Protegendo Redes Ad Hoc com Certificados Digitais: Uma Proposta de Arquitetura

Wagner Gaspar Brazil e Célio Vinícius Neves de Albuquerque

Instituto de Computação - Universidade Federal Fluminense (IC – UFF)
Rua Passo da Pátria 156, Bloco E - 3º andar, São Domingos
CEP: 24.210-240 -Niterói -RJ -Brazil

{wbrazil,celio}@ic.uff.br

Abstract: In this paper we propose a PKI architecture with high availability and resilience to mitigate the Black Hole and Spoofing attacks against ad hoc networks. The architecture is designed to reduce the amount of control messages exchanged, increasing its scalability and performance. In order to reach this goal, we propose that the various nodes within an ad-hoc network use a distributed digital certification service to authenticate and encrypt its messages. Particularly, the media access control and the network protocols must use the authentication and cryptography to protect the network from the above-mentioned attacks, granting confidentiality, authenticity and integrity when exchanging messages. Results show a decrease of up to 85% in the number of messages of the renovation certificates protocol compared to existing approaches.

Resumo: Este trabalho propõe uma arquitetura de ICP de alta disponibilidade e robusta de forma a minimizar os ataques do tipo buraco negro e falsificação de identidade contra redes ad hoc. A arquitetura se propõe a trocar um número de mensagens de controle reduzido aumentando assim a sua escalabilidade e desempenho. Para atingirmos este objetivo propomos que os diversos nós de uma rede ad hoc utilizem o serviço distribuído de certificação digital para autenticar e cifrar suas mensagens. Particularmente os protocolos de acesso ao meio e os de roteamento devem usar a autenticação e cifragem para se protegerem dos ataques citados acima garantindo assim confidencialidade, autenticidade e integridade na troca de mensagens. Resultados mostram uma redução de até 85% no número de mensagens no protocolo de renovação de certificados comparados com alguns métodos existentes.

1 - Introdução

Os ataques de negação de serviços tipo buraco negro e falsificação de identidade estão entre as maiores ameaças à segurança de redes sem fio e sem infra-estrutura fixa, redes ad hoc. Tais ataques ocorrem também em redes com fio, porém a arquitetura sem fio e sem infra-estrutura torna tais ataques mais difíceis de serem detectados e de serem contidos.

Serviços de infra-estrutura de chaves públicas (*ICP)* têm sido utilizados para garantir, em diversos níveis, confidencialidade, integridade e também garantia de não repúdio em diversas aplicações. Desta forma propomos utilizar um serviço de *ICP* conforme recomendado em [Haas and Zhou 1999] para autenticar e cifrar, quando necessário, as mensagens dos protocolos de acesso ao meio e de roteamento de uma rede ad hoc. Este trabalho se propõe a melhorar as propostas dos trabalhos [*Seung and Kravets* 2001], [Kong et al. 2001] e [Luo et al. 2002] utilizando cifragem por limiar para dividir a chave privada do serviço de *ICP* entre diversas autoridades certificadoras (AC) em um esquema de certificação

cruzada. Baseado no trabalho de [Haas and Zhou 1999], este trabalho propõe um protocolo que diminua o número de mensagens trocadas entre as ACs e entre os nós e as ACs.

Com a utilização destas técnicas que garantem autenticidade e confidencialidade podemos evitar que nós maliciosos participem das comunicações, mitigando os efeitos dos ataques de buraco negro e de falsificação de identidade.

1.1 - O ataque buraco negro

Este tipo de ataque se caracteriza quando um ou vários nós da rede deliberadamente descartam os pacotes que passam por eles após o estabelecimento da rota. O buraco negro pode também funcionar como um espião, apenas copiando os pacotes para uma base de dados interna sem descartá-los.

É importante ressaltar que o mau comportamento do nó atacante só começa após o estabelecimento das rotas. O nó atacante participa normalmente do protocolo de estabelecimento de rotas. Por esta razão, é importante que os nós de uma rede tenham confiança e autenticação entre si antes de estabelecerem suas rotas. Para a segunda variante do ataque, espionagem, o uso adicional de cifragem fim a fim dificulta a quebra de confidencialidade.

1.2 - O ataque de falsificação de identidade

Em um ataque de falsificação de identidade, um nó se anuncia como sendo outro para receber as mensagens endereçadas ao nó atacado. São conhecidos vários tipos de ataques de falsificação de identidade e eles podem ser lançados nos diversos níveis da arquitetura de comunicação.

Como nossa preocupação principal é proteger de ataques nos níveis de enlace e principalmente no nível de rede, o uso de *ICP* previne este tipo de ataque, pois é praticamente impossível que um nó possa se passar por outro quando cada nó tem seu par de chaves pública/privada renovado periodicamente. Estas chaves servem para dar garantia de autenticidade e de não repúdio nas comunicações entre os nós.

1.3 - Características de ataques em redes ad hoc

Redes ad hoc são suscetíveis a ataques que vão desde ataques passivos com falsificação de endereços (ou falsificação de identidade*)* e descarte de pacotes (buraco negro) até ataques em que o atacante ativamente injeta mensagens de erros na rede e falsas mensagens de roteamento. Outros ataques conhecidos em redes ad hoc são o *Jelly-Fish* e o *Sybill.*

O *Sybill* ocorre quando uma AC não autorizada, por exemplo, consegue gerar certificados para os nós da rede [Douceur 2002]. Já o *Jelly-Fish* [Aad et al. 2004] explora vulnerabilidades do algoritmo de controle de congestionamento do TCP e age de três maneiras: desordenando pacotes, fazendo um descarte periódico de pacotes em um determinado tempo e atrasando randomicamente os pacotes. O objetivo é reduzir o *goodput* de todos os fluxos TCP para próximo de zero.

Além destes, vários outros ataques das redes com fio podem ser observados em uma rede ad hoc. Como exemplo, podemos citar o *Man-in-the-midle, Playback attack, Denial of Service (DOS)* e *Distributed Denial of Service* (*DDOS)*. No *Man-in-the-midle* o atacante se interpõe na comunicação entre as partes capturando os dados de ambos os lados, no *Playback-attack* o atacante captura algum dado de autenticação e tenta usá-lo novamente para uma nova autenticação no sistema, *DOS* e *DDOS* são ataques que visam atingir a disponibilidade de algum nó ou serviço fazendo que o mesmo não consiga responder a requisições.

Neste ambiente hostil e com proteção fraca, é altamente recomendado que a arquitetura da rede seja distribuída. A introdução de qualquer ponto central de controle é uma vulnerabilidade pronta para ser explorada. Se o ponto central é comprometido, toda a rede é comprometida.

Devido à mobilidade dentro de uma rede ad hoc, a entrada e saída de nós podem ocorrer de maneira rápida e por isso nenhuma solução de segurança que possua uma configuração estática pode ser efetiva neste ambiente.

Por outro lado, uma rede ad hoc possui vários nós possivelmente no alcance uns dos outros e por isso há potencialmente vários caminhos ou rotas entre dois nós. Esta característica pode e deve ser aproveitada para aumentar tanto o desempenho da rede usando seus múltiplos caminhos [Villela and Duarte 2004] quanto à sua segurança. Qualquer solução de segurança deve ser escalável para suportar redes pequenas e grandes, densas ou esparsas. Esta solução deve também se aproveitar da característica de diversos caminhos e nós na rede para prover uma solução de alta disponibilidade e robusta a ataques.

Este trabalho está organizado como segue. A seção 2 descreve o uso de sistemas baseados em ICP para proteção de redes nas diversas camadas do modelo OSI. Na seção 3 são apresentados os trabalhos relacionados mais importantes nos quais nos baseamos para propor melhoramentos. A seção 4 descreve brevemente uma Infra-Estrutura *de chaves públicas* e os papéis que os processos e computadores podem assumir nesta infra-estrutura. A seção 5 apresenta a proposta do trabalho, algoritmos, cenário de implementação e protocolos da proposta. A seção 6 descreve o ambiente de simulação, análise de troca do número de mensagens, análise de disponibilidade, overhead dos protocolos e consumo de energia. Finalmente na seção 7 são apresentadas as conclusões do trabalho.

2 - Proteção nas diversas camadas

O esquema do uso de *ICP* pode ser usado para proteger as diversas camadas da arquitetura de redes conforme dito anteriormente.

No nível físico, técnicas como *Spread Spectrum* já são usadas para dar maior robustez e confiabilidade nas comunicações.

No nível de enlace, protocolos que provêem autenticação e cifragem podem usar do modelo de *ICP* proposto para evitarem ataques de spoofing de endereços MAC. Também no nível de enlace, se um nó não pode nem começar a se comunicar, não poderá estabelecer mais tarde rotas de nível de rede. Isso impede também a entrada de nós não confiáveis que se comportem como buracos negros.

No nível de aplicação assinatura digital e cifragem também são amplamente usadas para garantir a confidencialidade e autenticidade nas comunicações fim a fim.

3 - Trabalhos relacionados

Em [*Seung and Kravets* 2001] os autores usam *broadcast* para tentar contatar todas as *CAs* quando de um pedido de certificado (pacote CREQ) usando o esquema de cifragem por limiar. Através de simulações é mostrado que cada nó recebe em média $(2/3 \times N)$ respostas, sendo N o número de ACs. Esta técnica de flooding no pedido de certificados gera mais pacotes de controle do que o necessário e os próprios autores colocam o seguinte cenário:

Para 1000 requisições de certificados com 30 ACs, são gerados 119125 pacotes CREQ quando na verdade o número mínimo de pacotes deste tipo seria 1000 x T (limite criptográfico). Com T=10, por exemplo, o número de pacotes CREQ mínimo seria de 10000, bem inferior aos 119125 pacotes gerados.

Em [Kong et al. 2001] e [Luo et al. 2002] é apresentado um modelo de ICP com cifragem por limiar, mas não é especificado como certificados são guardados em cada nó. A CRL (Certificate Revocation List) é construída progressivamente através de contadores de certificados que são propagados por flooding quando da sua assinatura. Em nosso trabalho propomos um modelo de certificação cruzada que guarda em cada AC a lista de certificados revogados e uma lista de ACs confiáveis em cada nó e em cada AC. A atualização das CRL's de cada AC é feita por comunicação unicast (em oposição ao flooding) entre as ACs assim que um certificado é revogado.

A proposta em [Kong et al. 2001] e [Luo et al. 2002] é baseada em um esquema de serviço único de ACs distribuídas. Nossa proposta pode acomodar múltiplos serviços de AC distribuída bastando para isso, em um futuro trabalho, estabelecer o protocolo de controle e manutenção de confiança entre as ACs de diferentes serviços.

Em [Haas and Zhou 1999], a técnica de cifragem por limiar é usada na ICP e é proposto pelos autores o uso de um servidor denominado "combinador", cuja função é computar a assinatura para os certificados gerados pela infra-estrutura. Em nosso trabalho qualquer servidor da ICP pode fazer o papel de AC combinadora aumentando assim a disponibilidade do serviço.

4 - Infra-Estrutura de chaves públicas

Infra-estrutura *de chaves públicas* tem sido usada largamente para garantir confidencialidade, integridade e garantia de não repúdio em transações de diversas áreas, notadamente em comércio eletrônico. Este esquema é baseado em cifragem assimétrica onde cada usuário possui uma chave pública e uma chave privada. A chave pública é conhecida de todos e a chave privada fica com o usuário. Através de operações matemáticas tudo que é cifrado com a chave pública é decifrado com a chave privada e vice-versa.

A infra-estrutura é composta minimamente dos seguintes papéis:

- Autoridade Certificadora – É a entidade que gera os certificados e faz a verificação da validade dos mesmos através de sua chave pública e da Lista de Certificados Revogados;
- Autoridade Registradora – É a entidade que faz o cadastro dos usuários do serviço de certificação;
- Lista de certificados revogados (CRL) - Uma lista dos certificados que foram revogados e devem ser considerados fora de uso;
- Serviço de diretório - Serviço normalmente implantado em estrutura hierárquica que contém os usuários cadastrados e seus certificados;
- Usuários do serviço - Qualquer entidade que utilize a infra-estrutura;

Um nó pode assumir mais de um papel. Em um esquema de certificação cruzada, uma autoridade certificadora confia nos certificados gerados por outra autoridade certificadora e vice-versa.

5 - Proposta

A proposta deste trabalho consiste em um esquema com "N" ACs trabalhando com certificação cruzada. Neste modelo cada AC confia nos certificados emitidos pelas outras ACs que estão na sua lista de ACs confiáveis.

Cada AC possui seu par de chaves (pública e privada) e o serviço de certificação possui também seu par: a chave pública é conhecida de todos e a chave privada é dividida

entre as "N" ACs. Para que uma requisição de certificado seja processada pelo serviço é preciso que pelo menos T (nosso limite criptográfico [Shamir 1979]) partes desta chave privada para reconstruir a chave do serviço como um todo. Este limite pode variar e é configurado na inicialização das ACs. Quanto maior o limite mais confiável e seguro o serviço, porém menos tolerante a falhas, pois mais ACs deverão fornecer sua parte da chave privada corretamente.

Quando usamos a técnica de cifragem por limiar conforme detalhada em [Desmedt 1994, Haas and Zhou 1999], é preciso nos proteger contra servidores comprometidos. Por exemplo: um servidor (AC) comprometido poderia gerar uma assinatura parcial incorreta e o uso da mesma geraria uma assinatura inválida. Felizmente podemos verificar a validade de uma assinatura usando a chave pública do serviço. No caso de falha da verificação tenta-se uma outra combinação de assinaturas parciais. Este processo continua até que a AC combinadora consiga gerar a assinatura correta para a chave pública do serviço.

5.1 - O cenário de implementação

Em um primeiro cenário para este trabalho, as *CAs* são fixas e têm posições pré-estabelecidas dentro da rede. Também é considerado apenas um serviço de certificação e não vários pares de chaves pública/privada para vários serviços. As *CAs* estabelecem suas rotas com as *CAs* vizinhas usando o mesmo protocolo de roteamento dos outros nós da rede. Para estabelecer esta comunicação (Figura 1) as *CAs* utilizam suas chaves públicas/privadas para promover autenticação. Depois trocam uma chave simétrica de sessão para cifragem das mensagens de nível de enlace e de nível de rede. Esta autenticação e troca de chaves de sessão são feitas usando um protocolo de comunicação entre os nós antes do estabelecimento das rotas.

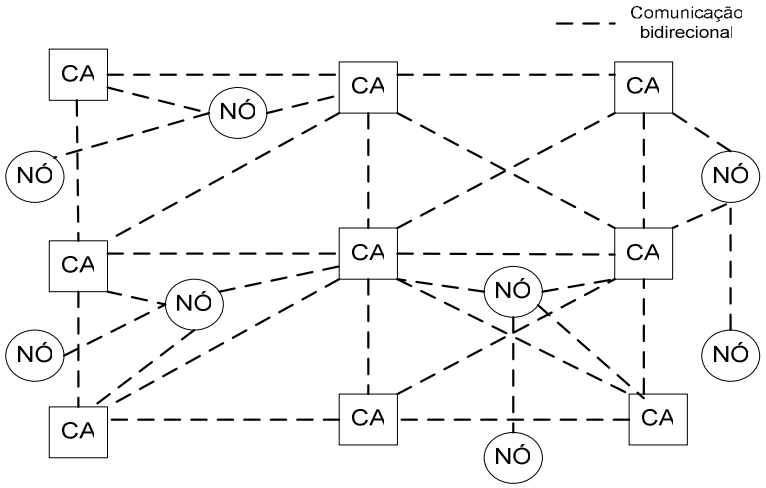


Figura 1: Exemplo de arquitetura proposta

Além de um algoritmo de cifragem de chave simétrica, que visa dar maior desempenho, é usado um esquema de troca de números de seqüência aleatórios na comunicação (*nonce)* visando proteção contra ataques do tipo *replay*.

A política padrão de requisição de novos certificados foi estabelecida como "NEGADA", ou seja, nenhum novo certificado pode ser requerido e a política de renovação de certificados como "ACEITA", ou seja, as *CAs* aceitam pedidos de renovação de certificados já gerados por elas no esquema de certificação.

5.2 - Inicialização das ACs e distribuição dos certificados

Visando dar maior confiabilidade e para diminuir a complexidade, as *CAs* e os nós recebem de maneira "off-line" (fora da rede) seus certificados iniciais. Também de maneira off-line é gerado o par de chaves do serviço de certificação e a chave privada do serviço é então dividida entre as ACs do sistema. Este procedimento tenta garantir que o modelo com ACs descentralizadas apresente as mesmas características que uma AC centralizada, evitando ataques de ACs externas que não pertençam ao conjunto ou que foram comprometidas [Douceur 2002] (*Sybill attack)*.

5.3 - Listas de certificados revogados e de ACs Confiáveis

A lista de certificados revogados é mantida replicada em cada AC. Propomos a criação e manutenção de uma lista de *CAs* "confiáveis" do sistema que é mantida em cada AC e em cada nó. Na inicialização das *CAs* a lista contém todas as *CAs* do sistema.

Quando da geração e instalação do certificado digital no nó, é gravada a lista de *CAs* confiáveis que servirá para o nó poder aceitar ou não certificados de *CAs* que porventura tenham sido comprometidas ou que estejam fora de serviço.

Existem duas maneiras de uma AC sair da lista de confiáveis. A primeira ocorre quando a AC não é alcançável e fica fora da tabela de roteamento das outras *CAs*. Não faz parte do escopo deste trabalho descrever as medidas que protegem o sistema de um ataque tipo DOS ou DDOS. O sistema deve prever medidas para estancar e prevenir estes ataques para que o mesmo não fique indisponível, porém este não é o foco deste trabalho.

A segunda ocorre quando uma AC envia a sua parte da chave privada do serviço de certificação corrompida ou errada, sendo comprovada esta falha por pelo menos duas ACs. Como veremos adiante, a comunicação entre as ACs é cifrada evitando assim que um intruso capture pacotes, altere-os e faça uma reinserção dos mesmos na rede.

Nas duas situações acima a AC sai da lista de confiáveis e fica fora de serviço até que o administrador faça alguma ação para recuperá-la. A AC que fez a verificação e removeu uma outra AC da lista de confiáveis, imediatamente altera sua lista e a envia por *unicast* para as outras ACs confiáveis da lista. A lista de ACs confiáveis é verificada a cada acesso a uma AC através da comparação do campo de *"aging*" da lista do nó com o da lista da AC.

5.4 - Renovação de certificados

A chave do serviço deve ser trocada em um período igual ou inferior ao da troca dos certificados dos nós quando então deve ser feita uma nova inicialização dos serviços. Desta maneira estaremos promovendo a renovação da chave do limiar criptográfico.

Para a renovação do seu certificado, o nó adota o procedimento descrito na Figura 2 que é explicado abaixo:

1. O nó envia uma mensagem de "Req" para $\lceil N / \text{Log } 2T \rceil$ ACs confiáveis (N é o número de ACs e T é o limite criptográfico). Vamos levar em consideração que o nó só precisa contatar uma AC e que em [Seung and Kravets 2001] cada nó recebe $2/3 \times N$ mensagens quando usa a técnica de *broadcast*. Vamos usar a função $S=\lceil N / \text{Log } 2T \rceil$ para diminuir a quantidade de mensagens trocadas achando um valor entre 1 e N que será o número de ACs contatadas. Poderíamos usar outra função "S(T)" contanto que $2/3 \times S(T) \geq 1$. Se não receber nenhuma mensagem de "reply" o nó escolhe outras $\lceil N / \text{Log } 2T \rceil$ ACs até conseguir comunicação ou até se esgotarem as ACs da sua lista de ACs confiáveis.

2. As ACs enviam um “reply” para o nó.
3. O nó recebe no máximo “n” respostas e escolhe como AC combinadora da chave do serviço a primeira que lhe respondeu lhe enviando um Ack e seu certificado para renovação.
4. A AC combinadora requisita (“key request”) então as partes da chave privada do serviço para outras C= (Log(T)+T-1) ACs. A função “C” deve sempre retornar valores maiores que “T” (limite criptográfico) para compensar possíveis perdas de mensagens enviadas para as outras ACs.
5. As ACs enviam suas partes da chave privada para a AC combinadora (“key sent”).
6. A AC combinadora faz então a verificação das partes da chave privada do serviço e renova o certificado digital do nó que recebe um novo par de chaves.

Com este protocolo evitamos o *flooding* de *broadcast* como em [*Seung and Kravets* 2001] e o número de mensagens trocadas no processo de certificação diminui drasticamente. É certo que no passo um existe uma possibilidade, já descrita, de que nenhuma AC responda ao pedido do nó. Isso implicaria em uma segunda rodada do passo um para selecionar outras ACs o que poderia levar a um maior tempo de processamento e espera do que na opção com uso de *broadcast*, porém em nossas simulações isso não ocorreu.

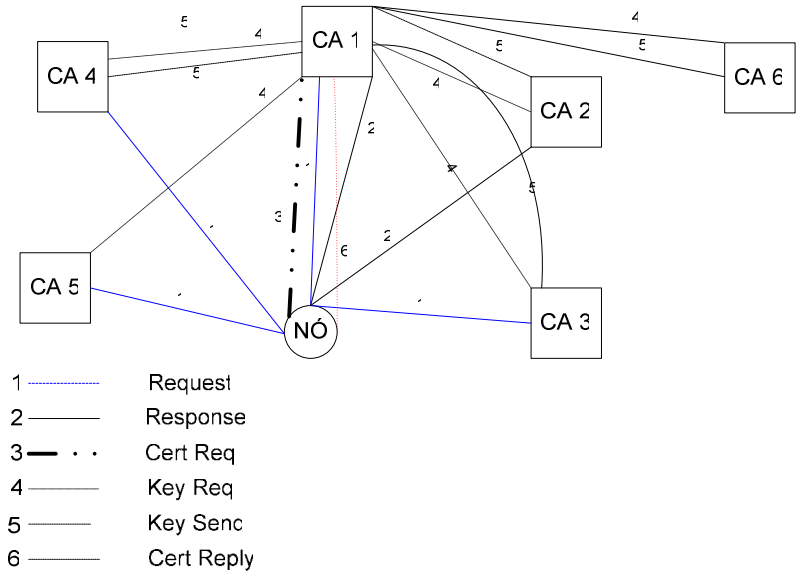


Figura 2: Protocolo de renovação de certificados

5.5 - Funcionamento do protocolo de autenticação

Sempre que um nó quiser estabelecer comunicação com outro, deverá haver autenticação mútua no processo de estabelecimento de rotas. Assim, um nó de origem N1 enviará para cada nó do caminho até um nó destino N2 a requisição de rota usando um algoritmo de acesso ao meio e de roteamento de forma segura e autenticada. Por sua vez o nó N2 fará o mesmo no estabelecimento do caminho reverso.

Estas autenticações ocorrem apenas no estabelecimento das rotas entre dois nós quaisquer N1 e N2 e é feita com a chave privada de cada nó do caminho que é verificada com sua chave pública.Adicionalmente poderia ser enviado junto com a requisição de rota (ROUTE REQUEST) um *nonce* para evitar ataques de repetição. A cada autenticação os nós

devem verificar se os certificados estão válidos checando a CRL em uma AC confiável usando a função $S= \lceil N / \text{Log } \overline{2T} \rceil$.

Após a autenticação, o receptor gera um número aleatório (*nonce*) baseado no tempo decorrido desde o último "*reboot",* uma chave simétrica com um algoritmo do tipo AES-128 bits ou DES e o pacote de ROUTE REPLY com a resposta ao ROUTE REQUEST. Os algoritmos de chave simétrica podem ter um tamanho de chave menor porque estas chaves serão regeradas em um intervalo de tempo menor (a cada sessão). O receptor envia o ROUTE REPLY, a chave simétrica e o *nonce* cifrados com a chave pública do emissor. O emissor então decodifica os parâmetros com sua chave privada. O emissor então responde com um reconhecimento (*ACK*) indicando que aceita a comunicação com o número de seqüência e a chave simétrica escolhida. A partir daí as trocas de mensagens entre os nós se darão usando a chave simétrica acordada. Este protocolo é detalhado de forma gráfica na Figura 3.

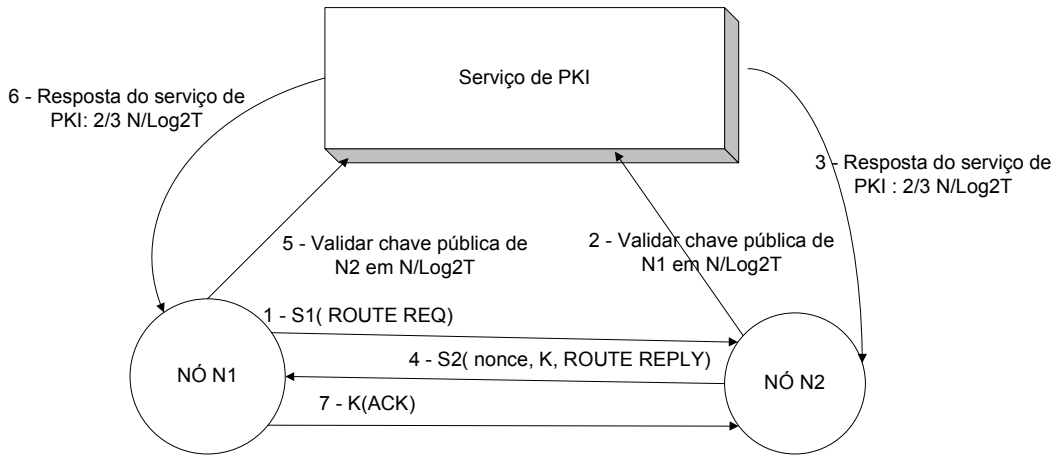


Figura 3: Protocolo de autenticação dos nós

Com o uso dos protocolos descritos no cenário proposto, consegue-se mitigar os efeitos dos ataques buraco negro no nível de rede e de falsificação de identidade nos níveis de enlace e de rede. O atacante pode se anunciar como um nó válido no ataque de falsificação de identidade, porém não consegue obter sucesso na comunicação, visto que não consegue estabelecer comunicação no plano de enlace nem no plano de rotas devido à impossibilidade de se autenticar usando o esquema de ICP proposto.

Com o atacante usando o buraco negro ocorre o mesmo. O atacante não consegue apresentar o mau comportamento descrito na introdução deste artigo nem descartar pacotes visto que estes comportamentos só ocorrem após o estabelecimento das rotas. Como o atacante não consegue se autenticar corretamente para estabelecer as rotas, o ataque é frustrado.

Defender uma estação de uma contaminação por algum vírus ou cavalo de tróia que apresente o comportamento e as características do buraco negro, não é o objetivo da proposta e por isso a mesma não endereça esta vulnerabilidade. Para resolver este problema seria necessário algum tipo de detecção de intrusão ou uso de antivírus, pois o usuário teria a sua chave privada e a estaria usando sem saber do intruso. Uma outra premissa do trabalho é que a chave privada de cada AC nunca fica exposta ou é decifrada por um atacante.

6 - Análise e Métricas

Usar cifragem por limiar implica em escolher um valor T para o número mínimo de chaves

diferentes que podem ser compostas para gerar a chave privada do serviço. Quanto maior o limite mais segura a arquitetura principalmente quando este limite é função do número de ACs e se aproxima dele. Por exemplo, podemos ter vários esquemas com (N,T), sendo N o número de partes em que a chave foi dividida (no nosso caso N é igual ao número de ACs) e T o limite de autenticação. Porém escolher valores de T próximos de N torna o sistema mais lento e com menos disponibilidade já que temos que ter pelo menos "T" ACs disponíveis e gerando chaves corretamente.

Desta forma, ajustar o parâmetro T depende de quão seguro ou disponível queremos que nosso sistema seja. Por outro lado, o fator N depende somente do tamanho ou cobertura da rede e do alcance e da potência dos nós.

Assim, foi usada a função *C=Log(T)+(T-1)* para o número de ACs contatadas por uma AC combinadora e a função $S=\lceil N/\mathrm{Log}\,\overline{2T}\rceil$ para denotar o número de ACs confiáveis que serão contatadas em uma tentativa de comunicação entre um nó e o serviço de certificação digital.

6.1 - Ambiente de Simulação

Foram escolhidos dois cenários de simulação. No primeiro cenário rodamos simulações usando *NS-2* [NS-2] com 5, 7, 9, 11 e 14 ACs distribuídas de maneira uniforme e fixas conforme a Figura 1. Escolhemos apenas dois nós com movimento randômico, com tempos de simulação variando entre 90 e 360 segundos e pudemos observar que, na média, o parâmetro "m" (quantidade de nós intermediários pelos quais as mensagens vão de N1 até N2) fica em torno de *N/4,* sendo N o número de ACs.

Na segunda parte das simulações, foram gerados cenários randômicos no *NS-2* com até 100 nós e a média para o parâmetro "m" não se alterou. Em todas as simulações foram usados os protocolos de roteamento DSR (*Dynamic Source Routing*) e DSDV (*Destination Sequenced Distance Vector)*, protocolo MAC 802.11b e os nós se comunicando em uma taxa CBR (*Constant Bit Rate*) de 4 a 100 Kbps com tamanho de pacote de 300 bytes. Mesmo que uma ou algumas das ACs estivessem desligadas (respeitando a quantidade mínima de ACs para não ferir a cifragem por limiar), os nós conseguem se comunicar e o serviço não fica indisponível.

6.2 - Análise de Disponibilidade

Conforme dito anteriormente, a escolha do limite criptográfico é o item que fornece o grau de segurança e também o nível de disponibilidade. Escolhendo limites altos teremos uma arquitetura menos robusta a falhas, pois mais ACs devem estar confiáveis e disponíveis para o serviço funcionar. Na prática usamos a equação definida em [Shamir 1979] que traz o limite criptográfico em função do número de ACs sendo *N= 2T -1*. Verificamos que precisamos de pelo menos "T" ACs funcionando. Se por acaso tivermos menos de "T" ACs operacionais então o sistema estará indisponível. Por este motivo propomos a inicialização periódica das ACs pelo administrador do sistema. Levando-se em conta que: estamos usando a relação *N=2T -1***,** que cada nó usa a função $S(T) = \lceil N/\mathrm{Log}\,\overline{2T}\rceil$ nas suas operações e que cada nó recebe $(2\times S(T)/3)$ [*Seung and Kravets* 2001] respostas; concluímos que o número mínimo de ACs para este modelo é igual a cinco usando um limite criptográfico igual a três. A função *C(T) =Log(T)+(T-1)* que denota o número de ACs contatadas pela AC combinadora também obedece a este limite inferior, pois nesta configuração será preciso que pelo menos quatro ACs (contando com a combinadora) estejam disponíveis. Obviamente que se usarmos uma função menos segura para a escolha do limite criptográfico teremos uma configuração menos dependente de falhas e conseqüentemente mais robusta.

6.3 - Métricas

Uma das métricas usadas neste trabalho é a quantidade de mensagens trocadas no processo de certificação. A análise foi dividida em duas partes: a primeira o número de mensagens para a renovação de um certificado e a segunda o número de mensagens trocadas no estabelecimento de uma rota entre dois nós. Também foram avaliados o overhead do número de bytes adicionais do protocolo e o consumo de energia pelos nós.

a) Overhead de mensagens na renovação de certificados

Na renovação dos certificados teremos os seguintes passos para cada requisição:

- $\lceil N / \mathrm{Log}\ 2\mathrm{T} \rceil$ mensagens CREQ enviadas;
- Pelo menos $2/3 \times \lceil N / \mathrm{Log}\ 2\mathrm{T} \rceil$ ACKS recebidos. Considerando que 2/3 das mensagens são respondidas como em [*Seung and Kravets* 2001];
- Uma mensagem CERT SEND;
- Log(T) +T-1 mensagens KEY REQ enviadas pela AC combinadora;
- Log(T) +T-1 mensagens KEY SENT enviadas pelas ACs para a AC combinadora;
- Uma mensagem CERT REPLY para o nó que fez o pedido.
- Com N=19 e T=10 (usando a equivalência N=2T-1 de [Shamir 1979]) para um total de 1000 requisições, temos:
- Total: $(5 + 1 + 12 + 12) \times 1000 = 30000$ mensagens.

Comparando com as 119125 mensagens obtidas no método [*Seung and Kravets* 2001] usando *broadcast*, sob as mesmas condições (total de mensagens CREQ) observa-se um ganho aproximado de 75%. Cabe ressaltar que no método proposto neste trabalho basta que apenas uma AC responda ao nó solicitante da renovação de certificados e usando a mesma relação de que 2/3 das mensagens enviadas geram respostas, não é preciso mais que uma rodada quando nosso algoritmo escolhe uma entre as $\lceil N / \mathrm{Log}\ 2\mathrm{T} \rceil$ ACs contatadas.

Fazendo a mesma análise para as mensagens recebidas pelos nós (chamadas de CREP's no método de [*Seung and Kravets* 2001]) temos:

Total: $(2/3 \times 5 + 1) \times 1000 = 4300$ mensagens.

Comparando com as 29776 mensagens recebidas no método de [*Seung and Kravets* 2001] temos um ganho aproximado de 86 %.

Na Tabela 1 é mostrada uma comparação entre o método proposto e o usado em [*Seung and Kravets* 2001] com *broadcast*.

b) Overhead de mensagens no estabelecimento de rotas

No estabelecimento das rotas entre dois nós N1 e N2 passando por "m" nós, teremos "$(m+1) \times 2 \times (S(T)+3)$" mensagens trocadas entre os nós e as ACs, 2+m assinaturas digitais geradas e 2+m verificações de assinaturas. A função *"S(T)+ 3"* representa a quantidade de mensagens enviadas por cada nó para a verificação da validade da assinatura digital juntamente com a verificação da CRL e das três mensagens trocadas entre cada nó para autenticação e passagem da chave simétrica de sessão e do número aleatório para comunicação. Em nossas simulações com *NS-2* encontramos um valor de "m" máximo de *"(N+1)/2"* quando N é ímpar e de *N/2* quando N é par. Usamos sempre topografias retangulares ou quadradas (500 x 500, 1000 x

1000, 500 x 750) com as ACs distribuídas de maneira uniforme conforme a Figura 1. Usamos também movimento randômico dos nós para as simulações.

QTDADE DE CÃS	MÉTODO COM BROADACST		MÉTODO PROPOSTO		GANHO %	
	MSGS CREQ	MSGS CREP	MSGS CREQ	MSGS CREP	CREQ	CREP
15	134694	14953	26000	4300	80,7	71,24
30	119125	29776	43000	4300	63,9	85,56
50	98962	49447	31000	7666	68,67	84,5

Tabela 1: Comparação do número de mensagens trocadas

Com o mesmo cenário de [*Seung and Kravets* 2001] que usa *broadcast*, com N=19 ACs e T=10, (limite criptográfico), com movimento randômico e topografia de 1000 x 1000, temos $10 \times 2((19+1)/2 + 10+3)$, fazendo um total de 460 mensagens trocadas para o estabelecimento de uma rota entre dois nós N1 e N2 no pior caso. Neste caso com uma média de 41,81 mensagens para cada nó. Conforme nossas simulações o parâmetro "m**"** é na média igual a N/4**.** Com isso, na média teríamos 230 mensagens trocadas para o estabelecimento de rota com uma média de 20,9 mensagens por nó.

c) Overhead do protocolo

Em termos de espaço em disco para cada nó, foram escolhidos certificados com uma chave RSA de 1024 bits e com o mínimo possível de campos do padrão X.509 perfazendo um total de 300 bytes (Figura 4) sem contar com os bits necessários para representar as regras de interoperabilidade usadas na codificação DER em ASN.1. A lista de ACs confiáveis mantida em cada nó é implementada usando-se um inteiro sem sinal (32 bits) para representar a idade (age) da lista e ser usado para verificação, dois inteiros sem sinal (64 bits) como um mapa de bits onde o valor "1" em determinada posição "k" corresponde que a AC número "k" é confiável. Quando uma AC se torna não confiável basta colocar o valor "0" na sua posição correspondente. Como exemplo, vamos supor que a AC 10 se torne não confiável. Basta que o algoritmo coloque um "0" na posição 10 da string de 64 bits. Desta forma temos um máximo de 64 ACs o que em nossas simulações foi mais que satisfatório.

Lista de CA´s

Confiáveis

Aging	Vetor de bits - número de CA´s
32 Bits	64 Bits

Lista de ACs confiáveis no instante inicial em uma arquitetura com 9 ACs

Aging	Vetor de bits
0x0	0x9

Campos do Certificado Digital

Campo	Bits
Chave Pública AC	1024
Chave Pública NÓ	1024
Assinatura digital da AC	128
Identificação do nó	32
Hash	128
Distribuição da CRL	8
Data da geração	24
Validade	24
Tipo de uso	8
Total em bits	2400
Total em Bytes	300

***Figura 4*: Campos do certificado digital e formato da Lista de ACs**

O overhead do protocolo em bytes é na verdade uma assinatura digital inserida ao final do cabeçalho dos protocolos de roteamento ou de enlace, ocupando assim 128 bits (usando o protocolo MD-5).

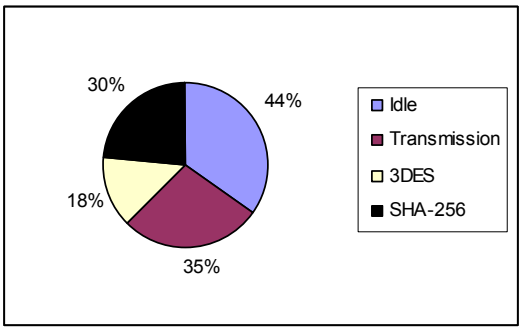
Existe também o overhead causado pelo uso de uma chave simétrica para a troca de mensagens. Tanto este overhead quanto o criado no uso das chaves públicas influi no consumo de energia dos nós. Este consumo torna-se um fator decisivo quando falamos em redes ad hoc, pois normalmente são nós usando baterias para funcionar. Na Figura 5 são mostradas algumas tabelas de consumo de energia quando são usados os diversos algoritmos de chaves simétricas, assimétricas e de funções de *hash*. As tabelas têm por base o consumo em um equipamento Pocket PC Symbol PPT 2800 (processador de 32 bits, 206Mhz, 16Mb flash Rom, 16Mb Ram, cache de instruções de 16 Kb, cache de dados de 8Kb, rodando Windows CE 3.0 e placa Wi-Fi de 11Mbps); Podemos concluir que o uso destes algoritmos não é impeditivo para a comunicação segura quando limitado e feito de maneira otimizada. Existem propostas como em [Karri and Mishra 2002] para otimizar o gasto de energia na comunicação segura de nós sem fio, mas não as estamos considerando em nossa análise por não ser o nosso escopo principal. Se os nós se movimentarem pouco, pouco overhead será acrescentado. Por outro lado se os nós trocarem constantemente informações de novas rotas e novos pedidos de estabelecimento de rotas, um overhead maior será adicionado. Por este motivo nosso modelo propõe a minimização das mensagens trocadas entre os nós e notamos que com movimento randômico, a média de mensagens assinadas ou cifradas, trocadas entre dois nós, visando estabelecimento de rotas, não chega a um número elevado.

Custo de energia dos algoritmos de Hash

Algoritmo	MD2	MD4	MD5	SHA	SHA1
ENERGIA (µJB)	4,12	0,52	0,59	0,76	1,16

(a)

Percentual do consumo de energia para transmissão de 64 Kbytes - 3DES



(b)

Custo de energia dos algoritmos assimétricos

Algo-ritmo	Tam. Chave (Bits)	Geração (mJ)	Assinatura (mJ)	Verificação (mJ)
RSA	1024	270,13	546,5	15,97
DAS	1024	293,20	313,6	338,02
ECDSA	163	226,65	134,2	196,23

(c)

Consumo de energia com algoritmos simétricos

	DES	3DES	ACST	AES	RC5
KEY SETUP (µJ)	27,53	87,04	37,63	7,87	66,54
Enc/Dec (µJ/byte)	2,08	6,04	1,47	1,21	0,79

(d)

Figura 5: Gasto de energia com cifragem.

Fonte: [Poltapally et al. 2003] e [Karri and Mishra 2002]

Por último temos também o overhead de um algoritmo para manter a lista de

ACs atualizada e íntegra para uso entre os diversos nós da rede. Esta lista será usada também para a escolha das ACs que farão parte do processo de autenticação (função *S(T))***.** Este algoritmo faz parte do processo de manutenção do ambiente e é de ordem *O(LogN)* para a busca e *O(N/T)* ou simplesmente *O(N)* para a escolha das ACs que farão parte do lote que será escolhido quando da renovação de certificados e da autenticação entre os nós.

7- Conclusões e Trabalhos Futuros

Neste trabalho é mostrado que com a implantação da arquitetura proposta diminuímos em até 85% o número de mensagens de controle trocadas quando comparado com o esquema proposto em [*Seung and Kravets* 2001], para os cenários analisados. Mostramos que a proposta deste trabalho produz um overhead baixo quanto ao número de bytes e de tempo de processamento e que endereça a forma de tratar a CRL de forma distinta e mais eficiente (via unicast) do que a proposta em [Kong et al. 2001, Luo et al. 2002]. Foi proposta a criação e manutenção de uma lista de ACs confiáveis assim como foram feitas considerações sobre seu uso e a complexidade dos algoritmos para sua manutenção. Foram descritos também os campos que devem ser usados minimamente pelos certificados da solução usando o padrão X.509.

Conforme descrito na seção 2, com a proposta deste trabalho podemos mitigar os ataques buraco negro e de falsificação de identidade nos níveis de rede e o ataque de falsificação de identidade no nível de enlace fazendo com que a rede não seja afetada por nós maliciosos. É importante salientar que a proposta não visa conter os ataques, mas sim torná-los sem efeito protegendo a infra-estrutura e as comunicações.

O gasto de energia da solução é aceitável para diversas plataformas de nós móveis visto que melhora o modelo proposto em [*Seung and Kravets* 2001] diminuindo o número de mensagens e diminuindo assim o consumo de energia e ainda pode ser melhorado com as soluções contidas em [Karri and Mishra 2002] que propõe uma otimização na utilização dos algoritmos de cifragem em redes sem fio.

A solução proposta também poderá ser usada com múltiplos serviços de autoridade certificadora com cifragem por limiar (múltiplas ACs distribuídas) aproveitando a técnica da certificação cruzada. Para um trabalho futuro poderemos descrever o protocolo de manutenção da confiança entre as ACs quando se usam múltiplas ACs distribuídas.

Referências

Aad, I., Hubaux, J.P. and Knightly, E. (2004). “Denial of Service Resilience in Ad Hoc Networks”, *Proceedings of the 10th annual international conference on Mobile computing and networking,* p 202-205, Philadelphia, USA.

Desmedt, Y. (1994). “Threshold Cryptography”, *European Transactions on Telecommunications,* 5(4), p. 449-457, July-August, California, USA.

Douceur, J. (2002). “The Sybil Attack”, *First Int. Workshop on Peer-to-Peer Systems* (IPTPS'02), p 251-260, MIT, MA, USA.

Gasser, M., Goldstein, A., Kaufman, C. and Lampson, B. (1989). “The digital distributed

systems security architecture", *Proceedings of the 12th National Computer Security Conference,* p 305-319.

Haas, Z. and Zhou, L. (1999). "Securing Ad Hoc Networks", *IEEE Network Magazine*, November, NY, USA.

Karri, R. and Mishra, P. (2002). "Minimizing Energy Consumption of secure Wireless Session with QOS Constraints", *Proceedings of the IEEE International Conference on Communications (ICC),* p 2053-2057, vol. 4, New York, USA.

Kaufman, C (1993). "DASS: Distributed authentication security service", Request for Comments: 1507.

Kong, J., Zefros, P., Luo, H., Lu, S. and Zhang, L., (2001). "Providing robust and ubiquitous security support for MANET", 9th IEEE International Conference on Networks Protocols (ICNP 2001), p 251-360, Riverside, California, USA.

Kurose, J. and Ross, K. (2005). "Computer Networking – A Top-Down Approach Featuring the Internet", Ed. Addison Wesley, 3rd Edition.

Luo, H., Zefros, P., Kong, J., Lu, S and Zhang, L. (2002). "Self-securing Ad Hoc Wireless Networks", *Proceedings of the 7th IEE International Symposium on Computer and Communications (ISCC'02),* p. 567-576, 2002, Giardini Naxos, Itália.

Oliveira, L. B., Wong, H. C., Bern, M., Habib, E., Loureiro, A. A. F. and Dahab, R. (2005) "SecLEACH – Uma solução segura de distribuição de chaves para redes de sensores sem fio hierárquicas", *V Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBseg´05),* Florianópolis, Brasil.

Poltapally, N., Ravi, S., Raghunathan, A. and Jha, N.K. (2003). "Analyzing the energy Consumption of Security Protocols", *Proceedings of the 2003 international symposium on Low power electronics and design,* p 30-35, Seoul, Korea.

Reiter, M.K. (1996) "Distributing trust with de Rampart toolkit", *Communications of the ACM,* 39(4) p 71-74, New York, USA.

RSA, http://www.rsa.com.

Seung, Yi and Kravets, R. (2001). "Practical ICP for Ad Hoc Wireless Networks", Department of Computer Science, Illinois University, USA.

Shamir, A. (1979). "How to Share a Secret", *Communications of the ACM,* vol. 22 issue 11, p.612-613, New York, USA.

Simulador NS-2, http://www.isi.edu/nsnam/ns/.

Villela, B. A. M. and Duarte, O. C. M. B. (2004). "Maximum Throughput Analysis in Ad Hoc Networks", *Proceedings of Third International IFIP-TC6 Networking Conference,* p. 223-234, Vol. 3042, Athens, Greece.