Segurança no Roteamento em Redes Móveis *Ad Hoc*

Aurelio Amodei Junior , Otto Carlos M. B. Duarte

[1] Grupo de Teleinformática e Automação
COPPE/EE - Programa de Engenharia Elétrica
Universidade Federal do Rio de Janeiro
http://www.gta.ufrj.br/

{aurelio,otto}@gta.ufrj.br

Abstract. *Routing plays a very important role in mobile ad hoc networks, though that in these networks there is no infrastructure, and all the nodes have to act as routers. So, malicious attacks from nodes in the network could interfere in the routing protocol mechanism, and affect the performance of the network. This paper presents the security problems of two of the most used routing protocols nowadays. It is also presented some of the proposed solutions, as the Secure Ad Hoc On Demand Distance Vector (SAODV) and Ariadne, and how they deal with these attacks. SAODV and Ariadne are, respectively, secure versions of AODV and DSR, added with some security mechanisms to accomplish integrity and non-repudiation of the routing messages.*

Resumo. *O roteamento exerce um papel fundamental nas redes móveis ad hoc, já que neste tipo de rede não há infra-estrutura, e todos os nós são responsáveis pelo roteamento. Logo, a ação de alguns nós maliciosos na rede pode comprometer o funcionamento do protocolo de roteamento, diminuindo o desempenho da rede. Este artigo apresenta os problemas de segurança encontrados nos protocolos de roteamento mais utilizados atualmente, e algumas das soluções propostas, o Secure Ad Hoc On Demand Distance Vector (SAODV) e o Ariadne, e como eles previnem esses ataques. O SAODV e o Ariadne são, respectivamente, versões seguras do AODV e do DSR, adicionados de alguns mecanismos de segurança, que garantem a integridade e não-repúdio das mensagens.*

1. Introdução

As redes móveis sem fio são sistemas de comunicação onde as estações, que são nós móveis, se comunicam por meio de enlaces de rádio. Esse tipo de rede pode ser classificada em dois tipos: as redes infra-estruturadas, e as redes *ad hoc*, ou sem infra-estrutura. As redes infra-estruturadas são caracterizadas pela presença de um terminal centralizador, chamado de ponto de acesso. O ponto de acesso é o responsável por centralizar certas funções da rede, como o roteamento e o controle de acesso ao meio.

As redes *ad hoc*, que são o alvo deste trabalho, possuem como característica não possuir nenhum tipo de infra-estrutura previamente estabelecida. Assim, esse tipo de rede pode ser rapidamente construído, mesmo em ambientes hostis ou onde seria muito difícil a prévia instalação de uma infra-estrutura. Nas redes *ad hoc*, os nós são responsáveis por

desempenhar as funções que antes eram desempenhadas pelo ponto de acesso, logo, cada nó da rede se torna um roteador, e é responsável por reencaminhar as mensagens de outros nós que desejem se comunicar com um nó que esteja fora da sua área de alcance. Como os nós podem se mover aleatoriamente, alterando dinamicamente a topologia da rede, os protocolos de roteamento utilizados nas redes *ad hoc* devem ser adaptativos e capazes de encontrar rotas nesse cenário de alta mudança de conectividade.

Muito já foi estudado, e vários protocolos de roteamento foram propostos para redes *ad hoc* [Royer and Toh, 1999]. Entretanto, a maior parte desses protocolos foram propostos e analisados em cenários idealizados, onde não foram abordados os problemas de segurança. Não foram definidos então requisitos de segurança, confiando-se em todos os nós das rede para realizar o roteamento. Entre os dois principais protocolos atualmente, o DSR [Johnson and Maltz, 1996] e o AODV [Perkins and Royer, 1999], este último já como uma RFC do IETF, e o primeiro ainda como *draft*, nenhum deles implementa algum mecanismo de segurança para o roteamento.

Vários artigos então analisaram os problemas de segurança nos protocolos de roteamento, e os tipos de ataques que poderiam ser feitos às redes [Wang et al., 2002] [Wang and Bhargava, 2002] [Zhou and Haas, 1999], e foram propostos então protocolos que também implementavam mecanismos de segurança, tentando impedir que a ação maliciosa de alguns nós nas tarefas de roteamento possa afetar o desempenho da rede. Entre eles, podemos citar o Ariadne [Hu et al., 2002a], que é uma versão segura do DSR, e o *Secure* AODV (SAODV) [Guerrero Zapata and Asokan, 2002], que é uma versão segura do AODV, que são duas propostas que serão apresentadas neste artigo. Além dessas, podemos citar algumas outras como o ARAN [Sanzgiri et al., 2002], o SEAD [Hu et al., 2002b] e o SRP [Papadimitratos and Haas, 2002].

Na Seção 2, abordaremos então uma visão geral dos mecanismos de roteamento existentes atualmente nas redes *ad hoc*, apresentando com mais detalhes o DSR e o AODV. Na Seção 3, são apresentados os aspectos principais da segurança no roteamento de redes *ad hoc*, os tipos de ataques que podem ser feitos a esses protocolos de roteamento e os mecanismo que são utilizados nas soluções propostas. As Seções 4 e 5 apresentam então, respectivamente, duas soluções propostas para o problema da segurança no roteamento, o SAODV e o Ariadne. Na seção 6 concluímos o trabalho.

2. Roteamento em Redes *Ad Hoc*

Tradicionalmente, podemos dividir os protocolos de roteamento em pró-ativos e reativos. Protocolos pró-ativos se caracterizam por estarem periodicamente aprendendo e atualizando a topologia da rede, mantendo em tabelas as informações atualizadas das rotas para se chegar a todos os nós da rede. Assim, sempre que é necessária uma rota para algum destino, essa informação está disponível imediatamente.

Esse tipo de protocolo já é classicamente utilizado em redes cabeadas, e pode utilizar diferentes algoritmos, como os baseados em vetor de distância e em estado do enlace. Sua principal vantagem é o baixo tempo de latência necessário para o estabelecimento da conexão, já que sempre que um nó precisa enviar um pacote para um destino, ele já possui a rota desejada. No entanto, essas rotas são obtidas ao custo de um maior *overhead* na rede, com mensagens de roteamento sendo enviadas periodicamente, e/ou utilizando-se

a técnica de inundação para propagar as informações de roteamento por toda a rede, e manter as informações consistentes, evitando ciclos (*loops*) indesejáveis e rotas erradas.

Os primeiros protocolos de roteamento utilizados em redes *ad hoc* também eram protocolos pró-ativos. Alguns exemplos desses protocolos são o OLSR (*Optimized Link State Routing*) [Clausen et al., 2001], que utiliza algoritmo de estado do enlace, e o DSDV (*Destination Sequence Distance Vector*) [Perkins and Bhagwat, 1994], que utiliza o algoritmo clássico de Bellman-Ford [Bellman, 1958] [Ford and Fulkerson, 1962], baseado em vetores de distância.

No entanto, com a mobilidade da rede, ocorrem constantes quebras de enlaces e frequentes mudanças de topologia, que podem aumentar muito o custo de se manter sempre atualizadas as informações topológicas da rede. Principalmente quando a carga da rede é baixa, fazendo com que muitas das rotas que são mantidas atualizadas não cheguem nem a ser utilizadas, causando um desperdício de banda e processamento.

Com isso, passou-se a utilizar os protocolos reativos, ou também chamados de sob demanda. Nesse tipo de protocolo, sempre que há a necessidade de uma rota para algum destino, o protocolo começa um processo de descoberta de rota para esse destino. Esse processo é baseado em uma espécie de inundação da rede com o pedido de rota, até que o destino seja alcançado. O destino envia então para a fonte um reconhecimento de que foi alcançado, com a devida rota utilizada. Nessa abordagem, a principal vantagem é o fato de se ter um menor *overhead* na rede, já que esse tipo de rede apresenta rotas muito dinâmicas. Muitas das mensagens de roteamento de um protocolo pró-ativo acabariam se tornando desnecessárias, já que enlaces poderiam se romper antes das rotas que passassem por ele chegassem a ser utilizadas, tornando mais eficiente a descoberta de rotas sob demanda. Contudo, esse mecanismo acarreta em um aumento do tempo de latência do estabelecimento das conexões, o que se mostra ser uma troca vantajosa na maioria dos casos. Os protocolos reativos mais comuns e mais utilizados em redes *ad hoc* são o DSR (*Dynamic Source Routing*) e o AODV (*Ad Hoc On Demand Distance Vector*).

Neste artigo, abordaremos estes dois protocolos para analisar os seus aspectos de segurança, suas falhas, e analisaremos também versões seguras dos dois, o SAODV e o Ariadne.

2.1. DSR

O DSR é um protocolo de roteamento reativo, ou sob demanda. Sua característica principal, é a utilização de roteamento por fonte (*source routing*), onde o nó que está originando um pacote sabe toda a rota salto a salto até o destino. Assim, o DSR permite que a estação originadora do pacote determine o caminho que será utilizado pelo pacote na rede para chegar até o seu destino. Esse caminho é listado no cabeçalho do pacote de dados e é chamado de *source route*.

Cada nó da rede mantém um *cache* de rotas dinâmico, onde ele armazena as rotas para outros nós da rede que ele aprendeu iniciando um pedido de rota (*route request*) para esse destino, ou por ter encaminhado pacotes ao longo de um outro caminho que passasse por esse nó. Além desse procedimento de descoberta de rota (*route discovery*), um nó também pode aprender rotas ouvindo a transmissão de nós vizinhos em rotas das quais ele não faz parte. Esse procedimento é opcional no DSR, e as interfaces de rede deverão funcionar no modo promíscuo para o seu funcionamento.

Quando um nó deseja enviar um pacote para outro nó da rede, ele primeiro verifica se ele já não possui uma rota armazenada para esse destino. Caso ele possua, ele insere essa rota no cabeçalho do pacote, listando os endereços dos nós pelos quais o pacote deverá ser encaminhado pela rede até o destino. O primeiro nó dessa lista receberá o pacote, e o enviará ao próximo nó da rede, e assim por diante até o nó final. Caso o nó originador do pacote não possua uma rota para esse destino, ele inicia o processo de descoberta de rota do DSR, enviando em difusão um pacote de pedido de rota (RREQ). Cada RREQ é unicamente identificado pelo conjunto endereço de origem, endereço de destino e identificador do pedido (*request id*).

Ao receber um pacote de RREQ, um nó verifica por esse conjunto se ele já não recebeu esse pedido de rota antes. Em caso positivo, ele descarta o pacote; do contrário, ele irá verificar se ele possui uma rota para o nó destino do pedido. Se ele possuir uma rota para o destino, ou for ele mesmo o nó destino, ele envia de volta para o originador do pacote, em *unicast*, um pacote de *route reply* (RREP), contendo a rota que será utilizada para chegar até o destino. Se o nó intermediário não possuir essa rota, ele simplesmente irá reenviar em difusão esse pacote de pedido de rota, adicionando ao final da lista de nós da rota o seu endereço, num processo de inundação, até que se chegue ao destino. A Figura 1 mostra um exemplo de um processo de descoberta de rota no DSR.

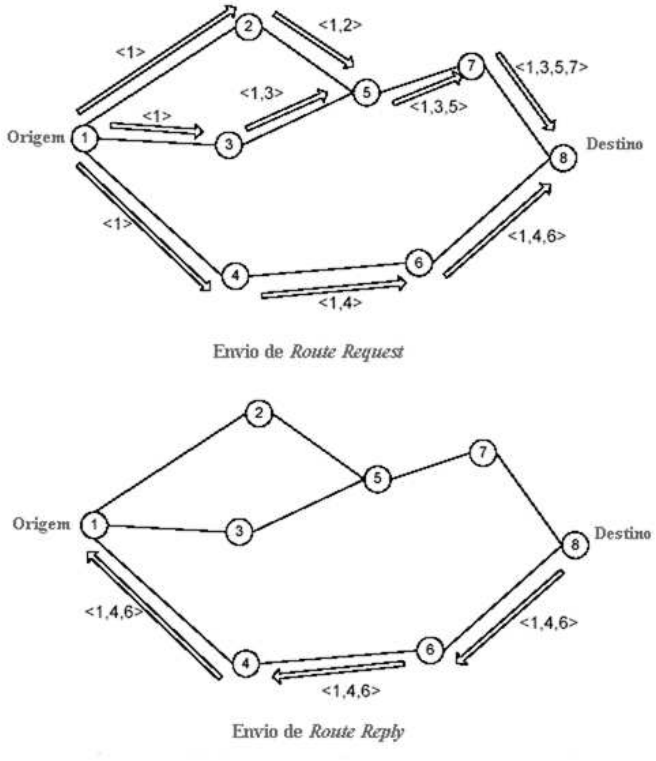


Figura 1: Descoberta de Rota no DSR.

Se algum dos enlaces em uma rota é rompido, o nó de origem é notificado com um pacote de *route error* (RERR). Ele então remove todos as rotas que utilizassem esse enlace, e utiliza outra rota para o destino, ou inicia uma nova descoberta de rota caso não possua mais rotas.

2.2. AODV

O AODV funciona de maneira muito semelhante ao DSR, já que ambos compartilham as características de protocolos reativos. Quando uma rota é requisitada, ele utiliza um processo de descoberta de rota similar ao do DSR, no entanto, o AODV utiliza um mecanismo diferente para armazenar as informações de roteamento. Ao invés de armazenar toda a rota, ele utiliza uma tabela de roteamento tradicional, o que significa que ele mantém apenas uma entrada por destino. Essa entrada armazena apenas o próximo salto para esse destino, enquanto o DSR armazena a rota completa, e até múltiplas rotas para um mesmo destino.

No AODV, cada nó descobre ou mantém informações de roteamento para outro nó somente se ele está se comunicando diretamente com esse nó, ou se ele é um nó intermediário em uma mesma rota. O AODV é capaz de manter rotas sem *loops*, mesmo quando os enlaces mudam em rotas ativas.

A descoberta de rotas é feita enviando em difusão pacotes de RREQ. Cada nó que recebe pela primeira vez um pacote de RREQ coloca na sua tabela de roteamento uma entrada com a rota reversa, para o originador da mensagem, e reenvia esse pacote de RREQ. Ao chegar no destino, ou em um nó intermediário que possua uma rota para o destino, o caminho para o nó origem já estará formado, e o nó destino enviará um pacote de RREP em *unicast* para o nó origem, utilizando essa rota. De acordo com que o pacote de RREP é enviado de volta, os nós intermediários estabelecem também as rotas no sentido direto, para o nó destino. Assim, ao chegar de volta no nó origem, a rota já estará estabelecida nos dois sentidos, e o nó origem pode começar a enviar pacotes para o nó destino. A Figura 2 apresenta um exemplo de um processo de descoberta de rota no AODV.

O AODV mantém rotas atualizadas utilizando um contador para cada nó chamado de número de sequência. O número de sequência de um nó é incrementado cada vez que a conectividade local desse nó muda. Um pacote de pedido de rota (RREQ) contém campos tanto para o número de sequência do nó de origem, como para o do nó de destino, chamados de *source_sequence_num* e *destination_sequence_num*, respectivamente. O pacote de RREP também possui um campo para o número de sequência do destino. Quando um nó recebe um pedido de rota, ele determina a idade da sua entrada na tabela de roteamento para esse destino - caso essa entrada exista - comparando o número de sequência do destino daquele RREQ com o número da tabela para esse destino. Caso a rota da sua tabela de roteamento seja mais recente, e consequentemente mais atualizada, ele responde ao pedido de rota com um RREP com a sua rota, do contrário ele reenvia o pedido de rota para os seus vizinhos para que uma nova rota seja descoberta.

Pacotes de pedido de rota também possuem um campo identificador chamado de *broadcast_id*. O valor desse campo é um contador mantido para cada nó, e que é incrementado cada vez que um nó realiza um novo pedido de rota. Os nós então podem armazenar temporariamente essa informação para cada RREQ recebido, para que eles possam identificar se um RREQ recebido já foi processado ou não. Caso o pacote recebido já tenha sido processado, ele descarta o pacote, prevenindo assim a formação de *loops*.

Da mesma forma que o DSR, quando um enlace de uma rota é rompido, é enviado

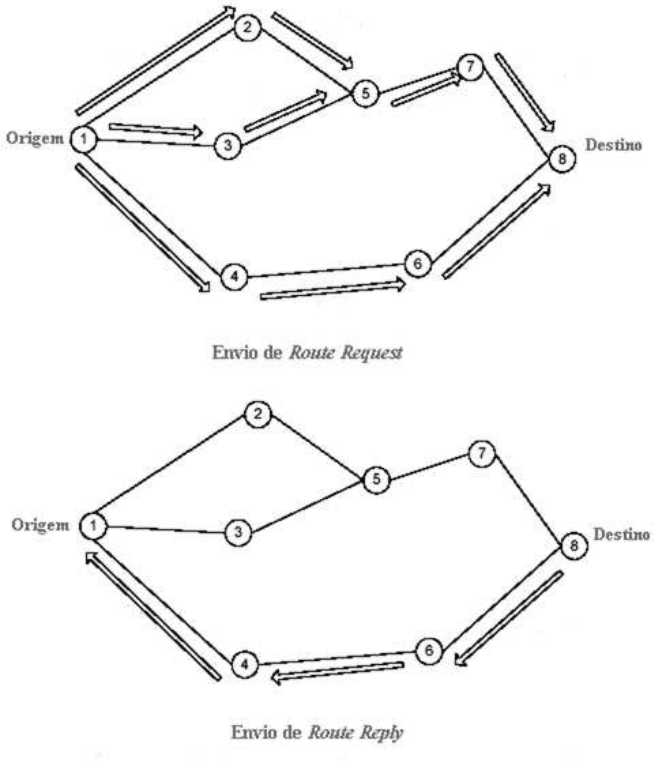


Figura 2: Descoberta de Rota no AODV.

um pacote de RERR de volta para a origem. Esse pacote ao ser reencaminhado pela rota para o nó de origem, apaga a rota dos nós intermediários pelos quais ele vai passando, cancelando a rota, e possibilitando um novo processo de descoberta de rota.

3. Segurança no Roteamento

Os protocolos de roteamento atualmente utilizados foram propostos e analisados em cenários idealizados, onde não foram abordados os problemas de segurança relativos ao roteamento nessas redes. Todos os nós da rede foram considerados confiáveis, e não foram levados em contas os tipos de ataques que poderiam afetar esses protocolos, especialmente com a utilização de nós maliciosos na rede. Esses nós poderiam então comportar-se de forma maliciosa e não cooperativa, resultando na degradação ou interrupção da operação da rede.

Podemos classificar inicialmente os ataques ao roteamento em ataques passivos e ativos. Os ataques passivos são considerados aqueles em que um nó ignora as funções que ele deveria realizar [Wang et al., 2002] [Wang and Bhargava, 2002], agindo como se ele não estivesse ali. Essa definição é ligeiramente diferente da definição utilizada para redes cabeadas, onde somente a escuta do meio seria considerado um ataque passivo, enquanto aqui o não reencaminhamento de mensagens também é considerado um ataque passivo. Isso se deve pois nas redes cabeadas, a probabilidade de perda de mensagem é muito pequena, e o descarte de uma mensagem seria um evento incomum nesse tipo de rede. Já numa rede sem fio, a probabilidade de perda aumenta muito, principalmente com a mobilidade da rede. Assim, a perda de uma mensagem seria um evento muito mais

comum em uma rede sem fio, então caso um nó não reencaminhe uma mensagem, pode-se simplesmente assumir que o nó não recebeu a mensagem, ou que ele se moveu para fora do alcance do nó anterior. A rede, na sua própria constituição, já se previne e trata esse tipo de eventos, logo os ataques passivos não serão o alvo deste artigo.

Os protocolos seguros abordados nesse artigo têm como objetivo lidar com o problema dos ataques ativos à rede. Em um ataque ativo, um nó malicioso pode injetar, modificar ou replicar informações errôneas sobre a operação da rede, e ainda comportar-se de forma maliciosa e não cooperativa, afetando então o mecanismo de roteamento da rede. Entre os ataques ativos, podemos citar três tipos principais de ataques que são utilizados: os ataques de Modificação, de Fabricação e de Personificação (*Impersonating*) [Sanzgiri et al., 2002].

3.1. Tipos de Ataque

3.1.1. Ataques de Modificação

Nesse tipo de ataque, o nó malicioso altera informações de mensagens de roteamento recebidas, gerando informações de rotas falsas, ou tentando atrair para si o tráfego da rede, fazendo com que todas as rotas passem por ele.

No DSR e no AODV, os principais ataques desse tipo são os ataques de vetor de distância, e os de número de sequência [Wang et al., 2002]. No ataque de vetor de distância, o nó malicioso informa uma métrica menor para a sua rota, fazendo com que os nós adjacentes prefiram utilizar a rota que passa por ele. No DSR isso pode ser feito diminuindo o número de nós na lista de nós da *source route*, diminuindo então o número de saltos. No AODV, o nó malicioso simplesmente diminui o valor do campo *Hop Count*.

Em um ataque de número de sequência, o nó malicioso, ao receber uma mensagem de roteamento, altera o valor do número de sequência do destino, fazendo assim com que todos os nós que receberem essa rota passem a utilizá-la, já que o seu número de sequência é muito maior, e eles a considerariam então uma rota mais atualizada que as outras. Esse ataque é realizado da mesma maneira no DSR e no AODV.

3.1.2. Ataques de Fabricação

Há uma grande variedade de possibilidades para se realizar um ataque de Fabricação. Um nó malicioso na rede poderia gerar qualquer tipo de mensagem de roteamento, RREQ, RREP ou RERR, cada uma tendo uma consequência diferente para o desempenho da rede. Juntamente com o tipo de mensagem, o nó malicioso também poderia utilizar valores de número de sequência muito altos, rotas falsas ou número de saltos pequenos, atraindo o tráfego para si.

Um dos ataques desse tipo que afetaria mais a rede, seria a fabricação de uma mensagem de RERR para um certo destino, tornando esse nó inalcançável. Caso seja utilizado também um número de sequência muito alto, esse ataque poderia fazer com que esse nó ficasse inalcançável durante um longo período na rede.

3.1.3. Ataques de Personificação

Em um ataque de personificação, o nó malicioso utiliza o endereço de outros nós da rede, fingindo ser quem ele não é. Como não nenhum tipo de autenticação nos protocolos de roteamento, não há como os outros nós saberem se o nó malicioso é ou não quem ele realmente diz ser.

Esse tipo de ataque é feito em conjunto com um ataque de Modificação ou de Fabricação. O nó malicioso poderia então reencaminhar uma mensagem de pedido ou resposta de rota, dizendo ser outro nó da rede, tentando assim criar *loops* na rede. Ou então poderia responder a pedidos de rota, informando ser o destino, e recebendo então as informações que deveriam ir para ele.

3.1.4. Ataques Cooperativos

Até agora foram apresentados vários tipos de ataque, com a ação de somente um nó malicioso. No entanto, vários nós maliciosos podem atuar em conjunto em uma rede, causando ataques cooperativos. Esse tipo de ataque é muito mais difícil de ser detectado, e até hoje nenhuma das soluções propostas para a segurança no roteamento das redes *ad hoc* trata completamente desse tipo de ataque.

Um exemplo desse tipo de ataque é o ataque *Wormhole* [Hu et al., 2001], onde um par de nós maliciosos na rede estão conectados por uma outra rede privada somente entre os dois. Então, um dos nós ao receber uma mensagem pode enviá-la pela conexão privada diretamente para o outro nó, que irá então reencaminhá-la na rede, sem que nenhum outro nó perceba. Esse ataque pode ser realizado para a análise do tráfego que passa na conexão privada entra os dois nós, ou para a utilização de rotas menores, atraindo então o tráfego para os nós maliciosos.

3.2. Como Evitar Esses Ataques

As soluções de segurança atualmente propostas podem utilizar vários mecanismos diferentes para garantir a segurança no roteamento das redes *ad hoc*. Com isso, eles conseguem evitar os principais tipos de ataques, os de Modificação, de Fabricação e de Personificação. Os ataques cooperativos ainda não podem ser totalmente evitados, e esse é um dos pontos falhos das atuais soluções de protocolos de roteamento seguros.

3.2.1. Chaves Simétricas

Um dos mecanismos utilizados para a autenticação, é a utilização de chaves secretas simétricas, que é um método que pode ser utilizado para o Ariadne. Assim, cada par de nós (origem e destino) que desejam se comunicar deve possuir uma chave secreta, para garantir uma associação de segurança. A dificuldade na utilização desse mecanismo é a distribuição das chaves secretas, que pode ser feita através de mecanismos de gerenciamento de chaves ou com chaves previamente combinadas. A utilização de chaves simétricas tem a vantagem de ser mais rápida computacionalmente, e não requerer muito processamento da estação.

3.2.2. Assinatura Digital - Chaves Assimétricas

Outro mecanismo utilizado é o de Assinatura Digital [Rivest et al., 1977], que é utilizado para autenticação no SAODV, e opcionalmente no Ariadne. A assinatura digital consiste na realização de um *Message Digest* da mensagem, utilizando-se a chave privada do nó que está assinando. Para a verificação da assinatura, utiliza-se a chave pública desse nó, assim todos os nós da rede podem verificar que o nó que enviou a mensagem é realmente quem ele diz ser, e somente o próprio nó pode gerar a assinatura digital, já que somente ele possui a sua própria chave privada. Para se utilizar esse método, cada nó deve possuir um par de chaves assimétricas, e deve também haver um mecanismo para distribuição de chaves, e certificação da associação de uma chave pública a um nó. Chaves assimétricas são muito mais seguras que chaves simétricas, pois somente as chaves públicas são divulgadas, mas no entanto, requerem um poder de processamento muito maior nas estações, o que nem sempre está disponível em estações móveis.

3.2.3. Hash Chains

As *hash chains* são utilizadas pelo SAODV para autenticação do campo de número de saltos, e também pelo Ariadne, em uma das suas opções para autenticação. Uma função *hash* [Bellare et al., 1996] é uma função que pode ter como entrada uma sequência de tamanho variável, e gera na saída uma sequência de tamanho fixo, por exemplo, 96 bits. Essa função é computacionalmente muito difícil de ser invertida, então, não é possível a partir do resultado da função *hash* chegar ao valor utilizado na sua entrada. Assim, o resultado de uma função *hash* corresponderá a um valor de entrada, e se este valor for alterado, a o resultado da saída também será diferente, com exceção das colisões, que podem ocorrer devido ao número de possibilidades na entrada ser muito maior que na saída. No entanto, com o exemplo de uma saída de 96 bits, teremos 2^{96} possibilidades na saída, e a probabilidade de ocorrer uma colisão é desprezível. Uma *hash chain* nada mais é então do que uma sequência, gerada a partir da aplicação sucessiva dessa função a uma semente, que pode ser um número gerado aleatoriamente. Quem possuir um dos valores da sequência, poderá garantir que os valores seguintes fazem parte da mesma sequência, aplicando a função novamente, mas não poderá a partir daí, chegar aos valores anteriores da sequência.

3.2.4. TESLA

O TESLA [Perrig et al., 2000] [Perrig et al., 2001] é uma outra opção de autenticação utilizada no Ariadne, que utiliza a técnica de *Hash Chains*. Esse mecanismo precisa que haja uma sincronização fraca entre os relógios dos nós da rede. Cada nó então gera a sua sequência, a partir de uma semente aleatória, e esses valores vão ser utilizados como chaves para a autenticação das mensagens. A estação divulga então o último valor gerado na sua sequência, e a partir daí usa essa sequência no sentido inverso da geração para autenticar suas mensagens. Ao enviar uma mensagem, a estação de origem então calcula o tempo médio que essa mensagem deve levar para chegar ao destino, e depois desse tempo, ela divulga a chave que foi utilizada para autenticar a mensagem. Assim, a estação de destino que recebeu a mensagem, receberá a chave após ter recebido a mensagem, e com

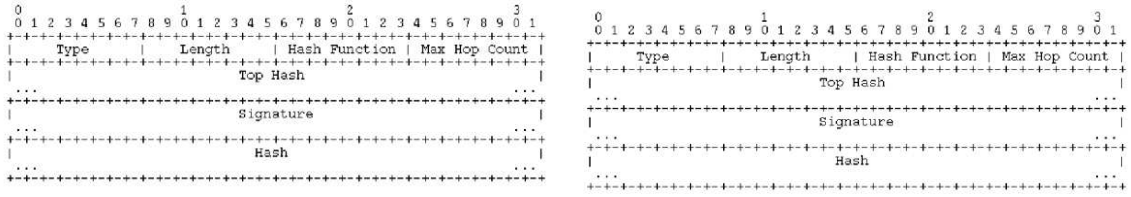
isso terá certeza que somente a estação de origem conhecia aquela chave para autenticar aquela mensagem. Para verificar que a chave está correta, ela aplica a mesma função de *hash* nessa chave, chegando no valor final da sequência que foi divulgado pela estação de origem, e assim por diante com as próximas chaves. Caso haja atraso no recebimento da mensagem, e a chave seja divulgada antes da estação de destino receber a mensagem, essa mensagem é então descartada.

4. SAODV

O SAODV é uma extensão ao protocolo AODV, que foi proposto para tentar solucionar o problema da segurança no roteamento nas redes móveis *ad hoc*. Seu objetivo é garantir requisitos de segurança como integridade, autenticação e não-repúdio ao mecanismo de descoberta de rota do AODV.

O SAODV se baseia na utilização de chaves assimétricas, onde cada nó da rede possui o seu par de chaves pública e privada. Inicialmente, assumimos que se dispõe de um sistema de gerenciamento de chaves na rede, onde cada nó é capaz de obter a chave pública de todos os outros nós da rede, e verificar seguramente que uma dada chave pública está associada a um certo nó da rede. Esse sistema pode ser implementado de diversas maneiras, dependendo do sistema de gerenciamento de chaves existente na rede.

A segurança é então garantida utilizando-se dois mecanismos diferentes, assinaturas digitais, e *hash chains*. As informações relativas aos *hash chains* e às assinaturas digitais são enviadas juntamente com as mensagens do AODV, que foram modificadas para conter campos para acomodar essas informações. As extensões às mensagens receberam o nome de ”Extensão de Assinatura”(*Signature Extension*), e seus formatos podem ser verificados na Figura 3.



(a) RREQ *Signature Extension*.

(b) RREP *Signature Extension*.

Figura 3: Extensão de Assinatura para as Mensagens do AODV.

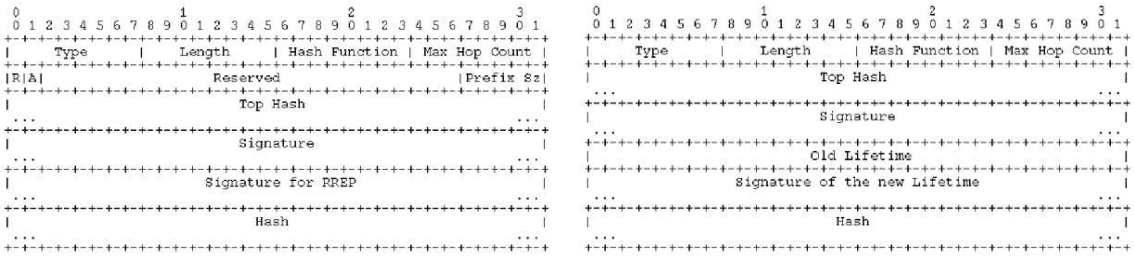
4.1. Assinatura Digital

A assinatura digital é utilizada para se autenticar e garantir a integridade dos campos imutáveis das mensagens de RREP e RREQ. Essa autenticação é feita fim-a-fim, entre a origem e o destino da mensagem de roteamento. Sempre que uma mensagem de RREP ou RREQ é criada, o originador da mensagem assina todo o conteúdo da mensagem, exceto o campo de Número de Saltos (*Hop Count*), e os campos de extensão do SAODV. Com isso, conseguimos garantir que não haverá alteração das partes imutáveis da mensagem de roteamento nos nós intermediários.

Um problema em se utilizar esse mecanismo, é o fato de o AODV permitir que nós intermediários respondam a um RREQ caso eles possuam uma rota atualizada para o destino. Essa técnica permite que o processo de descoberta de rota seja agilizado, tornando o protocolo mais eficiente. Contudo, ao responder a um pedido de rota, um nó intermediário não poderia assinar a mensagem de RREP pelo nó destino. Como a autenticação é feita fim-a-fim, o nó de origem ao verificar a assinatura do RREP não poderia validá-la com a chave pública do nó de destino, invalidando a rota.

O protocolo apresenta duas soluções para esse problema. Na primeira, apenas os nós de destino podem responder a pedidos de rota. Quando um nó intermediário já possui uma rota para o destino, ele simplesmente reenvia o RREQ, como se não possuísse essa rota, e ao reencaminhar o RREP ele atualiza a sua rota. Com essa solução, há uma perda de eficiência no protocolo, já que todos os pedidos de rota deverão ir até o destino para serem respondidos, mas no entanto, evita aumentar ainda mais a complexidade do protocolo com novos mecanismos para permitir que nós intermediários respondam ao RREQ.

A segunda solução consiste em, sempre que uma estação gerar um RREQ, ela incluir nesse pedido as informações de assinatura que permitirão que um nó intermediário responda a futuros pedidos de rota para esse mesmo destino. Então um nó intermediário, ao receber esse RREQ, armazena a rota reversa para a origem, a assinatura e o seu tempo de vida que serão utilizados caso ele venha a responder a um futuro pedido de rota para esse nó. Do mesmo modo, o destino ao enviar o RREP também enviará a informação de assinatura para que o nó intermediário responda a algum próximo pedido de rota para esse mesmo destino. Ao receber um RREQ de uma rota que ele já possui, o nó intermediário então responde a esse pedido, enviando no RREP a assinatura e o tempo de vida armazenados para aquela rota, e a sua assinatura e o tempo de vida atual, para garantir que esse nó intermediário é realmente quem ele diz ser. Para utilizar esse mecanismo foram criadas dois novos tipos de extensão, chamados de extensão de assinatura dupla, RREQ e RREP *Double Signature Extension*. A Figura 4 mostra os formatos dessas mensagens de extensão.



(a) RREQ *Double Signature Extension*.

(b) RREP *Double Signature Extension*.

Figura 4: Extensão de Assinatura Dupla para as Mensagens do AODV.

4.2. *Hash Chains*

O segundo mecanismo é do *hash chains*, que é utilizado para a autenticação do campo *Hop Count* das mensagens do SAODV. Essa autenticação passa a ser feita não somente

fim-a-fim, mas em todos os nós intermediários, garantindo que o vetor de distância não será alterado.

Quando um nó envia uma mensagem RREQ ou RREP, ele gera um número aleatório que será a semente da *hash chain*, e seleciona um número máximo de saltos (*Maximum Hop Count*), que deve ser configurado para o valor do campo TTL do cabeçalho IP. Ele coloca então essa semente no campo *Hash*, e aplica a função de *hash* um número de vezes igual ao número máximo de saltos nessa semente, colocando o valor encontrado no campo *Top Hash*.

Um nó intermediário, ao receber um RREQ ou RREP, aplica a mesma função um número de vezes igual a (*Max Hop Count* menos *Hop Count*) vezes no valor do campo *Hash*, e o compara com o valor de *Top Hash* para autenticar o campo do número de saltos, descartando o pacote em caso de os dois não coincidirem. Então, o nó intermediário incrementa o número de saltos em 1, e aplica a função de *hash* no campo *Hash* da mensagem. Com isso, um nó intermediário não é capaz de diminuir o número de saltos da mensagem de roteamento, já que a função *hash* é computacionalmente muito difícil de ser invertida, impedindo que se obtenha os valores anteriores da sequência.

Contudo, um certo tipo de ataque ainda se torna possível, já que um nó malicioso, apesar de não poder diminuir o valor do número de saltos, poderia simplesmente reencaminhar a mensagem sem alterar esse valor, diminuindo assim em 1 o valor total do número de saltos. Entretanto, a eficiência desse tipo de ataque é limitada.

4.3. Adaptações ao AODV

Uma outra adaptação que deverá ser feita em relação ao AODV original se refere aos números de sequência do destino (*destination_sequence_number*). De acordo com o AODV, o originador de um RREQ poderia colocar um valor muito maior no número de sequência do destino, permitindo então ataques de número de sequência que atrairiam mais tráfego para o nó malicioso, por suas rotas parecerem mais atualizadas. Isso acontece porque ao ser reiniciado, um nó não se lembra do seu número de sequência, e logo, não confia em ninguém que o mande uma mensagem com um certo número de sequência, já que essa mensagem poderia vir de um nó malicioso.

Por isso, o SAODV propõe que todos os nós deveriam possuir um meio de armazenar os seus números de sequência, mesmo quando são reiniciados. Então, no caso de um nó receber um número de sequência em um RREQ maior que o seu número, ele desconsideraria o número de sequência recebido. Nesse caso ele consideraria que o nó originador da mensagem está agindo maliciosamente, e enviaria de volta um RREP com o seu número de sequência correto. Com isso, evita-se também os ataques de número de sequência.

4.4. Mensagens de Erro - RERR

As mensagens de erro possuem um tratamento ligeiramente diferente das mensagens de RREQ e RREP. Nessas mensagens há muita informação variável, além de somente o campo *Hop Count*, e o que se deseja assegurar nas mensagens de RERR é simplesmente que o nó que está enviando a mensagem é quem ele diz ser.

Então, nesse caso a proposta é que, cada nó, gerando ou encaminhando uma mensagem de RERR, irá assinar essa mensagem, que será então verificada pelo próximo nó.

Assim, o próximo nó pode verificar que quem esta mandando o RERR é realmente quem ele diz ser.

5. Ariadne

O Ariadne é um protocolo de roteamento seguro para redes móveis *ad hoc*, baseado no protocolo DSR. Ele possui alguns pré-requisitos para ser utilizado, como a necessidade de que haja uma sincronização fraca de tempo entre os nós da rede, de modo de um nó possa estimar o tempo de transmissão fim-a-fim para qualquer outro nó da rede. Também é necessário um mecanismo para o estabelecimento de chaves secretas entre os nós comunicantes, e um meio de fazer a distribuição de uma chave pública TESLA autêntica para cada nó.

Três mecanismos principais de segurança compõem o Ariadne: um mecanismo para verificar a autenticidade de uma mensagem de RREQ pelo destino, feita fim-a-fim; três alternativas para um mecanismo de autenticação dos dados nas mensagens de RREQ e RREP, que é feita em todos os nós intermediários; e um mecanismo para garantir a integridade da lista dos nós da rota em um pedido.

5.1. Autenticação Fim-a-Fim

Para garantir ao destino a legitimidade da mensagem de RREQ, o nó de origem utiliza um esquema de chaves simétricas. Ele inclui um MAC (*Message Authentication Code)* na mensagem, computado com a chave simétrica que somente os nós de origem e destino possuem. Assim, o nó destino pode ter certeza que as informações do RREQ estão corretas, e a mensagem veio realmente deste nó de origem.

5.2. Autenticação dos Nós Intermediários

Assim como no DSR, cada nó intermediário vai atualizando a mensagem de RREQ, adicionando e alterando as informações necessárias. Para garantir que um nó malicioso não possa alterar indevidamente uma mensagem de RREQ, é necessário um mecanismo que para autenticar os dados da mensagem nó a nó. O Ariadne possui três possíveis técnicas para realizar essa autenticação: a utilização de um MAC com chaves secretas; assinaturas digitais, utilizando chaves assimétricas; ou a utilização do esquema de autenticação TESLA.

Utilizando chaves secretas, a autenticação funcionaria do mesmo modo que a autenticação fim-a-fim, somente que agora entre todos os nós intermediários da rota. A consequência da utilização dessa técnica é que serão necessários pares de chaves secretas entre todos os nós da rede, e um modo de distribuição dessas chaves, o que torna essa opção inviável. A opção de utilização de assinaturas digitais com chaves assimétricas funcionaria do mesmo modo que no SAODV, onde cada nó assinaria uma mensagem com a sua chave pública, e o nó seguinte verificaria a sua assinatura. O problema ligado à utilização de assinatura assimétrica é o alto poder de processamento requerido para a sua verificação. O Ariadne se propõe a ser utilizado por qualquer tipo de dispositivo móvel, e nós em uma rede *ad hoc* podem não possuir recursos suficiente para essa verificação. Isso permitiria inclusive um ataque de inundação com assinaturas inválidas a nós com menor poder de processamento, já que essa verificação exaustiva seria muito cara para certos dispositivos.

A terceira opção seria a utilização do TESLA. Apesar de apresentar as duas opções anteriores, o autor da proposta sugere que somente esse mecanismo seja utilizado, devido às dificuldades apresentadas pelos outros dois mecanismos. Quando o TESLA é utilizado, cada nó autentica as suas mensagens de RREQ com as suas chaves TESLA atuais. Então, o nó destino armazena a mensagem de RREP até que os nós intermediários divulguem as suas chaves TESLA. Com a condição de segurança verificada, o nó destino calcula o MAC para a mensagem de RREP, e a envia de volta para a origem.

Ao enviar uma mensagem de RREQ, o nó de origem deve estimar um tempo T para o máximo atraso fim-a-fim para aquela mensagem. A escolha errada de um valor de T não irá afetar a segurança do protocolo, já que caso uma chave seja divulgada antes do seu pacote correspondente ser autenticada, a condição de segurança não é verificada, e o nó destino não envia a mensagem de RREP. Valores muito pequenos de T irão causar então a falha na descoberta de rota. O Ariadne poderá então escolher valores de T adaptativamente, aumentando esse valor sempre que uma descoberta de rota falhar, e o nó destino poderia informar ao nó de origem sempre que o valor escolhido para T for muito longo.

5.3. Integridade da Lista de Nós

Para garantir a integridade da lista de nós da rota na mensagem de RREQ, é utilizado um mecanismo de *hash chains*, idêntico ao utilizado no SAODV. Assim, sempre que um nó adicionar o seu endereço na lista de nós, ele irá calcular um novo *hash* para garantir que a lista não será alterada.

No Ariadne, esse mecanismo não possui o problema encontrado no SAODV, onde um nó poderia simplesmente não calcular um novo *hash*, diminuindo o número de saltos da rota em 1. Como no DSR os nós intermediários são informados explicitamente na mensagem de RREQ, o nó intermediário sempre deverá adicionar o seu próprio endereço na lista de nós da rota, o que o impede de diminuir o número de saltos da rota. Caso ele tente remover algum dos nós anteriores, o resultado da função *hash* será diferente, denunciando o ataque.

6. Conclusões

Neste trabalho apresentamos os problemas de segurança existentes atualmente no que diz respeito ao roteamento em redes móveis *ad hoc*. Os protocolos de roteamento atuais apresentam muitas deficiências no aspecto de segurança, e é fundamental que se desenvolva novas propostas com a utilização de mecanismos de segurança que solucionem as vulnerabilidades existentes atualmente.

Apresentamos também algum dos tipos de ataques que podem ser realizados à esses protocolos de roteamento, e alguns dos mecanismos que podem ser utilizados para se tornar os protocolos atuais mais seguros, como chaves simétricas e assimétricas, assinaturas digitais, *hash chains*, e o esquema de autenticação TESLA.

Entre as soluções propostas atualmente, apresentamos o SAODV e o AODV, que são adaptações seguras dos protocolo AODV e DSR, respectivamente. Ambos são capazes de prevenir os três tipos de ataques apresentados nas seções 3.1.1, 3.1.2 e 3.1.3, mas ainda não possuem nenhum mecanismo que impeça a realização de um ataque do tipo

Wormhole, conforme apresentado na seção 3.1.4, abrindo então uma necessidade de novas propostas mais completas que venham a melhorar as soluções de segurança existentes atualmente para o roteamento em redes móveis *ad hoc*.

Referências

Bellare, M., Canetti, R., and Krawczyk, H. (1996). Keying hash functions for message authentication. *Lecture Notes in Computer Science*, 1109:1–??

Bellman, R. (1958). On a Routing Problem. In *Quaterly of Applied Mathematics*, volume 16, pages 87–90.

Clausen, T., Jacquet, P., Laouiti, A., Muhlethaler, P., a. Qayyum, and Viennot, L. (2001). Optimized link state routing protocol. In *IEEE INMIC Pakistan*. Best paper award.

Ford, L. and Fulkerson, D. (1962). *Flows in Networks*. Princeton University Press.

Guerrero Zapata, M. and Asokan, N. (2002). Securing Ad hoc Routing Protocols. In *Proceedings of the 2002 ACM Workshop on Wireless Security (WiSe 2002)*, pages 1–10.

Hu, Y., Perrig, A., and Johnson, D. (2001). Packet leashes: A defense against wormhole attacks in wireless ad hoc networks.

Hu, Y., Perrig, A., and Johnson, D. (2002a). Ariadne: A secure on-demand routing protocol for ad hoc networks.

Hu, Y.-C., Johnson, D. B., and Perrig, A. (2002b). Sead: Secure efficient distance vector routing in mobile wireless ad hoc networks. In *Fourth IEEE Workshop on Mobile Computing Systems and Applications* (WMCSA '02), pages 3–13.

Johnson, D. B. and Maltz, D. A. (1996). Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers.

Papadimitratos, P. and Haas, Z. J. (2002). Secure routing for mobile ad hoc networks. In *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*.

Perkins, C. and Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244.

Perkins, C. and Royer, E. (1999). Ad-hoc on-demand distance vector routing. In *2nd IEEE Workshop on Mobile Computing Systems and Applications*.

Perrig, A., Canetti, R., Song, D., and Tygar, D. (2001). Efficient and secure source authentication for multicast. In *Network and Distributed SystemSecurity Symposium, NDSS'01*.

Perrig, A., Canetti, R., Tygar, J. D., and Song, D. X. (2000). Efficient authentication and signing of multicast streams over lossy channels. In *IEEE Symposium on Security and Privacy*, pages 56–73.

Rivest, R. L., Shamir, A., and Adelman, L. M. (1977). A METHOD FOR OBTAINING DIGITAL SIGNATURES AND PUBLIC-KEY CRYPTOSYSTEMS. Technical Report MIT/LCS/TM-82.

Royer, E. M. and Toh, C.-K. (1999). A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. *IEEE Personal Communications*, 6(2):46–55.

Sanzgiri, K., Dahill, B., Levine, B. N., Shields, C., and Belding-Royer, E. M. (2002). A secure routing protocol for ad hoc networks.

Wang, W. and Bhargava, B. (2002). On vulnerability and protection of ad hoc on-demand distance vector protocol. Technical report, Technical report, TR-2002-18, CERIAS Security Research Center, Purdue University.

Wang, W., Lu, Y., and Bhargava, B. (2002). On security study of two distance-vector routing protocols for mobile ad hoc networks. Technical report, Technical report, Dept. of Computer Sciences, Purdue University.

Zhou, L. and Haas, Z. J. (1999). Securing ad hoc networks. *IEEE Network*, 13(6):24–30.