and Networks*, Santa Clara, California, October 2004

[38] GANESAN, P., VENUGOPALAN, R.; PEDDABACHAGARI, P.; DEAN, A.; MUELLER, F.; SICHITIU, M. - Analyzing and modeling encryption overhead for sensor network nodes - *Proceedings of the 2nd ACM International Conference on Wireless sensor networks and applications* - San Diego, CA, USA, 2003

[39] YUAN, L.; QU, G. - Design Space Exploration for Energy-Efficient Secure Sensor Network. *Proceedings of IEEE International Conference on Application-Specific Systems, Architectures, and Processors* (ASAP'02) July 17 - 19, 2002, San Jose, California

[40] LAW, Y. W.; CORIN, R; ETALLE, S.; HARTEL, P. H. - A Formally Verified Decentralized Key Management Architecture for Wireless Sensor Networks. *Procedings of Personal Wireless Communications, PWC*, Venice, Italy, 2003

[41] R. J. WATRO, D. KONG, S. FEN CUTI, C. GARDINER, C. LYNN, AND P. KRUUS, TinyPK: securing sensor networks with public key

technology. in *2nd ACM Workshop on Security of ad hoc and Sensor Networks* (SASN'04), 2004, pp. 59–64, Washington, DC, USA

[42] OLIVEIRA, Leonardo B. ; SCOTT, M. ; LOPEZ, J. ; DAHAB, R. . TinyPBC: Pairings for Authenticated Identity-Based Non-Interactive Key Distribution in Sensor Networks. *Proceedings of 5th International Conference on Networked Sensing Systems* (INSS'08), 2008, Kanazawa, Japan, 2008

[43] HU, Y.; PERRIG, A.; AND JOHNSON, D. B. – *Wormhole detection in wireless ad hoc networks* - Technical Report TR01-384, Rice University Departament of Computer Science, June 2002

[44] HU, Y.; PERRIG, A.; JOHNSON, D. – Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks - *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM 2003), April 2003.

[45] KARLOF, C.; LI, Y.; POLASTRE, J. - *ARRIVE: Algorithm for Robust Routing in Volatile Environments* - Technical Report UCB/CSD-03-1233, University of California at Berkeley, May 2002

[46] GANESAN, D.; GOVINDAN, R.; SHENKER, S. AND ESTRIN, D. - Highly-resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks - *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):11–25, 2001.

[47] STADDON, J.; BALFANZ, D.; DURFEE G. – Efficient Tracing of Failed Nodes in Sensor Networks – *Proceedings of First ACM International Workshop on Wireless Sensor Networks and Applications* - WSNA'02, Atlanta, Geórgia, USA, 2002

[48] DIMITRIOU, T, KRONTIRIS, I. - Autonomic Communication Security in Sensor Networks – *Procedings of 2nd International Workshop on Autonomic Communication*, WAC, Athens, Greece, 2005

[49] RUIZ, L. B. RUIZ, NOGUEIRA, J. M. S. AND LOUREIRO, A. A. F. - MANNA: A management architecture for wireless sensor networks. *IEEE communications Magazine*, 41(2): 116-125, February 2003

[50] SONG, H.; KIM, D.; LEE, K. AND SUNG, J. - Upnp-Based Sensor Network Management Architecture - *Proceedings of Second International Conference on Mobile Computing and Ubiquitous Networking*, Osaka, JAPAN, Apr. 2005.

[51] ZHENG, H.; WANG, S.; NICHOLS, R.A. - Policy-based security management for ad hoc wireless systems – *Proceedings of IEEE Military Communications Conference*, 2005. MILCOM 2005 – Oct. 2005, Atlantic City, New Jersey, USA

[52] ANDERSON, R.; KUHN, M. - Low Cost Attacks on Tamper Resistant Devices – *Proceedings of 5th International Workshop on Security Protocols*, Paris, France, April, 1997, Proceedings, Springer LNCS 1361, pp 125

[53] ANDERSON, R.; KUHN, M. - Tamper Resistance - a Cautionary Note. *Proceedings of the Second Usenix Workshop on Electronic Commerce*, pp. 1—11, Oakland, California, November 1996.

[54] MARTINS, M. H. T., SILVA, A. P. R. DA ; LOUREIRO, A. A. F. AND RUIZ, L. B. (2005) *An IDS Simulator for Wireless Sensor Networks* - Technical Report, Comp Sci Dept, Federal University of Minas Gerais, May 2005.

[55] RUIZ, L. B.; SILVA, F. A.; BRAGA, T. R. M.; NOGUEIRA, J. M.; LOUREIRO, A. A. F. - On Impact of Management on Wireless Sensors Networks - *IEEE/IFIP Network Operations and Management Symposium (IX NOMS 2004)*, ISBN 0-7803-8230-7, p. 657-670, 2004, Seul, South Korea

[56] JIANG, Y.; LIN, C.; SHI, M.; SHEN, X. – Key Management Schemes fo Wireless Sensor Networks – *Security in Wireless Sensor Networks* – book chapter - Auerbach Publications, Boca Raton, FL, USA, 2006

[57] VOGT, H. - Exploring Message Authentication in Sensor Networks – *Proceedings of European Workshop on Security in Ad-Hoc and Sensor Networks* – EDAS 2004, Heidelberg, Germany

[58] BARBOSA, Valmir C., (1996) *An Introduction to Distributed Algorithms*, The MIT Press, Cambridge, Massachusetts, 1996.

[59] PARK, S.; SAVVIDES, A. AND SRIVASTAVA, M. B. (2000) - Sensorsim: A Simulation Framework for Sensor Networks, *Proceedings of 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2000, Boston, Massachusetts, USA

[60] LEVIS, P.; LEE, N.; WELSH, M. AND CULLER, D. (2003) Tossim: Accurate and Scalable Simulation of Entire Tinyos Applications, *Procedings of 1st International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, 2003

[61] SHNAYDER, V.; HEMPSTEAD , M.; CHEN , B. RONG; ALLEN, G. W.; AND WELSH, M. (2004), Simulating the Power Consumption of Large-scale Sensor Network Applications, *Proceedings of 2nd International Conference on Embedded Networked Sensor Systems*, 2004, Baltimore, MD, USA

[62] ANTON, E. R.; DUARTE, O. C. - Group Key Establishment in Wireless Ad Hoc Networks - *Workshop em Qualidade de Serviço e Mobilidade* – Nov, 2002, Angra dos Reis, RJ, Brasil

[63] BASAGNI, S.; HERRIN, K.; BRUSCHI, D. AND ROSTI, E. - Secure pebblenet. - *Proceedings of the 2001 ACM Iternational Symposium on Mobile Ad Hoc Networking & Computing*, MobiHoc 2001, pages 156-163, Long Beach, CA, October 4-5 2001

[64] DENG, J.; HAN, R.; MISHRA, S. – A Performance Evaluation of Intrusion-tolerant Routing in Wireless Sensor Network – *Proceedings of 2nd IEEE International Workshop on Information Processing in Sensor Network* – Palo Alto, California, 2003

[65] HEINZELMAN, W; CHANDRAKASAN, A; BALAKRISHNAN, H - Energy-Efficient Communication Protocols for Wireless Microsensor Networks, *Proceedings of Hawaaian Internacional Conference on Systems Science*, January 2000

[66] INTANAGONWIWAT, C.; GOVINDAN, R.; ESTRIN, D. – Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor

Networks – *Proceedings of International Conference on Mobile Computing and Networking*, Mobicom, August 2000, Boston, USA

[67] LEONG, P.H.W.; LEUNG, I.K.H. - A microcoded elliptic curve processor using FPGA technology – *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, Volume 10, Issue 5, Oct 2002 Page(s): 550 - 559

[68] JANSSENS, S., THOMAS, J.; BORREMANS, W.; GIJSELS, P.; VERBAUWHEDE, I.; VERCAUTEREN, F.; PRENEEL, B; VANDEWALLE, J. - Hardware/Software Co-Design Of An Elliptic Curve Publickey Cryptosystem. *In Proceedings of IEEE Workshop on of Signal Processing Systems*, pages 209-216, 2001, Antwerp, Belgium

[69] JOHNSON, D. B. AND MALTZ, D. A.. Dynamic Source Routing in Ad Hoc Wireless Networks. In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, Chapter 5, pages 153-181, Kluwer Academic Publishers, 1996

[70] LIU, D.; NING, P. - Multi-Level µTESLA: A Broadcast Authentication System for Distributed Sensor Networks, *ACM Transactions on Embedded Computing Systems* (TECS), Vol. 3, No. 4, pages 800--836, November 2004.

[71] LIU, D.; NING, P. - *Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks*, NDDSS - Network and Distributed System Security Symposium Conference Proceedings, Internet Society: 2003

[72] LYNCH, C; O'REILLY, F - Processor Choice For Wireless Sensor Networks - REALWSN'05 – *Proceedings of Workshop on Real-World Wireless Sensor Networks*, Stockholm, Sweden, June, 2005

[73] MARTI, S.; GIULI, T.J.; LAI, K.; BAKER, M. - Mitigating Routing Misbehavior in Mobile Ad Hoc Networks, *Proceedings of International Conference on Mobile Computing and Networking*, ACM MobiCOM, August 2000, Boston, Massachusets, USA

[74] MENEZES, A.; OORSCHOT, P. VAN; VANSTONE, S.– *Handbook of Applied Cryptography* – CRC Press, 1996, California

[75] MONKS, J.; BHARGHAVAN, V.; AND HWU, W. W. - A Power Controlled Multiple Access Protocol for Wireless Packet Networks. *Proceedings of Conference on Computer Communications*, INFOCOM, pages 219-228, Tel-Aviv, Israel, 2000

[76] OLIVEIRA, S.; WONG, H. C.; NOGUEIRA, J. M. (2004) - NEKAP: Estabelecimento de Chaves Resiliente a Intrusos em RSSF – *Anais do Simpósio Brasileiro de Redes de Computadores*, Fortaleza, 2004

[77] PERKINS, C. E. - *Ad Hoc Networking*. Addison Wesley, 2001.

[78] PERKINS, C. E. AND ROYER, ELIZABETH M.. Ad hoc On-Demand Distance Vector Routing - *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp. 90-100

[79] PERRIG, A.; CANETTI, R.; SONG, D.; TYGAR, J.D. – Efficient and Secure Source Authentication for Multicast – *Network and Distributed System Security Symposium*, NDSS'01, 2001, San Diego, California

[80] PERRIG, A.; CANETTI, R.; SONG, D.; TYGAR, J.D. – Efficient Authentication and Signing of Multicast Streams over Loss Channels – *IEEE Symposium on Security and Privacy*, 2000, Berkeley, California

[81] RUIZ, L. B.; LOUREIRO, A. A. F.; NOGUEIRA, J. M. S. – Functional and Information Models for the MANNA Architecture - *In Colloque Francophone sur la Gestion de Reseaux et de Services*, pages 455–470, February 2003, Fortaleza, Brazil

[82] SILVA, F. A.; BRAGA, T. R.; RUIZ, L. B.; NOGUEIRA, J. M. - *Tecnologia de Nodos Sensores Sem Fio* – Relatório Técnico DCC/UFMG 001/2003 – janeiro de 2003

[83] SOHRABI, K.; GAO, J.; AILAWADHI, V. AND POTTIE, G. J. - Protocols for self-organization of a wireless sensor network. - *IEEE Personal Communications*, 7(5): 16-27, October 2000.

[84] STEINER, M.; TSUDIK, G. AND WAIDNER, M. - CLIQUES: A New Approach to Group Key Agreement. – *Proceedings of 18th International Conference on Distributed Computing Systems*, pages 380–387, Amsterdam, The Netherlands ,1998.

[85] STEINER, M.; TSUDIK, G., AND WAIDNER, M. - Key agreement in dynamic peer groups. *Proceedings IEEE Transactions on Parallel and Distributed Systems*, 11(8):769–780, 2000.

[86] WALLNER, D.; HARDER, E., AND AGEE, R. - *Key management for multicast: Issues and architectures*. *RFC 2627*, IETF, June 1999.

[87] WOLLINGER, T; PAAR, C. - Hardware Architectures proposed for Cryptosystems Based on Hyperelliptic Curves – *Proceedings of 9th IEEE International Conference on Electronics, Circuits and Systems*, September 15-18, 2002, Dubrovnik, Croatia

[88] WONG, C.; GOUDA, M.; AND LAM, S. - Secure group communications using key graphs - *IEEE/ACM Transactions on Networking* (TON), 8(1):16–30, 2000

[89] Y. W. LAW, S. DULMAN, S. ETALLE, P. HAVINGA - *Assessing Security-Critical Energy-Efficient Sensor Networks*, Technical Report - Department of Computer Science, University of Twente, The Netherlands, junho de 2002

[90] YE, F.; CHEN, A.; LU, S.; ZHANG, L. – A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks – *Proceedings of 10th International Conference on Computer Communications and Networks*, Scottsdale, Arizon, USA, 2001

[91] YE, W; HEIDEMANN, J; ESTRIN, D. - An energy-efficient MAC protocol for wireless sensor networks - *Proceedings of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies* - INFOCOM 2002 - pp. 1567-1576. New York, NY, USA

[92] ZHU, S.; SETIA, S; JAJODIA, S.; XU, S; - LHAP: A Lightweight Hop-by-Hop Authentication Protocol For Ad-Hoc Networks – *Proceedings of 23rd International Conference on Distributed Computing Systems Workshops* (ICDCSW'03), Rhode Island, USA, 2003

Apêndice A – Publicações obtidas

OLIVEIRA, S.; OLIVEIRA, T. R.; NOGUEIRA, J. M. - *A Policy based Security Management Architecture for Sensor Networks – to appear in* 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009), New York, June, 2009. *(short paper)*

Classificação pelos critérios QUALIS: Internacional A

OLIVEIRA, S.; WONG, H. C; NOGUEIRA, J. M. (2008) - *Multiphase Detection and Revocation of Distributed Intrusion in Wireless Sensor Network*, International Journal of Communication Networks and Distributed Systems (IJCNDS), Inderscience Publishers, 2008

Sem classificação pelo QUALIS

OLIVEIRA, S.; OLIVEIRA, T. R.; NOGUEIRA, J. M. (2008) - *Security Management for Sensor Network* - IEEE/IFIP Network Operations and Management Symposium (NOMS 2008), Salvador, Brazil *(short paper)*

Classificação pelos critérios QUALIS: Internacional A

OLIVEIRA, S.; OLIVEIRA, T. R.; NOGUEIRA, J. M. (2008) – *Um Modelo de Gerenciamento de Segurança em Redes de Sensores Sem Fio* – Simpósio Brasileiro de Redes de Computadores (SRBC 2008), Rio de Janeiro, Brasil

Classificação pelos critérios QUALIS: Nacional A

NOGUEIRA, JOSÉ MARCOS, WONG, HAO CHI, LOUREIRO, ANTÔNIO ALFREDO, BEKARA, C, LAURENT-MAKNAVICIUS, M., A. P. RIBEIRO DA SILVA, OLIVEIRA, S., TEIXEIRA, F. A. - *Sécurité dans les réseaux de capteurs sans fil In: La sécurité dans les réseaux sans fil et mobiles 3 : technologies émergentes*. 1ed. Londres - Inglaterra : Hermes Science Publications, 2007, v.3, p. 183-229.

OLIVEIRA, S.; WONG, H. C; NOGUEIRA, J. M. (2007) - *NEKAP: Intruder Resilient and Energy Efficient Key Establishment in Sensor Networks* - Proceedings of 16th International Conference on Computer Communications

and Networks, 2007. ICCCN 2007. Volume , Issue , 13-16 Aug. 2007 Page(s):803 – 808 – Honolulu, Hawai

Classificação pelos critérios QUALIS: Internacional C

Obs: O artigo foi apresentado no "Workshop on Advanced Networking and Communications", evento paralelo ao ICCCN 07, classificado como Qualis A, mas foi publicado no mesmo volume, Procedings of ICCCN 2007, conforme pode ser conferido em:

http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/4317769/4317770/04317916.pdf?arnumber=4317916

OLIVEIRA, S., WONG, H. C., AND NOGUEIRA, J. M. S. - *Alternate Routes for Detection and Increase of Resilience to the Distributed Intrusion in WSN* - IFIP NETWORKING 2006 Workshop on Security and Privacy in Mobile and Wireless Networking (SecPri_MobiWi 2006), Coimbra, Portugal

Classificação pelos critérios QUALIS: Internacional C

OLIVEIRA, S., WONG, H. C., AND NOGUEIRA, J. M. S. – *Rotas Alternativas para Detecção e Aumento da Resiliência à Intrusão Distribuída em RSSF* – Simpósio Brasileiro de Redes de Computadores, 2006, Curitiba, PR

Classificação pelo QUALIS: Nacional A

OLIVEIRA, S., WONG, H. C., AND NOGUEIRA, J. M. S. - *NEKAP: Estabelecimento de Chaves Resiliente a Intrusos em RSSF* – Simpósio Brasileiro de Redes de Computadores, 2005, Fortaleza, CE

Classificação pelo QUALIS: Nacional A

PAULA, W. P.; OLIVEIRA, S.; NOGUEIRA, NOGUEIRA, J. M. S.; WONG, H. C. - *Detecção de Intrusos por Rotas Redundantes em RSSF* - Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, Florianópolis: SBC, 2005.

Classificação pelo QUALIS: Nacional C

OLIVEIRA, S., WONG, H. C., AND NOGUEIRA, J. M. S. - *Segurança em Redes de Sensores sem Fio* – Workshop de Comunicação sem Fio (WCSF'03) - Minicurso, São Lourenço, Brasil

Classificação pelo QUALIS: Nacional C