Um Protocolo de Roteamento Anônimo para Redes Móveis Ad-Hoc

Andréa Martins Araujo[1 *], **Aloysio de Castro P. Pedroza**[1]

[1]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

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

Abstract. *An ad-hoc wireless network permits wireless mobile nodes to communicate without help of a fixed infraestructure. Futhermore, it is subject to various security attacks including traffic analysis. In this paper, we propose an anonymous routing protocol that prevents against traffic analysis hiding the source and destination of packets. The protocol is specified using formal description techniques.*

Resumo. *Uma rede sem fio ad-hoc permite que nós móveis sem fio se comuniquem sem ajuda de uma infraestrutura fixa. Além disso, ela está sujeita a vários ataques incluindo análise de tráfego. Nesse artigo, propomos um protocolo de roteamento anônimo que previne a análise de tráfego escondendo a fonte eo destino dos pacotes. O protocolo é especificado usando técnicas de descrição formal.*

1. Introdução

Uma rede sem fio ad-hoc consiste de um grupo de nós móveis que se comunicam entre si através de *links* de rádio freqüência (RF) sem ajuda de uma infraestrutura fixa. Cada nó móvel atua como um roteador tornando o mecanismo de roteamento mais vulnerável a falhas e ataques.

Aplicações como operações de resgate, incursão em terreno hostil e operações militares necessitam trocar informações de maneira segura. Enquanto mecanismos de segurança fim-a-fim protegem os dados, informações como as identidades dos usuário podem ser reveladas através de análise de tráfego. Daí a importância do estudo de métodos que proporcionem confidenciabilidade e privacidade à comunicação.

Para a Internet, vários protocolos foram desenvolvidos, *Web-MIXes* [Berthold et al., 2000], *Stop-and-Go-MIXes* [Kesdogan et al., 1998], *Onion Routing* [Reed et al., 1998] e muitos outros, que se basearam no método proposto por Chaum para prover anonimato em correio eletrônico: *MIX networks* [Chaum, 1981].

*Este trabalho é financiado com recursos da FAPERJ.

Nessas redes, o nó fonte pode enviar mensagens secretas que só poderão ser decriptadas pelo receptor.

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 infomation*, 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 usando criptografia e assinatura digital.

Nesse artigo, 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 protocolo desenvolvido é uma extensão do DSR que provê privacidade à comunicação.

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].

Este artigo está organizado da seguinte forma. Na seção 2 faremos uma revisão 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 alguns considerações 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 dentre os existentes, e portanto, foi selecionado para servir de base para o desenvolvimento do protocolo proposto.

O DSR opera sob demanda, ou seja, sua tabela de rotas é montada somente se houver necessidade de envio de mensagens. 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 o retransmite, por *flooding* (inundação) e envia um *ack* (confirmação de recebimento) ao nó antecessor. 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* do nó seguinte, sem que haja retransmissão deste *ack* até o nó fonte.

Caso ocorra quebra de um *link*, o nó que encaminha um pacote, 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.

3. Protocolo Proposto

Na Figura 1 apresentamos a rede ad-hoc usada para o desenvolvimento das primeiras versões do protocolo. Esta rede é bastante simples, mas adequada ao estágio inicial do projeto. Posteriormente, serão incluídos mais nós, alguns nós intrusos, e desta forma a complexidade do protocolo aumentará.

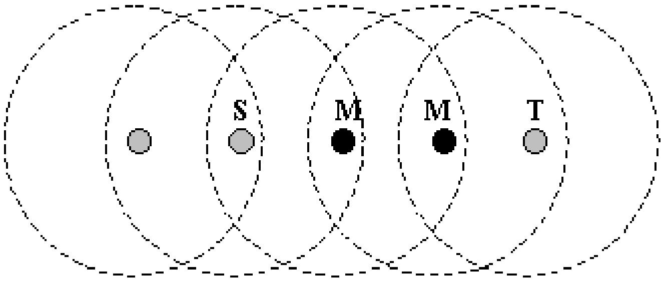


Figura 1: Rede Ad-Hoc simples

O protocolo apresentado neste trabalho basea-se no funcionamento do protocolo DSR e tem como objetivo o estabelecimento de um caminho anônimo entre a fonte e o destino da comunicação. A idéia 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] e estabelecemos que o encaminhamento de pacotes só seria realizado por estes nós, pois eles são confiáveis e responsáveis pelo anonimato da comunicação.

3.1. Considerações de Projeto

Neste trabalho só foram considerados 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. O encaminhamento de pacotes para estabelecimento de uma comunicação anônima só deve ser feito através de nós do tipo misturador. Esses nós confiáveis formam um grupo que divide o mesmo par de chaves assimétricas. Somente eles conhecem todas as chaves públicas da rede.

3.2. Notação

As notações são definidas a seguir:

- Ki: Chave pública do nó i;
- Si: Chave privada do nó i;

- Km : Chave pública dos misturadores;
- Sm : Chave privada dos misturadores.

3.3. O Protocolo

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* por difusão (*b*radcast).

- Passo 2 - Os nós misturadores que receberem um *mix request* respondem com um *mix reply* contendo a chave pública Km, vide Figura 2(a).

- Passo 3 - Ao receber um *mix reply*, S monta um *route request* que contém: o endereço de origem (endereço de S) criptografado por Ks, endereço de destino final (endereço de T) criptografado por Km e informações destinadas somente a T, como a chave privada de S, criptografadas por Kt. S envia o *route request* para o misturador do qual recebeu um *mix reply*, vide Figura 2(b).

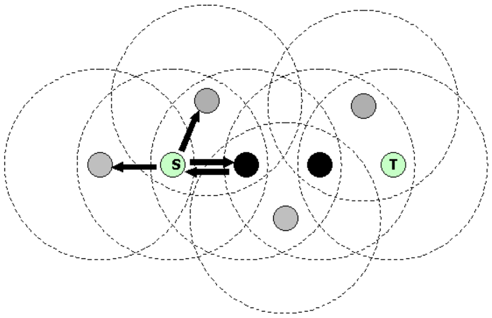
- Passo 4 - Quando um misturador recebe um *route request*, ele decifra o endereço de destino final através de sua chave privada e verifica se é vizinho de T. Caso negativo, ele cifra novamente o endereço de T com Km e envia o pacote por difusão.

- Passo 5 - O próximo misturador age da mesma forma até que o pacote seja recebido por um misturador vizinho de 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. S enviou sua Ss para T, desta forma T pode decifrar o endereço de origem.

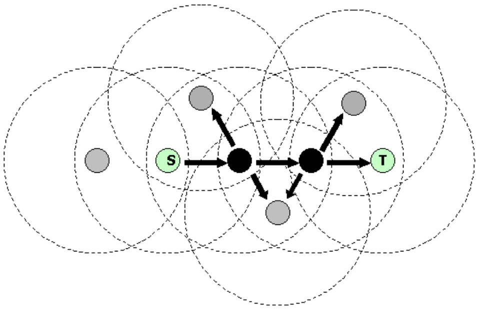
- Passo 6 - T monta seu *route reply* que contém: o endereço de origem (endreço de T) e endreço de destino final (endereço de S) criptografados por Ks. T envia o *route reply* para S pelo caminho reverso, vide Figura 2(c).

- Passo 7 - Como o endereço de destino final do *route reply* está criptografado por Ks, o misturador vizinho de S não sabe para quem entregar o pacote. Então, o pacote é enviado em difusão. E somente o nó S é capaz de decifrá-lo, os demais nós descartam o pacote. Fica estabelecido o caminho anônimo entre S e T.

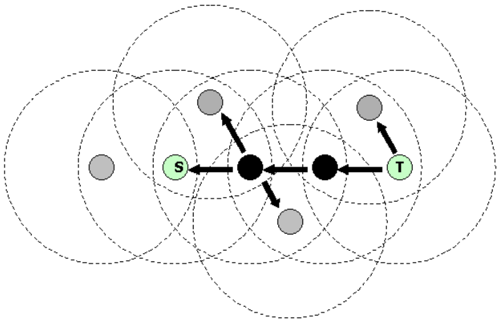
- Passo 8 - Caso haja uma quebra de link, o nó misturador deve enviar um *route error* para seu nó antecessor (do qual recebeu um *route request*) e assim por diante até alcançar o misturador vizinho ao nó S. Como esse nó misturador desconhece a identidade da fonte, ele envia o *route error* por difusão, vide Figura 2(d). Todos os nós da vizinhança recebem esse pacote, que só terá significado para S. Caso não exista outra rota armazenada para T, S deve iniciar o processo de descoberta de rota.



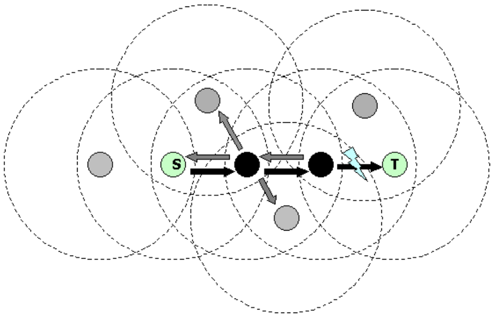
(a) Envio do Mix Request e recebimento do Mix Replay



(b) Envio do Route Request



(c) Envio do Route Replay



(d) Envio do Route Error

Figura 2: Sequência de operação do Protocolo

4. Metodologia e Análise

Para a modelagem do protocolo foi utilizada uma técnica 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, além de atuar como um guia para a implementação. Em particular, essas técnicas são adequadas à análise dos aspectos de segurança do protocolo.

O projeto em questão, adotou uma rede ad-hoc simples, composta por cinco nós, dos quais, dois são nós misturadores (*mix nodes*). O ponto de partida do trabalho foi uma rede com apenas dois nós, podendo parecer desnecessário essa simulação, mas fundamental, pois promove a correção rápida da especificação. E a partir desses dois nós, adicionou-se mais nós, até atingir o cenário atual.

Com o objetivo de facilitar à análise, foi considerada duas situações: rede sem a presença de nós intrusos e com nós intrusos. A especificação do protocolo inclui a descrição, em LOTOS, do comportamento interno do nó e de suas interações com o meio externo e os demais nós da rede.

Utilizou-se para análise, o pacote de ferramenta para engenharia de protocolos CADP. O uso da interface gráfica TCL-TK, Eucalyptos, proporcionou um acesso fácil às funcionalidades das ferramentas CADP, é mais simples do que a chamada pela linha de comando. Faz parte dessa análise com CADP, a compilação do código, simulação, verificação, validação e testes da descrição formal do protocolo. Entre todas as propriedades, duas são as mais testadas:

- Equivalência Observacional: Usada para verificar se o protocolo especificado representa os serviços discriminados inicialmente como requisitos de projeto.

- Minimização: Usada para reduzir os grafos do sistema de transições rotuladas ou LTS (*Labelled Transition Systems*), retirando, então, as redundâncias e permitindo uma análise mais simples.

4.1. Linguagem LOTOS

LOTOS é uma linguagem de descrição formal adequada para a especificação de protocolos e sistemas distribuído e concorrentes em geral. Esta linguagem, padronizada pela ISO, é composta por duas sub-linguagens:

- LOTOS Básico: sua representação é inspirada em álgebra de processos, CCS (*Calculus of Communicating Systems*) e CSP (*Communicating Sequencial Process*), e descreve o modelo comportamental do sistema;

- LOTOS Completo: inclui a linguagem de especificação ACTONE (*Abstract Data Type Formalism*). Os tipos de dados são descritos por seus operadores e equações, as quais são especificadas mediante o emprego de operações algébricas.

OPERADOR	DESCRIÇÃO
P[f,g]	Instanciação do processo P, onde f e g são as portas por onde dados são trocados
P[f,g](S,T)	Chamada do processo P com portas f e g e parâmetros de valores S e T
hide g, h in	Portas serão escondidas na apresentação dos resultados
A[ ]B	Escolha entre processos A ou B, com igual probabilidade
p!V?X:T	Interação pela porta p, envio do valor V e recepção em X de um valor do tipo T
exit	Termina com sucesso
stop	Parada de processo

Tabela 1: Alguns operadores de LOTOS

4.2. Análise

De forma a ilustrar a metodologia adotada para a verificação do protocolo, toma-se como exemplo a versão 3 do protocolo, que leva em consideração somente o route discovery, numa rede sem presença de intrusos. Nessa verificação foram definidos cinco processos, que representam os cinco nós, interligados por um meio de comunicação simples para troca de dados. A representação do comportamento do processos é descrita no trecho de código a seguir:

```
Specification ANDSR_V3 [U0, U1, U2, U3, Ut]: noexit

library ANDSR_LIB3 endlib (* Declaração da biblioteca *)

behaviour

hide M0_in, M0_out, M1_in, M1_out, M2_in, M2_out, M3_in, M3_out, Mt_in, Mt_out in
(
(* Declaraçao de processos *)
(
 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)
 |||
 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, Mt_in, Mt_out]|

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

)
```

A biblioteca **ANDSR_LIB3.lib** é declarada no início do código e contém a declaração e comportamento de todas as funções usadas no código do protocolo.

O operador | | | demonstra que os processos **Source, No1_mix, No2, No3_mix** e **Target** possuem comportamento independentes e paralelos, enquanto o operador | **[M0_in, ..., Mt_out]** | representa o sincronismo entre esses processos e o processo **Meio** através das portas de comunicação M0_in, ..., Mt_out. As variáveis entre () são utilizadas dentro dos processos. O operador **hide** esconde as portas internas do processos, ou seja, somente as portas **U0, ..., Ut** terão vizualização externa.

O próximo trecho se refere ao modelo do serviço oferecido pelo protocolo. O serviço é modelado somente usando as portas externas, pois são suficientes para modelá-lo. **U0 ! Rota_para_Destino** demonstra o pedido de rota para o destino e **U0 ! RotaEncontrada** demostra que a rota foi encontrada ao final do algoritmo.

```
Specification SERVICE [U0, U1, U2, U3, Ut]: noexit

library ANDSR_LIB3 endlib

behaviour

SERVICE [U0, U1, U2, U3, Ut]

where

process SERVICE [U0, U1, U2, U3, Ut]: noexit :=

U0 ! Rota_para_Destino;
U0 ! RotaEncontrada;
SERVICE [U0, U1, U2, U3, Ut]

endproc

endspec
```

4.3. Verificação

O resultado mais importante para este exemplo é comprovar que o serviço foi oferecido, ou melhor, se o protocolo conseguiu ou não estabelecer rota para o destino. Com o uso da ferramenta ALDEBARN [Fernandez et al., 1996] obteve-se resultado **TRUE** para a equivalência entre o modelo de protocolo e serviço descritos em LOTOS, ou seja, eles apresentam equivalência observacional, demonstrando que o protocolo especificado atende aos requisitos do serviço, vide Figura 3 e Figura 4. Além, disso comprovamos que o protocolo proposto não bloqueia (não possui deadlocks), é reinicializável, de qualquer estado alcançamos o estado inicial, e não possui iterações não executáveis. As trocas de mensagens foram verificadas uma a uma no simulador de testes da ferramenta. Sendo assim, este modelo inicial do protocolo está validado.

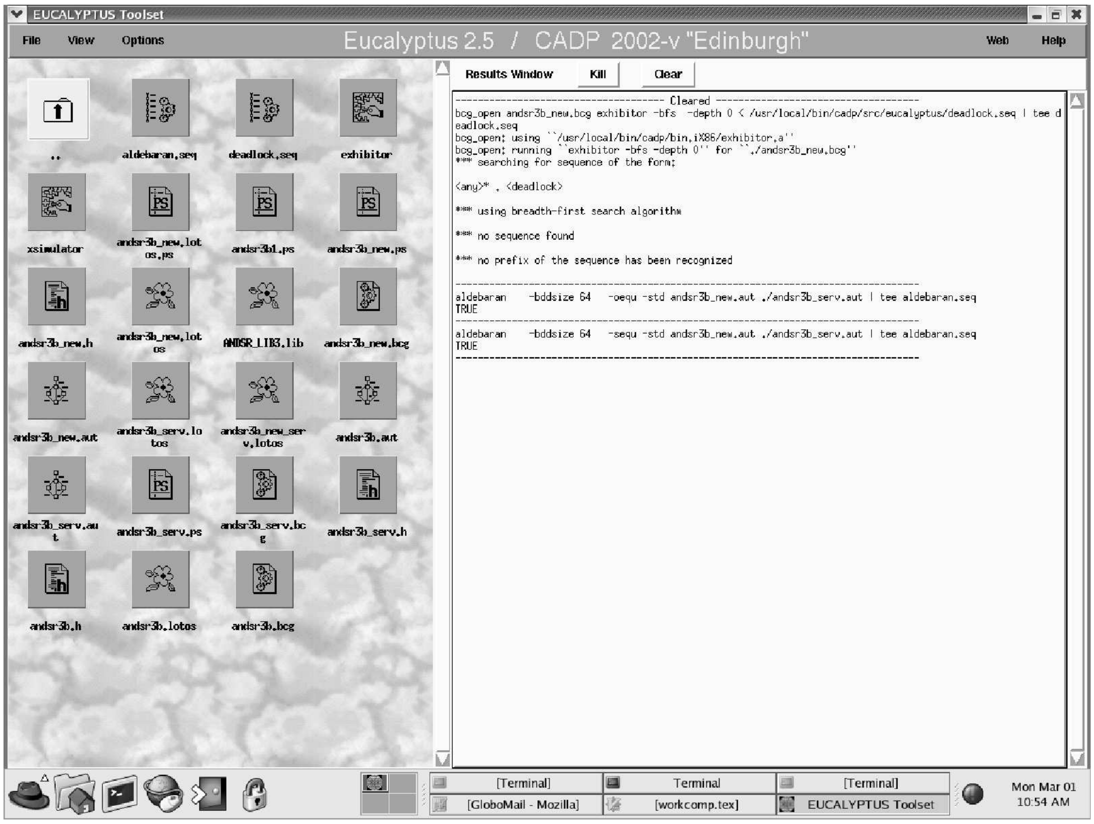


Figura 3: Eucalyptus - janela de resultados

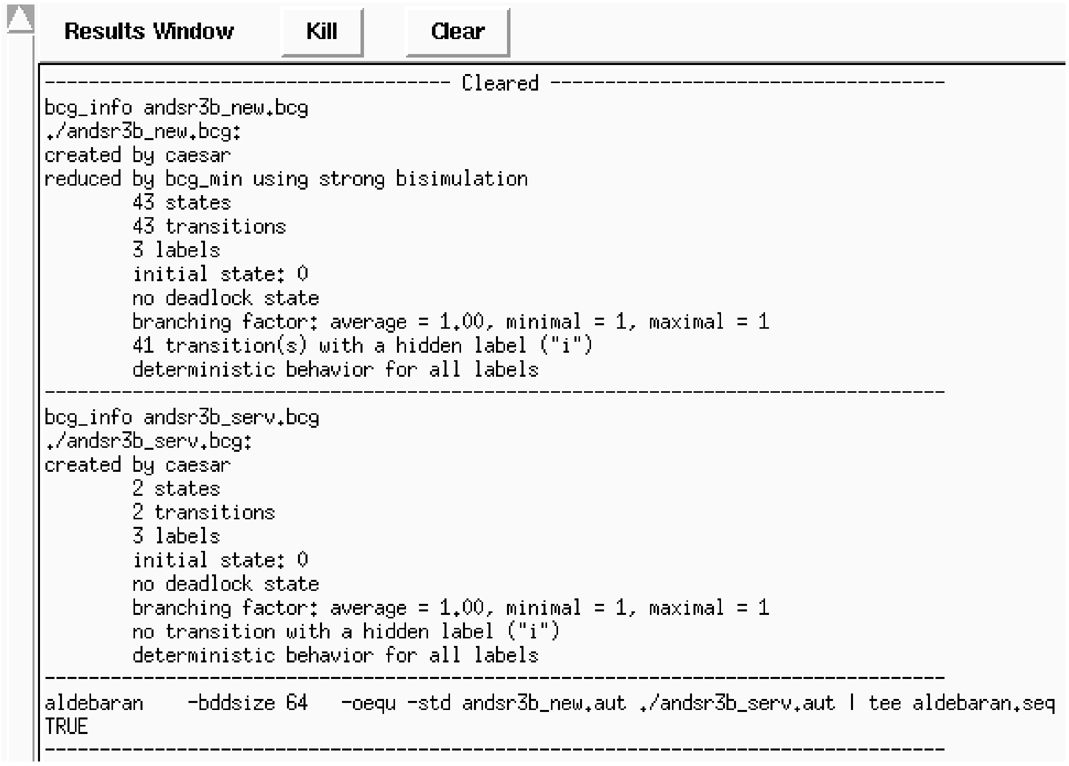


Figura 4: Alguns Resultados

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 cria rotas anônimas com o uso de encaminhamento de pacotes por nós misturadores. O objetivo é criar um caminho anônimo entre a fonte e o destino, escondendo a identidade de ambos. Desta forma, um ataque do tipo análise de tráfego teria menores chance de conseguir identificar a origem e o destino do pacote, bem como as informações trocadas devido a criptografia. A vantagem desse algoritmo é que a procura por nós misturadores está inserida no algoritmo de roteamento como um único algoritmo, ao contrário do Mix Method desenvolvido por Jiang *et al.* [Jiang et al., 2001] que propõe um protocolo para fazer a busca dos misturadores na rede.

Na verdade, o algoritmo desenvolvido é uma extensão do DSR, devido a inclusão do anônimado da comunicação. Nossa abordagem, utilizando descrição formal, oferece uma especificação livre de ambigüidades e erros, além de facilitar a expansão e alteração do modelo. Existe uma variedade de trabalhos que utilizam LOTOS para desenvolvimento de protocolos com os realizados por Bagatelli *et al.* [Bagatelli et al., 2002] e Leduc *et al.* [Leduc and Germeau, 2000].

Este trabalho constitui o nosso primeiro passo na tentativa de prover uma comunicação anônima para redes ad-hoc. Estamos caminhando para o aperfeiçoamento do algoritmo, com objetivo de alcançar um modelo mais completo do protocolo. Em futuros trabalhos, planejamos avaliar o desempenho do protocolo.

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)*.

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. In *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. volume 1102, pages 437–440, New Brunswick, NJ, USA. Springer-Verlag.

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). In *Work in progress, draft-ietf-manet-dsr*. IEEE.

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. In *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.

Pfitzmann, A. and Köhntopp, M. (2000). Anonymity, unobservability and pseudonymity - a proposal of terminology. volume 2009 of *Lecture Notes in Computer Science*, Berkeley, CA, USA. Springer-Verlag, Berlin, Germany.

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.