NEKAP: Estabelecimento de Chaves Resiliente a Intrusos em RSSF

Sérgio de Oliveira[1,2] Hao Chi Wong[2] José Marcos Silva Nogueira[2,3]

[1]Universidade Presidente Antônio Carlos - UNIPAC
MG 482 Km 3 – 36400-000 – Conselheiro Lafaiete – MG – Brazil

[2]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627 – Belo Horizonte – MG – Brazil

[3]Em período sabático nas Universidades de Evry e UPMC/Paris6/Lip6, França

sergiool@lafaiete.unipac.br, {hcwong, jmarcos}@dcc.ufmg.br

Abstract. *Hop-by-hop authentication in Wireless Sensor Networks (WSNs) prevents outside intruders from taking part in the network, and launching various attacks. Due to resource constraints, standard key distribution schemes are inapplicable in this context. Furthermore, the network should be resilient to key compromises by intruders. This paper presents NEKAP - Neighborhood-based Key Agreement Protocol - a key distribution protocol for WSNs. In NEKAP, an intruder that compromises a key has access to the links in a limited neighborhood only, but not those in other regions of the network.*

Resumo. *A autenticação* ponto-a-ponto *em redes de sensores sem fio (RSSF) impede que nós intrusos externos participem da rede e efetuem ataques. Devido a restrições de recursos dessse ambientes, esquemas convencionais de distribuição de chaves não se aplicam adequadamente. Além disso, a rede deve ser resiliente ao comprometimento de chaves por parte de intrusos. Este trabalho apresenta o NEKAP - Neighborhood based Key Agreement Protocol, um protocolo de distribuição de chaves para RSSF. Nesse protocolo, um intruso que compromente uma chave tem acesso a links de uma vizinhança restrita, mas não das outras regiões da rede, característica que é uma evolução em relação a protocolos da mesma categoria existentes na literatura. São aqui apresentados a concepção do protocolo, uma análise da segurança provida pelo mesmo e a implementação dos algoritmos em nós sensores.*

1 Introdução

Redes de Sensores Sem Fio (RSSF) estão surgindo rapidamente como uma tecnologia que irá permitir monitoramento e sensoriamento automatizado em ambientes críticos. Redes de Sensores Sem Fio são redes *ad hoc* formadas por centenas ou milhares de nós sensores de baixo custo com comunicação sem fio. Os nós sensores coletam e enviam informações em comunicação multipasso até uma unidade central, chamada de estação base. Diversas aplicações têm sido propostas para RSSF, desde espionagem militar até supervisão de áreas de monitoração ambiental.

As restrições energéticas, computacionais e de comunicação em RSSF restringem o uso de protocolos e algoritmos complexos. Os nós sensores têm capacidade limitada de processamento, memória e comunicação. Essas limitações vêm

* Este trabalho foi realizado com apoios do CNPq/MCT, processo 55.2111/2002-3, e da CAPES/MEC.

dos requisitos de baixo custo, necessário para que o nó seja usado de forma descartável, e de miniaturização, necessária para o tipo de aplicação.

Como as RSSF são usadas em aplicações críticas, elas podem se tornar alvo de ataques. Alguns fatores tornam as RSSF mais vulneráveis à ação de inimigos que as redes convencionais: recursos computacionais limitados, ambiente hostil e comunicação sem fio. Diversos tipos de ataques podem ser realizados em RSSF: escuta, inserção de mensagens falsas, alteração de mensagens e negação de serviço, entre outros. [Wood & Stankovic 2002] apresentam diversos ataques do tipo negação de serviço que podem ser disparados nessas redes.

O controle de acesso à rede pode impedir e eliminar diversos tipos de ataques, a menos que o inimigo comprometa nós legítimos da rede. Uma forma de efetuar o controle de acesso à rede é implementando autenticação ponto-a-ponto das mensagens que são enviadas dos nós para a estação base, bem como das mensagens que são enviadas da estação base para os nós, como comandos e consultas. Para implementar autenticação ponto-a-ponto, um nó deve ser capaz de autenticar os nós que estão em seu alcance de rádio, também chamados vizinhos, não importando o tipo de comunicação, ponto-a-ponto ou em modo difusão. Faz-se necessário, então, um esquema de distribuição de chaves seguro e eficiente que permita a autenticação em diferentes tipos de comunicação.

A mais detalhada solução de distribuição de chaves apresentada para garantir controle de acesso em RSSF planas e multipassos foi recentemente proposta por Zhu et al [Zhu et al 2003a]. Essa solução, conhecida por LEAP - *Localized Encryption and Authentication Protocol* - provê chaves para diferentes tipos de comunicação. Para o estabelecimento dessas chaves, o LEAP confia em uma simples chave mestra global, carregada com todos os nós antes de seu lançamento, e nos identificadores dos nós trocados entre os nós vizinhos para inicializar todas as outras chaves. É um protocolo altamente distribuído e seguro, desde que a chave global não seja comprometida. Entretanto, se a chave global for comprometida, todas as outras chaves da rede serão também comprometidas. Outro problema do LEAP é o número excessivo de mensagens que são trocadas para o estabelecimento de chaves, pois tanto as chaves ponto-a-ponto quanto as chaves difusão exigem uma mensagem de cada nó para cada vizinho.

Este trabalho apresenta o NEKAP, um protocolo de estabelecimento de chaves a serem usadas na autenticação ponto-a-ponto que não sofre os mesmos inconvenientes do LEAP, sendo possível usá-las também na encriptação. Nesse protocolo, cada nó é carregado inicialmente com uma chave mestra diferente e cada nó envia sua chave para os seus vizinhos, em difusão, encriptada com uma chave conhecida globalmente. A chave difusão é gerada a partir da chave mestra, de forma que todos os vizinhos podem gerar essa chave após a divulgação das chaves mestras. As chaves ponto-a-ponto 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 deste trabalho é o protocolo de distribuição de chaves, onde 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 através da captura de poucos nós. Além disso, o custo energético da solução aqui apresentada é menor que os outros até então apresentados.

2 Preliminares

2.1 Controle de Acesso em RSSF

O controle de acesso é uma função crítica em RSSF. Se qualquer nó tiver permissão para participar da rede, um nó malicioso pode se inserir na rede, e efetuar todo tipo de ataque. O controle de acesso é tipicamente implementado através de mecanismo criptográfico e distribuição adequada de chaves. Um mecanismo de controle de acesso efetivo deve suportar comunicação autenticada para possibilitar a um nó reconhecer o transmissor de uma mensagem recebida como sendo um nó legítimo. Por esse motivo, deve existir um mecanismo de chaves que suporta os vários padrões de comunicação existentes em RSSF. Além disso, devido aos recursos limitados desse tipo de ambiente, deve existir uma relação muito eficiente entre segurança das chaves estabelecidas e energia gasta para estabelecê-las.

Os padrões de comunicação em RSSF são determinados pela forma que a rede é organizada, entre outros aspectos. Em uma rede plana, sem hierarquia entre os nós, na qual os nós confiam em seus vizinhos para rotear suas mensagens, pode haver quatro tipos de comunicação: 1) comunicação ponto-a-ponto entre um nó e a estação base; 2) comunicação ponto-a-ponto entre dois nós vizinhos; 3) difusão local de um nó para seus vizinhos e; 4) difusão global da estação base para o restante da rede. Em cada tipo de comunicação, todos os elementos envolvidos devem dispor de chaves de autenticação.

Existem alguns esquemas de estabelecimento de chaves que poderiam resolver o problema: 1) esquemas baseados em chave pública; 2) esquemas baseados em um KDC (*Key Distribuction Center*, uma entidade central confiável), com o qual cada nó mantém uma chave compartilhada; 3) compartilhamento completo de chaves ponto-a-ponto entre quaisquer dois nós da rede. Nenhum desses esquemas é viável, todavia, devido aos recursos limitados das redes de sensores: o esquema 1), pelo alto custo computacional exigido pelos algoritmos de chave pública; o esquema 2), pelo alto custo de comunicação para estabelecimento de chaves entre dois nós quaisquer; e o esquema 3), pelo alto custo de armazenamento de todas as chaves de uma rede com grande número de nós.

Para resolver o problema de distribuição apresentado, Zhu *et al.* propuseram recentemente o protocolo LEAP[Zhu et al 2003a], cujas principais idéias serão apresentadas a seguir.

2.2 LEAP

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 ponto-a-ponto (*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.

As chaves individuais são geradas e pré-carregadas nos nós antes do seu lançamento. As chaves ponto-a-ponto 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. Cada 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 ponto-a-ponto 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 ponto-a-ponto, 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 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 deve é enviada a todos os vizinhos junto à chave difusão. A cada mensagem enviada, uma nova chave da cadeia é divulgada. Detalhes sobre esse tipo de cadeia de chaves são apresentados na seção 3.1.

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 [Perrig *et al.* 2002].

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, K_I é apagada de todos os nós após a geração das chaves ponto-a-ponto, 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 obtido o conteúdo de chaves, o atacante terá acesso à chave K_I, e assim poderá obter todas as chaves ponto-a-ponto da rede.

Outro problema do LEAP é o número excessivo de mensagens para configuração 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 ponto-a-ponto e outra para distribuição da chave difusão. Em redes densas, com média acima de 20 vizinhos, o custo pode ser consideravelmente alto para esse tipo de rede.

3 NEKAP

3.1 Modelo e Definições

Este trabalho foi desenvolvido para RSSF planas e foi assumido o mesmo modelo de rede e segurança adotado pelo LEAP. O nó usado como referência é o nó Berkeley Mica2 Motes [Crossbow 2004], desenvolvido na Universidade de Berkeley. Esse tipo de nó sensor foi escolhido por estar disponível comercialmente, o que viabiliza a validação pela implementação e testes em ambientes reais. Em particular, cada nó tem capacidade de armazenar algumas centenas de bytes de chaves.

A energia utilizada para transmissão (27 mA a 38,4 Kbaud) é razoavelmente maior que a energia gasta no processamento (8 mA a 4 MHz) [Crossbow 2004]. Dessa forma, as operações de rede são mais caras e demoradas que as operações de processamento.

A distribuição dos nós é aleatória e a vizinhança de qualquer nó não é conhecida previamente. A comunicação sem fio não é segura e é sujeita a escuta, inserção de pacotes e replicação de mensagens. Os nós são sujeitos a adulteração física (*tampering*). Se um nó é comprometido, todas as informações que ele manipula podem ser conhecidas pelo atacante. Um pressuposto válido é que a estação base, única e com recursos ilimitados, não é nunca comprometida.

Uma cadeia de chaves de via única (*one way key chain*) é uma seqüência de chaves geradas por uma função não reversível. A chave de ordem *n-1* da cadeia é gerada a partir da *n*-ésima chave. Assim, o conhecimento da chave *n-1* não permite identificar a chave *n*. Porém, o conhecimento da chave *n* permite identificar a chave *n-1*. Neste documento, todas as vezes que for citado o termo cadeia de chaves, será uma referência para uma cadeia de chaves de via única.

3.2 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 difusão, usada pelo nó *A* para enviar mensagens em difusão;
- K_{AB}: Chave ponto-a-ponto, usada nas comunicações ponto-a-ponto 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, 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 ponto-a-ponto;

3.3 Descrição do Protocolo

NEKAP é um protocolo para estabelecimento de chaves em RSSF. Essas chaves são usadas para autenticação ponto-a-ponto, e possivelmente para encriptação. São estabelecidas chaves difusão e chaves ponto-a-ponto. Chaves individuais e chaves grupais não fazem parte do escopo deste trabalho, por já terem sido exploradas de forma satisfatória na literatura, como em [Perrig *et al* 2002] e [Karlof *et al* 2004].

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.

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 1 apresenta o conjunto dos vizinhos de um nó sensor A, delimitados pelo alcance do seu rádio. Depois do estabelecimento da chave difusão, todo nó deve ter uma chave difusão para cada um de seus vizinhos.

Como no LEAP, além da chave 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.

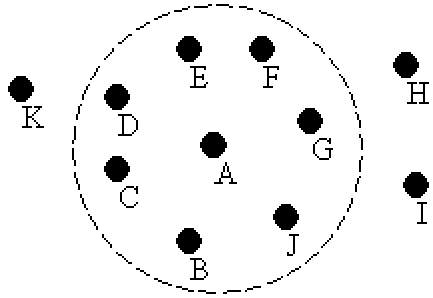

Figura 1 - Vizinhos do nó sensor A

Ao término da cadeia de chaves, que é finita e pequena, devido às 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 ponto-a-ponto, 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.

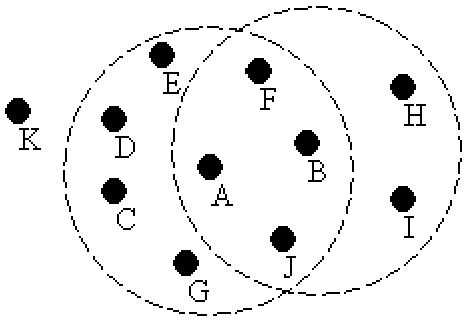

Figura 2 - Intersecção dos vizinhos de A e vizinhos de B

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 2, 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. Tanto o nó A quanto o nó B conhecem as chaves mestras desses nós e podem usar essas informações para gerar a chave ponto-a-ponto que será usada na comunicação entre esses dois nós. Para aumentar a segurança desse protocolo, são usadas todas as informações em comum, além dos identificadores dos 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 da 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.

3.4 Protocolo

3.4.1 Chaves de difusão

Inicialmente, cada nó envia seu identificador em difusão e escuta todas as mensagens enviadas em difusão por seus vizinhos (Figura 3, 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 3, 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. $A \Rightarrow *: \{K_{MA}, K_{A1}\}_{K_G}, HMAC(K_G, \{K_{MA}, K_{A1}\}_{K_G})$

Figura 3 - 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 4. Caso seja necessária a encriptação da mensagem de difusão, ela será feita somente com a chave difusão, como mostra a figura 5.

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

Figura 4 - Mensagem em difusão autenticada

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

Figura 5 - Mensagem em difusão encriptada e autenticada

3.4.2 Chaves ponto-a-ponto

Dados dois nós A e B, sua chave ponto-a-ponto 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). As mensagens incluem os identificadores dos vizinhos e são enviadas usando os mecanismos de autenticação em difusão.

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

Figura 6 - 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 ponto-a-ponto 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 ponto-a-ponto 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. Como exemplo, na figura 2, a chave ponto-a-ponto a ser usada pelos nós A e B seria dada por: $K_{AB} = f(Id_A, Id_B, K_A, K_B, K_F, K_J)$.

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

3.4.3 Inserção de Novos Nós

A inserção de novos nós é um problema em esquemas de autenticação ponto-a-ponto [Vogt 2004]. 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 7, passo 1). Em seguida, informações necessárias para o estabelecimento das chaves são enviadas junto aos rótulos de inserção (Figura 7, 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 7, 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 7, passo 4). Essa chave é encriptada e enviada ponto-a-ponto, 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 7 - 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 ponto-a-ponto, 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 ponto-a-ponto (Figura 6). Assim, as chaves ponto-a-ponto podem ser estabelecidas da mesma forma que são estabelecidas durante a inicialização.

4 Análise de Segurança

O controle de acesso, através da autenticação ponto-a-ponto, conforme proposta deste trabalho e também apresentada no LEAP, elimina a possibilidade de diversos 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.

Para efetuar um ataque interno, promovido por nós maliciosos reconhecidos 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, através de um nó que não tenha se inicializado corretamente.

Com o uso do protocolo NEKAP, as chaves são distribuídas de forma caótica. 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 ponto-a-ponto 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 ponto-a-ponto 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. Como as chaves são limitadas à vizinhança, seu uso em outros pontos não permite ao inimigo ser reconhecido como um nó da rede.

4.1 Avaliação

Para verificar o impacto da inserção de uma escuta e obtenção da chave global pelo inimigo no protocolo NEKAP, foram realizadas simulações para verificar quantas chaves ponto-a-ponto poderiam ser obtidas por uma escuta. As simulações foram realizadas a partir de distribuições aleatórias, além de uma distribuição uniforme, geradas no Microsoft Excel. As coordenadas dos nós foram salvas em arquivos e lidas posteriormente por um programa escrito em C, que realizou o processo de troca de chaves, indicando as possíveis vulnerabilidades.

Uma escuta durante o processo de inicialização e a conseqüente descoberta da chave global têm 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.

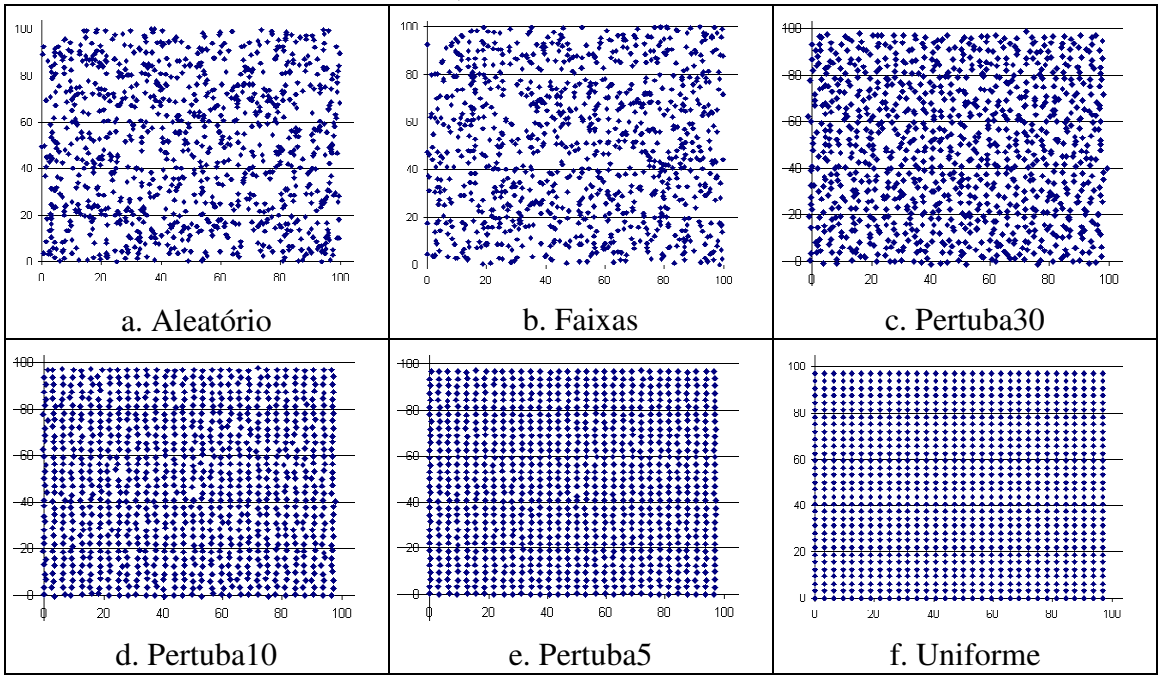

Figura 8 - Distribuições de nós simuladas

A figura 8 apresenta graficamente as distribuições de nós sensores. Verifica-se a redução da aleatoriedade a partir da primeira para a última distribuição.

A intenção do 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 ponto-a-ponto têm maior interesse do inimigo. Foi verificado, então, o número de chaves ponto-a-ponto que podem ser conhecidas através da adulteração de um nó na inicialização. Os resultados da simulação podem ser vistos na Tabela 1. Foram coletados os seguintes dados:

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

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

Tabela 1 - Resultados da simulação

Essa última informação (III) é calculada a partir de cada nó. Durante a simulação, são verificadas quantas chaves cada nó poderia obter, caso estivesse adulterado 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 ponto-a-ponto, 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 ponto-a-ponto que ele estabelece com seus vizinhos, outras 17 chaves ponto-a-ponto 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.

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 ponto-a-ponto da rede, e não apenas aquelas trocadas na vizinhança. Na simulação Pertuba30, isso representaria 9320 chaves ponto-a-ponto.

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.

5 Implementação

A implementação do protocolo de distribuição de chaves aqui apresentado foi realizada para o nó sensor Mica2 Motes [CrossBow 2004], 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 ponto-a-ponto 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 ponto-a-ponto, 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 Trabalhos Relacionados

O protocolo LEAP [Zhu *et al* 2003b] é a proposta para distribuição de chaves que mais se aproxima desse trabalho. O protocolo foi descrito e discutido na seção 2.2.

[Eschenauer & Glicor 2002] e [Chan *et al.* 2003] apresentam protocolos de pré-distribuição aleatória de chaves. Cada nó é lançado com algumas chaves que pertencem a um conjunto maior de chaves. A distribuição dessas chaves é aleatória. Para que dois nós vizinhos possam se comunicar, é necessário que eles compartilhem uma ou mais chaves. Caso dois nós não compartilhem chaves, eles podem estabelecer chaves através de outros nós que tenham conseguido estabelecer comunicação segura entre eles. 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, multiplicando 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.

Liu e Ning propuseram variações para a pré-distribuição aleatória de chaves com o objetivo de aumentar a probabilidade de existirem chaves em comum e reduzir o consumo de energia no processo de estabelecimento [Liu & Ning 2003].

7 Conclusões

Este artigo apresentou o protocolo NEKAP, para estabelecimento de chaves para RSSF. As chaves são usadas para garantir a autenticação ponto-a-ponto e conseqüente controle de acesso. O uso dessa abordagem permite eliminar diversos tipos de ataques promovidos pelo inimigo. O principal objetivo desse trabalho é 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 ponto-a-ponto 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.

O uso da solução aqui apresentada, em conjunto com outras soluções já existentes na literatura, como autenticação fim-a-fim [Perrig *et al.* 2002] permite um aumento significativo da eficiência da rede na presença de um inimigo, evitando a maioria dos ataques conhecidos.

8 Referências

A. D. Wood; J. A. Stankovic – *Denial of Service in Sensor Networks* – IEEE Computer, October 2002.

A. Perrig; R. Szewczyk; J. D. Tygar; V. Wen; D. E. Culler – *SPINS: Security Protocols for Sensor Networks* – Wireless Networks 8, 2002, Kluwer Academic Publishers, Netherlands.

Chris Karlof, Naveen Sastry, and David Wagner, *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.

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 10 de março de 2004, San Jose, CA, USA, February 2004.

Donggang Liu, Peng Ning - *Establishing pairwise keys in distributed sensor networks* - Proceedings of the 10th ACM conference on Computer and communication security - Washington D.C., - USA – 2003.

H. Chan; A. Perrig; D. Song – *Random Key Predistribution Schemes for Sensor Networks*, 2003 IEEE Symposium on Security and Privacy May 11 - 14, 2003 Berkeley, CA, p. 197, 2003.

Harald Vogt - *Exploring Message Authentication in Sensor Networks* - European Workshop on Security in Ad-Hoc and Sensor Networks – EDAS 2004, Heidelberg, Germany.

L. Eschenauer; V. D. Glicor – *A Key-Management Scheme for Distributed Sensor Network* – in Proceedings of the 9th ACM conference on Computer and Communication Security, November 2002.

Sencun Zhu, Sanjeev Setia, Sushil Jajodia - *LEAP: efficient security mechanisms for large-scale distributed sensor networks* - 10th ACM Conference on Computer and Communications Security (CCS '03), Washington D.C., October, 2003.

Stefan Schmidt, Holger Krahn, Stefan Fischer, and Dietmar Wätjen - *A Security Architecture for Mobile Wireless Sensor Networks* - European Workshop on Security in Ad-Hoc and Sensor Networks - ESAS 2004, Heidelberg, Germany, 2004.

Yann-Hang Lee, Amit Deshmukh, Vikram Phadke, Jin Wook Lee – *Key Management in Wireless Sensor Networks* - European Workshop on Security in Ad-Hoc and Sensor Networks - ESAS 2004, Heidelberg, Germany, 2004.