UM PROTOCOLO DE ROTEAMENTO ANÔNIMO PARA REDES AD HOC

Andréa Martins Araujo*, Aloysio de Castro P. Pedroza*

* *Grupo de Teleinformática e Automação*
PEE/COPPE - DEL/POLI
Universidade Federal do Rio de Janeiro
Caixa Postal 68504 - CEP 21945-970
Rio de Janeiro, RJ

Emails: andrea@gta.ufrj.br, aloysio@gta.ufrj.br

Abstract— An ad-hoc wireless network consists of a group of mobile nodes that communicate with each other without help of a fixed infraestructure. Unfortunately, it is subject to various security attacks including traffic analysis that threats the privacy and confidentiality of exchanged data. In this paper, we propose an anonymous routing protocol that prevents against traffic analysis hiding the source and destination of packets. An formal description technique (LOTOS) is used to design specifications and validate the protocol.

Keywords— Routing, Anonymity, Formal Description.

Resumo— Uma rede sem fio ad-hoc consiste num grupo de nós móveis que se comunicam entre si sem ajuda de uma infraestrutura fixa. Infelizmente, ela está sujeita a vários ataques incluindo análise de tráfego que ameaça a privacidade e confidencialidade dos dados trocados. Nesse artigo, propomos um protocolo de roteamento anônimo que previne contra a análise de tráfego escondendo a fonte e o destino dos pacotes. Uma técnica de descrição formal (LOTOS) é usada para projetar especificações e validar o protocolo.

Keywords— Roteamento, Anônimato, Descrição Formal

1 Introdução

Uma rede sem fio ad-hoc consiste num grupo de nós móveis que se comunicam entre si através de *links* de rádio freqüência (RF) sem ajuda de uma infraestrutura fixa. Essas redes são adequadas para aplicações como operações de resgate, incursão em terreno hostil e operações militares. Além dessas aplicações, as redes ad hoc já começam a serem usadas em ambientes comerciais e industriais (Meshnetworks, 2004). Qualquer que seja a aplicação, a segurança na troca das informações é fundamental.

A privacidade da comunicação pode ser ameaçada por ataques como análise de tráfego, onde informações e a identidade dos nós estão expostas. Vários trabalhos foram desenvolvidos na tentativa de prover privacidade. Para a Internet, destacamos *Web-MIXes*(Berthold et al., 2000), *Stop-and-Go-MIXes*(Kesdogan et al., 1998), *Onion Routing*(Reed et al., 1998), entre outros. Todos se basearam no método desenvolvido por Chaum (Chaum, 1981) para prover anonimato em correio eletrônico. Nessas redes, o nó fonte pode enviar mensagens secretas que só poderão ser decriptadas pelo receptor. Essas mensagens, antes de chegar ao destino, passam por nós misturadores que são responsáveis por esconder a identidade da origem e destino dos pacotes.

Em redes ad-hoc alguns estudos estão sendo realizados para prover anonimato como o protocolo de roteamento anônimo ANODR (Kong and Hong, 2003) que usa o conceito de *pseudonymity* (Pfitzmann and Köhntopp, 2000) e *trapdoor information*, o *Mix Method* (Jiang et al., 2001), onde são determinados os *mix nodes* que encaminharão pacotes, e um algoritmo de roteamento seguro (El-Khatib et al., 2003) que provê roteamento anônimo. A idéia é transportar o conceito de nós misturadores usado na Internet para redes ad-hoc.

Dentro desse contexto, propomos um protocolo de roteamento anônimo cujo funcionamento é baseado no protocolo DSR (*Dynamic Source Routing*) (Johnson et al., 2003). Os dados trocados são protegidos por criptografia e o encaminhamento de pacotes é realizado somente por *mix nodes*, criando então, um caminho anônimo entre fonte e destino. Na verdade, o algoritmo desenvolvido é uma versão extendida do DSR, incluindo o estabelecimento de uma rota anônima entre fonte e destino.

O protocolo proposto é especificado utilizando a linguagem para descrição formal LOTOS (*Language of Temporal Ordering Specification*)(ISO/IFC, 1988), acompanhada de verificação com o software de análise CADP (Pacote de Desenvolvimento CAESAR/ ALDEBARAN)(Fernandez et al., 1996). Uma variedade de trabalhos que utilizam LOTOS estão sendo desenvolvidos, entre eles destacamos os realizados por Bagatelli *et al.*(Bagatelli et al., 2002) e Leduc *et al.* (Leduc and Germeau, 2000). A validação do algoritmo proposto é feita numa rede livre de intrusos e em situações de ataque, onde consideramos dois casos, a presença de um nó com comportamento malicioso e a presença de um grupo de nós formando um conluio.

Este artigo está organizado da seguinte forma. Na seção 2 faremos uma breve revisão do funciona-

mento do protocolo de roteamento DSR. Na seção 3, descreveremos o protocolo proposto. Na seção 4, será apresentada a metodologia e análise dos resultados. Por fim, na seção 5, descrevem-se algumas consiferações finais sobre o trabalho, bem como propostas de pesquisas futuras.

2 O protocolo de Roteamento DSR

Atualmente, o DSR é o protocolo mais antigo do IETF para roteamento em redes ad-hoc e o mais difundo e testado entre os existentes. Por isso, foi selecionado para servir de base para o desenvolvimento do nosso protocolo.

O DSR opera sob demanda, ou seja, sua tabela de rotas é montada somente se houver necessidade de envio de mensagens e é composto por dois mecanismos principais: *route discovery* (descoberta de rota) e *route maintenance* (manutenção de rota).

A descoberta de rota é iniciada quando um nó recebe um pacote a ser enviado e nao possui rota armazenada para o destino. Então, o nó, chamado fonte, envia um *route request* (requisição de rota) por difusão. Cada nó que recebe esse pacote verifica se ele já foi recebido anteriormente. Caso afirmativo, o pacote é descartado. Caso contrário, o nó, que não é o nó destino, coloca seu endereço como próximo nó intermediário e retransmite o pacote, por *flooding* (inundação) e envia um *ack* (confirmação de recebimento) ao nó anterior. O próximo nó que receber o *route request* fará o mesmo até que o pacote alcance o destino. Este nó, ao receber o *route request*, envia um *route reply* para a fonte através do caminho reverso.

A manutenção das rotas é feita quando cada nó que transmite um pacote é responsável por armazenar o *ack* recebido do nó seguinte, sem que haja sua retransmissão até o nó fonte. Caso ocorra quebra de um *link*, o nó que encaminha um pacote e ao não receber o *ack*, procura uma nova rota para o destino em sua tabela. Caso não tenha esta rota, o nó envia um *route error* ao seu nó antecessor que age da mesma forma até que o *route error* alcance o nó fonte. Quando o nó fonte receber esse pacote, verifica se existe outra rota para o destino. Caso negativo, deve-se iniciar nova descoberta de rota.

3 O Protocolo Proposto

O protocolo apresentado neste trabalho baseia-se no funcionamento do protocolo DSR e tem como objetivo o estabelecimento de um caminho anônimo entre a fonte e o destino da comunicação. O objetivo é esconder a identidade da fonte e do destino através de criptografia, oferecendo, então, privacidade à comunicação. Para tal, usamos o conceito de nós misturadores desenvolvido por Chaum(Chaum, 1981) na Internet, e transportamos essa idéia para as redes ad-hoc fazendo o encaminhamento de pacotes através dos nós misturadores. Esses nós são confiáveis e responsáveis pelo anônimato da comunicação.

Neste trabalho foram feitas algumas considerações para início de projeto, entre elas citamos: o protocolo foi projetado para prevenir ataques do tipo passivo, mais especificamente análise de tráfego; todos os nós da rede possuem um par de chaves assimétricas, uma chave pública e outra privada, ou seja, a distribuição destas não faz parte do projeto; os nós misturadores formam um grupo que divide o mesmo par de chaves assimétricas e também conhecem todas as chaves públicas da rede.

O funcionamento básico do nosso protocolo está descrito abaixo:

- Passo 1 - O nó *source* S deseja estabelecer comunicação anônima com o nó *target* T, porém não tem caminho para T. Então, S deve iniciar a descoberta de rota com encaminhamento de pacotes através de nós misturadores. S não sabe quais nós são misturadores, então, envia um *mix request* em difusão.
- Passo 2 - Os nós misturadores que receberem um *mix request* respondem com um *mix reply* contendo a chave pública Km, vide Figura 1.

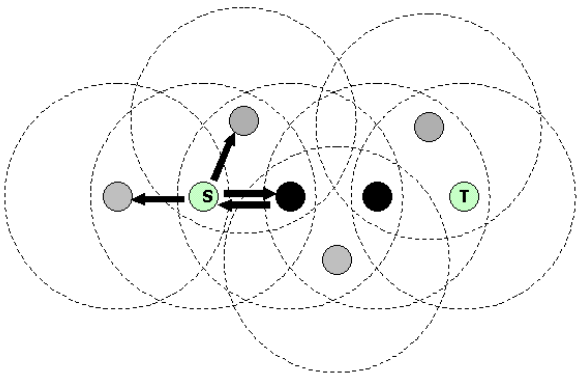

Figura 1: **Envio do Mix Request e recebimento do Mix Replay**

- Passo 3 - Ao receber um *mix reply*, S monta um *route request* que contém: o endereço de origem (endereço de S) criptografado por Kt (chave pública de T), o endereço de destino final (endereço de T) criptografado por Km e as informações destinadas somente a T criptografadas por Kt (chave pública de T). S envia o *route request* para o misturador do qual recebeu um *mix reply*.
- Passo 4 - Quando um misturador recebe um *route request*, ele decifra o endereço de destino final através de sua chave privada e verifica se é vizinho de T. Caso negativo, ele encripta novamente o endereço de T com Km e envia o pacote em difusão, vide Figura 2.

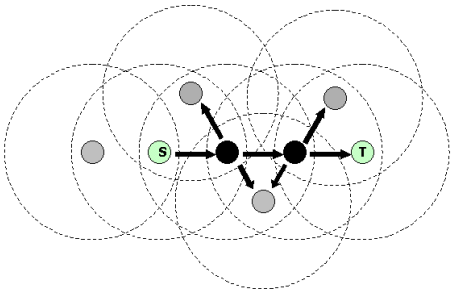

Figura 2: **Envio do Route Request**

- Passo 5 - O próximo misturador age da mesma forma, até o pacote seja recebido por um misturador vizinho a T. Neste caso, o misturador criptografa o endereço de T com Kt e envia por difusão. Somente T pode decifrar a mensagem, os demais nós descartam o pacote.
- Passo 6 - T monta seu *route reply* que contém: o endereço de origem (endereço de T) e o endereço de destino final (endereço de S) criptografados por Ks. T envia o *route reply* para S pelo caminho reverso, vide Figura 3.

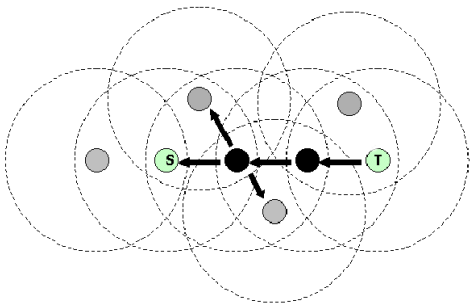

Figura 3: **Envio do Route Replay**

- Passo 7 - Como o endereço de destino final do *route reply* está criptografado por Ks, o misturador vizinho a S não sabe para quem entregar o pacote. Então, o pacote é enviado por difusão. E somente o nó S é capaz de decifrá-lo, os demais nós descartam o pacote. Fica estabelecido, então, o caminho anônimo entre S e T.
- Passo 8 - Caso haja uma quebra de link, o nó misturador deve enviar um *route error* para seu nó antecessor (do qual recebeu um *route request*) e assim por diante até alcançar o misturador vizinho ao nó S. Como esse nó misturador desconhece a identidade da fonte, ele envia o *route error* por difusão, vide Figura 4. Todos os nós vizinhança recebem esse pacote, que só terá significado para S. Caso não exista outra rota armazenda para T, S deve iniciar o processo de descoberta de rota.

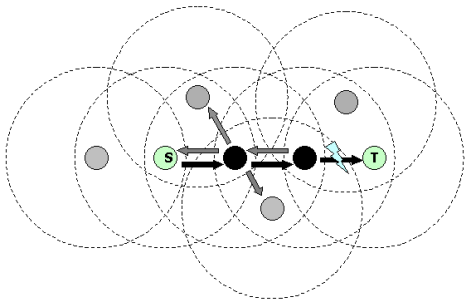

Figura 4: **Envio do Route Error**

4 Metodologia e Análise dos resultados

Para a modelagem do protocolo foram utilizadas técnicas de descrição formal (*FDT - Formal Description Techniques*), no caso a linguagem LOTOS. A vantagem de usar as FDTs se deve, sobretudo, a sua capacidade de tornar as especificações livres de ambigüidades e erros, além de atuar como um guia para a implementação. Em particular, essas técnicas são adequadas à análise dos aspectos de segurança do protocolo.

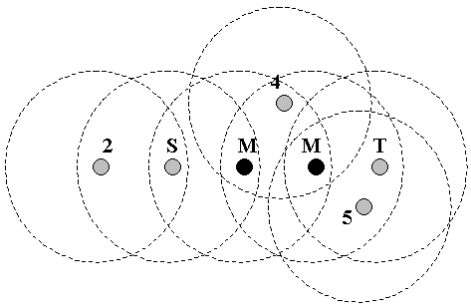

Figura 5: **Rede Ad-Hoc simples**

Utilizou-se para análise, o pacote de ferramenta para engenharia de protocolos CADP. Faz parte dessa análise com CADP, a compilação do código, simulação, verificação, validação e testes da descrição formal do protocolo. As propriedades do protocolo, bem como a visualização dos grafos, são exibidos a partir do emprego do conjunto de ferramentas *BCG* (integrante do pacote CADP). Entre todas as propriedades, a minização e a equivalência observacional são as mais testadas. A primeira reduz os grafos do sistema de transições rotuladas ou LTS (*Labelled Transition Systems*), retirando, então, as redundâncias e permitindo uma análise mais simples. A segunda verifica se o protocolo especificado representa os serviços discriminados inicialmente como requisitos de projeto.

De forma a ilustrar a metodologia adotada para a verificação do protocolo, toma-se como exemplo a versão 8 do protocolo, inicialmente, numa rede sem intrusos, vide Figura 5. Nessa verificação foram definidos três processos, que representam os

três comportamentos, *route discovery*, *route maintenance* sem falhas e outro com quebra de *link*. Dentro desses processos foram definidos subprocesso que representam o comportamento dos nós da rede, além de uma subprocesso *Meio* que representa o canal de comunicação onde são trocados pacotes entre os nós. O trecho de código a seguir exemplifica o comportamento do mecanismo *route discovery*.

```
process
Route_Discovery[U0, U1, U2, U3, U4, U5, Ut] : exit :=

hide M0_in, M0_out, M1_in, M1_out, M2_in, M2_out,
     M3_in, M3_out, M4_in, M4_out, M5_in, M5_out,
     Mt_in, Mt_out in
(
(
Source[U0, M0_in, M0_out]
(Kt, Km, Ks, R, Ss, As, At, Ab, A1)
 |||
No1_mix[U1, M1_in, M1_out]
(Kt, Km, Ks, R, Sm, As, At, Ab, A1)
 |||
No2[U2, M2_in, M2_out](As, Ab)
 |||
No3_mix[U3, M3_in, M3_out]
(Kt, Km, Ks, R, Sm, As, At, Ab, A1, A3)
 |||
No4[U4, M4_in, M4_out]
    (Kt, Km, Ks, R, Sm, As, At, Ab, A1, A3)
 |||
No5[U5, M5_in, M5_out]
    (Kt, Km, Ks, R, Sm, As, At, Ab, A1, A3)
 |||
Target[Ut, Mt_in, Mt_out]
(Kt, Km, Ks, R, St, As, At, Ab, A1, A3)
)

|[M0_in, M0_out, M1_in, M1_out, M2_in, M2_out,
  M3_in, M3_out, M4_in, M4_out, M5_in, M5_out,
  Mt_in, Mt_out]|

(
Meio[M0_in, M0_out, M1_in, M1_out, M2_in, M2_out,
     M3_in, M3_out, M4_in, M4_out, M5_in, M5_out,
     Mt_in, Mt_out]
    (Kt, Km, Ks, R, As, At, Ab, A1, A3)
)
)
```

O operador ||| demonstra que os processos possuem comportamentos independentes e paralelos, enquanto o operador |[M0_in, ..., Mt_out]| representa o sincronismo entre os processos através das portas de comunicação M0_in, ..., Mt_out. Então, os nós da rede possuem comportamento independentes e paralelos entre si e se sincronizam com o *Meio* através das portas de comunicação por onde entram e saem os pacotes.

A especificação completa contém também a representação do comportamento do mecanismo *route maintenance*, com e sem situações onde haja quebra de link.

O serviço proposto para este modelo tem por objetivo estabelecer rotas para um destino e restabelecer a transmissão de dados,caso ocorra uma quebra num link, através do mecanismo de *route maintenance*. A Figura 6 apresenta o grafo LTS desse serviço. Os círculos representam os estados, as setas representam as transições e os rótulo representam as ações executadas. Porém, essas ações são interações com a camada superior (usuário) a camada do protocolo proposto. As trocas de pacotes do protocolo são ações internas e escondidas (função *hide*) e podem ser verificadas ação por ação através de simulações usando a ferramenta *Open/Caesar*.

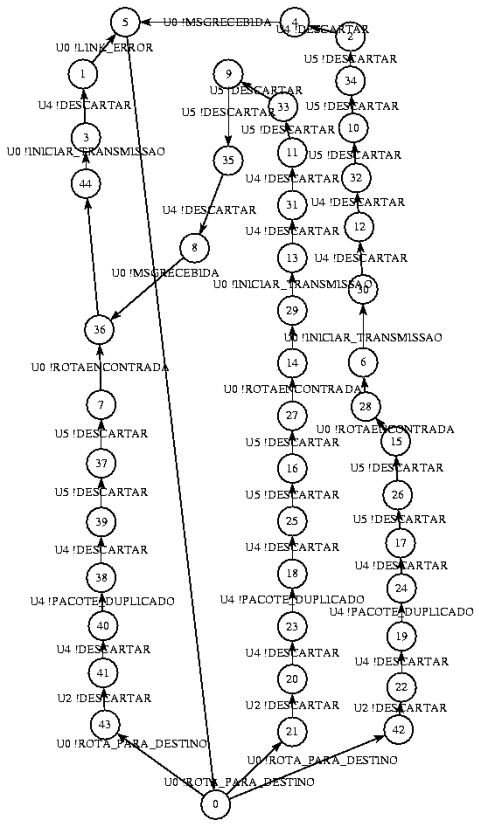

Figura 6: **LTS do Serviço - grafo minimizado**

O resultado mais importante para este exemplo é comprovar que o serviço foi oferecido, ou melhor, se o protocolo conseguiu estabelecer rota para o destino e fazer sua devida manutenção. Com o uso da ferramenta *Aldebaran* obteve-se TRUE como resultado, ou seja, o modelo de protocolo e serviço descritos em LOTOS apresentam equivalência observacional, demonstrando que o protocolo especificado atende aos requisitos do serviço. Este resultado comprova a exatidão do modelo proposto. Constata-se, além da equivalência entre protocolo e serviço, que o protocolo descrito também é vivo e reinicializável, visto que, de qualquer estado, é possível alcançar o estado inicial sem a presença de *deadlocks* e *livelocks*. A Tabela 1 apresenta o número de estado e transições do protocolo antes e depois da minimização para vário estágios de evolução do algoritmo (aumento do número de nós, logo definições de mais comportamentos).

4.1 Estudo de Casos

Para verificar que o algoritmo, além de encontrar a rota, não permitia a identificação da origem e destino, introduzimos algumas situações de ataque. O nós intrusos através da análise do tráfego ten-

MODELO	NÓS	ESTADOS	TRANSIÇÕES	MIN.
Protocolo	5	234	236	não
Protocolo	5	211	213	sim
Serviço		19	21	não
Serviço		14	16	sim
Protocolo	6	372	376	não
Protocolo	6	326	328	sim
Serviço		37	39	não
Serviço		31	33	sim
Protocolo	7	503	505	não
Protocolo	7	457	459	sim
Serviço		51	53	não
Serviço		45	47	sim

Tabela 1: Geração dos LTS

tarão descobrir a identidade e outras informações sobre a fonte e o destino da comunicação.

Caso 1 - Consideramos que o nó 4 da rede, vide Figura 5 tem comportamento malicioso. Ele passou a observar o tráfego trocado entre os misturadores, na tentativa de identificar a origem e o destino dos pacotes, bem como obter informações confidenciais. A Figura 7 apresenta parte do grafo LTS do serviço referente ao *route discovery* na presença do nó malicioso. Observando os rótulos verifica-se que o intruso não conseguiu obter informação alguma.

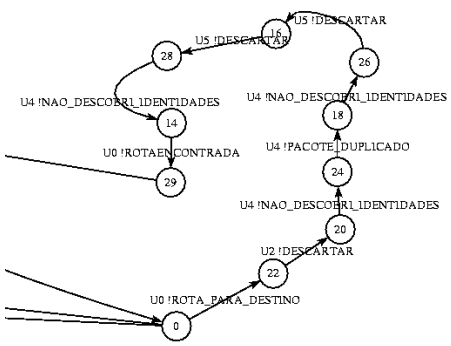

Figura 7: **LTS minimizado do Serviço - Nó4 malicioso**

Caso 2 - Consideramos que todos os nós misturadores formam um conluio. Desta forma, o nó que descobrisse alguma informação consistente repassava-a aos outros misturadorres, até que eles pudessem se certificar da identidade da fonte e do destino da comunicação. A Tabela 2 apresenta o número de estados e transições alcançados por estes dois casos.

MODELO	INTRUSO	ESTADOS	TRANS.	MIN.
Protocolo	nó4	538	540	não
Protocolo	nó4	485	487	sim
Serviço	nó4	56	58	não
Serviço	nó4	49	51	sim
Protocolo	nó1 e nó3	530	532	não
Protocolo	nó1 e nó3	483	485	sim
Serviço	nó1 e nó3	69	71	não
Serviço	nó1 e nó3	62	64	sim

Tabela 2: Geração dos LTS - rede com intrusos

Constatou-se que foi possível identificar o destino da comunicação. A Figura 8 apresenta o LTS do mecanismo de *route discovery* deste caso. O rótulo **U1 ! Descobri_destino_no_Target** indica que o primeiro misturador descobriu a identidade do nó destino. De forma análoga, a ação **U3 ! Descobri_destino_no_Target** demonstra que o **No3_mix** descobre a identidade do destino. Porém, os misturadores não tem como provar que o nó Source é a origem dos pacotes, isso pode ser verificado através do rótulo ou ação **U1 ! Acho_Source_igual_Origem**.

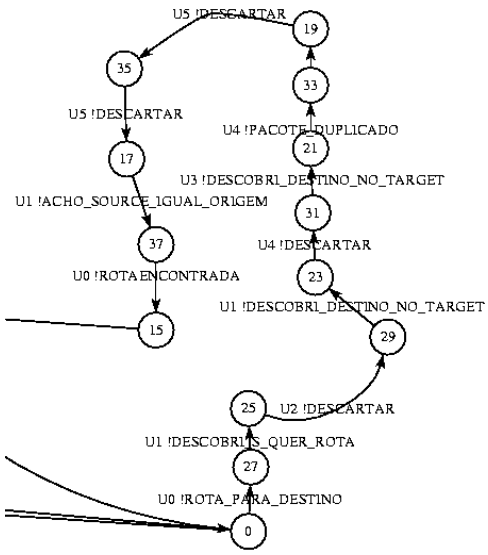

Figura 8: **LTS minimizado do Serviço - conluio de misturadores**

5 Considerações Finais

Privacidade e Anonimato são aspectos de segurança cada vez mais importantes para a comunicação em redes móveis ad-hoc. O protocolo apresentado neste trabalho está baseado no funcionamento do DSR e cria rotas anônimas com o uso de encaminhamento de pacotes por nós especiais, os nós misturadores. O objetivo é criar um caminho anônimo entre a fonte e o destino impedindo que um ataque do tipo análise de tráfego consiga identificar a origem e do destino do pacote, bem como as informações trocadas. Para isso, escondemos a identidade da fonte e do destino com o uso de criptografia.

A vantagem do algoritmo é que a busca por nós misturadores está inserida no algoritmo de roteamento como um único algoritmo, ao contrário do método desenvolvido por Jiang *et al.* (Jiang et al., 2001). Além disso, o algoritmo proposto pode ser considerado uma versão extendida do DSR, devido a inclusão do conceito de anônimato surgido com o trabalho de Chaum (Chaum, 1981). Neste modelo proposto, o nó fonte não precisa ter conhecimento sobre a topologia geral da rede. Em relação à segurança, o algoritmo não foi projetado para enfrentar ataques como *Denial-of-Service (DoS)*, somente ataques

de análise de tráfego. Nos estudos de casos analisados, verificações e simulações mostraram que no caso 1, o nó malicioso não conseguiu obter nenhuma informação, enquanto no caso 2, a identidade do nó destino é desvendada e existe uma suspeita sobre a identidade da origem que não pode ser confirmada.

A abordagem adotada se diferencia das demais, pela utilização de técnicas de descrição formal, em particular a linguagem LOTOS, com seu rigoroso processo de validação gerou uma especificação livre de ambigüidades e erros, além de facilitar a expansão e alteração do algoritmo.

Nesta primeira fase, foi considerado um modelo referente a uma topologia específica que facilita a modelagem e a inserção de funcionalidades. Entretanto, esse modelo se torna bastante complexo, não permitindo descrever o protocolo por completo. A solução é criar um modelo genérico, que seja independente da topologia e do número de nós considerados. Este modelo genérico estará desenvolvido numa fase posterior.

Agradecimentos

Este trabalho foi financiado com recursos da FAPERJ.

Referências

Bagatelli, R., Moura, D. F. C. and de Castro P. Pedroza, A. (2002). Especificação Formal de uma Arquitetura de Suporte à Descoberta de Serviços em Redes Móveis Ad Hoc, *V Workshop de Métodos Formais, WMF'2002*, Gramado, RS, Brasil.

Berthold, O., Federrath, H. and Köhntopp, M. (2000). Project "Anonymity and Unobservability in the Internet", *In Computers Freedom and Privacy Conference, CFP'2000, Workshop on Freedom and Privacy by Design.*

Chaum, D. L. (1981). Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, *Communications of the ACM* **24**(2): 84–90.

El-Khatib, K., Korba, L., Song, R. and Yee, G. (2003). Secure Dynamic Distributed Routing Algorithm for Wireless Ad Hoc Networks, *Proceedings in International Conference on Parallel Processing Workshops.*

Fernandez, J. C., Garavel, H., Kerbrat, A., Mounier, L., Mateescu, R. and Sighireanu, M. (1996). CADP: A Protocol Validation and Verification Toolbox, Vol. 1102, Springer-Verlag, New Brunswick, NJ, USA, pp. 437–440.

ISO/IFC (1988). *Is 8807: Information Processing Systems - Open Systems Interconnection - LOTOS - A Formal Description Techniques Based on The Temporal Ordering of Observacional Behavior.*

Jiang, S., Vaidya, N. and Zhao, W. (2001). A Dynamic Mix Method for Wireless Ad-Hoc Networks, *Military Communications Conference, MILCOM 2001)* .

Johnson, D. B., Maltz, D. A. and Hu, Y.-C. (2003). The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), *Work in progress, draft-ietf-manet-dsr*, IEEE.
*http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt

Kesdogan, D., Egner, J. and Buschkes, R. (1998). Stop-and-Go-Mixes Providing Probabilistic Security in an Open System., *Information Hiding: Second International Workshop* **1525**: 83–98.

Kong, J. and Hong, X. (2003). ANODR: ANonymous on Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks, *Proceedings of the 4th ACM International Symposium on Mobile ad hoc Networking Computing*, ACM Press.

Leduc, G. and Germeau, F. (2000). Verification of Security Protocols using LOTOS-Method and Application, *Computer Communications* **23**: 1089–1103.

Meshnetworks (2004). *http://www.meshnetworks.com.*

Pfitzmann, A. and Köhntopp, M. (2000). Anonymity, Unobservability and Pseudonymity - A Proposal of Terminology, Vol. 2009 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Germany, Berkeley, CA, USA.

Reed, M. G., Syverson, P. F. and Goldschlag, D. M. (1998). Anonymous Connections and Onion Routing, *IEEE Journal on Special Areas in Communications* **16**(4): 482–494.