Um Mecanismo de Segurança para Redes DTN

Cristiane B. Piaia, Cristina M. Nunes [1]

[1]Instituto de Informática
Pontifícia Universidade Católica do Rio Grande do Sul (PUC-RS)
Av. Ipiranga, 6681– Porto Alegre – RS – Brazil
cristiane.piaia@acad.pucrs.br, cristina.nunes@pucrs.br

Resumo. *A segurança em redes Delay Tolerant Networks (DTN) foi o foco deste trabalho. As DTN estão em crescente desenvolvimento, e a segurança das mesmas é um dos pontos que apresenta lacunas. Este artigo apresenta um mecanismo que utiliza algoritmo RSA para criptografar os dados, onde a confidencialidade do nodo enviador da mensagem é garantida.*

1. Introdução

As redes DTN (*Delay Tolerant Network*) são redes sem fio que utilizam o ar como meio de propagação, tendo como principais características [Carina T. de Oliveira 2007]:

1. possíveis longos e/ou variáveis atrasos, podendo variar de poucos segundos até horas ou dias;
2. frequentes desconexões, devido a movimentação dos nodos, provocando mudanças na topologia das redes.

Segundo [Jain et al. 2004], em DTN as desconexões são tão frequentes que pode não existir um caminho pré-estabelecido na rede entre um nodo origem e um determinado nodo destino, que desejam se comunicar. Para que a entrega das mensagens ocorra em tais redes, a mobilidade dos nodos é usada como forma de repasse. Esse mecanismo recebeu o nome de *store-carry-forward* [Warthman 2003], sendo que os nodos intermediários podem armazenar mensagens por um longo tempo. Quando um nodo encontra outro, eles compartilham as mensagens que encontram-se armazenadas em seus *buffers*. Esse processo ocorre até que a mensagem seja entregue ao destino.

Outra característica dessas redes refere-se aos protocolos utilizados na mesma. Como a rede é suscetível a atrasos e desconexões, protocolos normalmente utilizados na Internet, como por exemplo TCP/IP (*Transmission Control Protocol/Internet Protocol*), não são adequados. Esses protocolos foram projetados para trabalhar em redes onde existe conectividade fim a fim entre origem e destino [Nunes et al. 2010], o que não é garantido em DTN.

Alguns exemplos de projetos que utilizam redes DTN são:

1. ZebraNet [Juang et al. 2002]: projeto de pesquisa, desenvolvido pela universidade de Princeton, onde um colar com GPS (*Global Positioning System*) e memória *flash* é colocado em zebras. O colar com GPS informa uma estação-base móvel (podem ser carros conduzidos por pesquisadores) as coordenadas referentes à localização dos animais a cada três minutos.
2. InterPlaNet [Akyildiz et al. 2003]: projeto financiado pela DARPA (*Defense Advanced Research Projects Agency*), grupo de pesquisa da NASA, que tem como

objetivo desenvolver uma arquitetura e protocolos que possibilitem a comunicação entre a Terra e naves espaciais em trânsito ou até mesmo outros planetas.

3. *Drive-thru Internet* [Ott and Kutscher 2004]: projeto realizado pela Universidade de Tecnologia de Helsinki que visa prover acesso à Internet para veículos que encontram-se em movimentação nas estradas, sendo que a velocidade desses veículos pode chegar a quase 200 km/h.

Os requisitos de segurança para DTN são diferenciados de redes tradicionais ou até mesmo de outras redes sem fio [Zhu 2009]. A maioria dos métodos de segurança atuais utilizam autenticação dos usuários e integridade das mensagens. Contudo, diferentemente das demais redes, em DTN os roteadores e nodos intermediários também necessitam de autenticação. Essa autenticação se torna interessante em função dos mesmos receberem as mensagens que serão transmitidas para novos nodos até que elas cheguem em seu destino. Criptografia assimétrica, chaves públicas e autoridades certificadoras (CA) [Durst 2002] também podem ser utilizadas para garantir a segurança em redes DTN.

A segurança em DTN deve ser tratada de forma diferente de outras redes devido a falta de conectividade fim a fim entre os nodos origem e destino, característica já mensionada anteriormente como um dos aspectos peculiares de DTN. A fragmentação das mensagens, escassez de recursos e o acúmulo de mensagens são outros motivos importantes [Carina T. de Oliveira 2007] .

De acordo com [Symington 2009], possíveis ameaças de segurança que podem existir em DTN são ataques de modificação nas mensagens com um intuito de mascarar o atacante ou alterar o conteúdo (*payload*) da mensagem, acessos não autorizados de aplicações para assim tomar posse da infra-estrutura da rede e ataques de inclusão de pacotes.

Visando a confidencialidade do nodo enviador da mensagem, este trabalho propõe e desenvolve um mecanismo de segurança para redes DTN. A partir de testes realizados com um simulador projetado para redes DTN, verificou-se a eficiência do mecanismo proposto.

Este documento está dividido como segue. Na Seção 2 é apresentada a descrição de trabalhos relacionados. A Seção 3 apresenta o mecanismo proposto neste trabalho. Na Seção 4 são apresentados os resultados obtidos e os aspectos observados nas simulações realizadas. Na Seção 5 estão apresentadas as conclusões finais do trabalho e os trabalhos futuros.

2. Trabalhos Relacionados

Em [Fernandes 2009] é apresentado um mecanismo de segurança baseado em troca de chaves assimétricas onde a troca é realizada a cada novo contato entre os nodos da rede. O trabalho apresenta a implementação de um algoritmo de criptografia de chaves assimétricas e um mecanismo de assinatura digital. Essas implementações foram feitas em um simulador para redes DTN, chamado ONE (*Opportunistic Network Environment*). De acordo com avaliações de desempenho realizadas em [Fernandes 2009], o mecanismo de segurança implementado não insere grande impacto no desempenho da rede e provê um nível de segurança aceitável.

Em contrapartida, [Symington 2010] acredita-se que em função dos atrasos dema-

siadamente grandes e das desconexões, as técnicas de segurança utilizadas em DTN não devem ter como base métodos que distribuam certificados e chaves encriptadas fim-a-fim, já que geralmente as chaves e os servidores que geram os certificados são criados com um tempo determinado de duração (dias ou semanas).

O trabalho apresentando em [Zhu 2009] foca-se na autenticação eficiente da mensagem, problema de incentivo e revogação de certificados. Um mecanismo chamado OBBA (*Opportunistic Batch Bundle Authentication Scheme*) foi definido com o intuito de autenticar as mensagens. O custo da autenticação do pacote foi reduzido através do uso da integração entre identidade baseada em assinaturas e técnicas de árvores *Merkle*[1]. O esquema de incentivo foi definido para estimular a cooperação entre os nodos da rede, em relação ao encaminhamento de pacotes. Sua implementação pode ser realizada de maneira totalmente distribuída, o que dificultaria vários ataques. A proposta para o método de validação de chaves públicas explora a propagação oportunista das mensagens na rede para transmitir uma lista com a revogação dos certificados (CRL - *Certificate Revocation List*)[2].

3. Mecanismo de Criptografia Proposto

Este trabalho tem como objetivo garantir a confidencialidade do nodo enviador da mensagem. Para tanto, uma técnica de criptografia assimétrica de dados é implementada. As mensagens do nodo enviador são criptografadas utilizando a chave pública do nodo destino. Quando o destino receber a mensagem, a mesma será descriptografada utilizando a chave privada do destino.

O algoritmo de chaves assimétricas utilizado neste trabalho foi proposto por [Rivest et al. 1978] e recebe o nome de RSA (*Rivest-Shamir-Adleman*). Esse nome foi atribuído em função da primeira letra dos nomes dos criadores desse método.

3.1. O Mecanismo

Quando um nodo tem interesse em enviar uma mensagem para outro nodo, o enviador aguarda até receber a chave pública do destinatário para assim criptografar o campo nodo origem da mensagem, garantindo a sua privacidade. Após a criação da mensagem, o nodo enviador utiliza um protocolo de roteamento para entregar a mensagem a seu destino.

Criação das Chaves O algoritmo de RSA foi utilizado nesse mecanismo para criptografar o campo nodo enviador da mensagem. Cada chave possui 1024 bits. Quanto maior o tamanho da chave, mais segura a mesma se torna, sendo que cada nodo é responsável pela criação de suas próprias chaves.

Conectividade com Outros Nodos Quando um nodo entra no raio de alcance de outro nodo, eles conectam-se e trocam todas as chaves que conhecem até o momento, inclusive suas próprias chaves públicas, armazenando-as em uma lista. Em função da troca de todas as chaves conhecidas, o processo de divulgação das chaves se torna mais rápido.

[1] *Merkle tree* é uma árvore de *hash* em que as folhas são o valor de *hash* de chaves individuais. O valor de *hash* de um determinado ramo da árvore é calculado a partir do valor de *hash* das suas folhas [Szydlo 2004].

[2] CRL é um arquivo que contém uma lista de certificados revogados, seus números de série e suas respectivas datas em que ocorreram a revogação. Um arquivo da CRL também contém o nome do emissor da CRL e a próxima data de atualização. Por padrão, o menor período de validade de uma CRL é uma hora.

Geração das Mensagens No momento em que um nodo deseja enviar uma mensagem para um outro nodo, o enviador espera até receber a chave pública do destinatário para assim encriptar o campo nodo enviador da mensagem. Dessa forma, apenas o nodo destino poderá saber quem enviou-lhe a mensagem, possibilitando a confidencialidade do nodo origem. Quando as mensagens são replicadas na rede, elas já estão com seu nodo enviador encriptado, assim os outros nodos que receberem a mensagem não saberão quem a enviou.

Recebimento da Mensagem No momento em que o nodo destino recebe a mensagem encriptada, o mesmo decripta o campo nodo origem utilizando sua própria chave pública, descobrindo assim qual nodo enviou-lhe a mensagem. Caso um nodo que não é o destinatário da mensagem tente abri-la, o mesmo não obterá sucesso, pois a chave pública do destino foi usada na criptografia da mensagem. Dessa forma, apenas o nodo destino, com sua própria chave privada, obterá êxito na decodificação das mensagens.

4. Avaliação Experimental

Os resultados obtidos utilizando o método descrito na Seção 3 estão descritos a seguir.

4.1. Simulação

Para diversos tipos de redes são utilizados simuladores para demonstrar o comportamento das mesmas. Para que seja possível analisar o comportamento dos nodos utilizando a técnica de segurança proposta, foi utilizado um simulador para redes DTN. Tal simulador é chamado de ONE (*Opportunistic Network Environment*) [Keränen et al. 2009], sendo bastante referenciado na literatura [Nunes et al. 2010], [Zhu 2009] e [Fernandes 2009].

Simulador One O simulador ONE foi definido pela Universidade de Helsinki na busca de criar um simulador que atendesse todas as necessidades não supridas pelos demais simuladores disponíveis até então. ONE combina modelagem de mobilidade, roteamento e visualização do movimento dos nodos em uma rede DTN. ONE é implementado em Java e possui alguns protocolos de roteamento utilizados em redes DTN, entre eles estão os protocolos: Epidêmico [Vahdat and Becker 2000], PRoPHET [Lindgren et al. 2003] e Spray and Wait [Spyropoulos et al. 2008], o que o torna ainda mais atraente.

Para o desenvolvimento do mecanismo de segurança proposto, as bibliotecas com suporte para grandes números inteiros (*Big Integer*) e geração de números randômicos foram necessárias para a implementação do algoritmo de RSA.

As classes *Message, MessageCreateEvent, Connection, MessageRouter, DTNHost* e *ActiveRouter* originais do simulador foram alteradas. Uma classe que contém a implementação do algoritmo RSA e uma segunda classe que possui as chaves públicas foram incluídas no simulador.

4.2. Cenário para Simulação

O cenário utilizado nos testes realizados está descrito a seguir. Alguns dos parâmetros utilizados são comumente referenciados em simulações de redes DTN na literatura [Nunes et al. 2010] e [Keränen and Ott 2007].

1. Duração da simulação: a simulação ocorre por aproximadamente 12 horas. Esse valor é o padrão do simulador;

2. Estratégia das filas: a estratégia utilizada nas filas dos nodos é FIFO (*First In First Out*) [Nunes et al. 2010];
3. Número de rodadas de simulações: 7 rodadas de simulação. Os resultados apresentados nesta seção foram obtidos pela média de todas as rodadas, com um intervalo de confiança de 95% [Jain 1991];
4. Padrão de movimentação: o padrão utilizado é o *Shortest Path Map Based Movement* [Keränen and Ott 2007]. Esse padrão é um exemplo de padrão de movimentação que possui um comportamento realístico, ou seja, os nodos irão se comportar desviando dos obstáculos, caso os mesmos existam. Os obstáculos, como paredes e calçadas, são herdados dos mapas utilizados na simulação [Ekman et al. 2008];
5. Protocolos de Roteamento: os protocolos utilizados para a simulação, foram o *Spray and Wait*, *Epidemic*, *Prophet* e *APRP-Ack* [Nunes et al. 2009];
6. Quantidade de nodos: a quantidade de nodos definida é 60 [Nunes et al. 2010];
7. Raio de alcance: o raio de alcance para a transmissão das mensagens entre os nodos é 30 metros [Nunes et al. 2010]. Valor caracteriza um raio usual em dispositivos móveis;
8. Tamanho do *Buffer*: o tamanho especificado para o *Buffer*, que armazena as mensagens foi 5MBytes. Esse é o valor padrão do simulador;
9. Tipos dos nodos: metade dos nodos representam pessoas que utilizam dispositivos móveis e a outra metade representam automóveis. Ambos são tipos tradicionais em um perímetro urbano [Nunes et al. 2010];
10. TTL (*Time To Life*): o tempo de vida das mensagens é de 5 horas;
11. Velocidade de transmissão: a velocidade definida é a padrão do simulador, 250kbps. Essa é uma velocidade aceitável para dispositivos móveis.

4.3. Resultados

Para analisar o mecanismo proposto uma série de simulações foram realizadas. As mesmas analizaram a diferença encontrada entre nodos com o mecanismo de segurança em uso e sem o mesmo.

A primeira série de simulações ocorreu com o *warmap*[3] em 500 segundos e a segunda série de testes com 1000 segundos. Serão apresentados apenas os resultados obtidos com *warmap* em 500 segundos pois não ocorreram alterações significativas nos resultados quando foram utilizados 1000 segundos.

Tabela 1. Resultados com o método proposto.

	Spray and Wait	**Epidemic**	**Prophet**	**APRP-Ack**
Número de mensagens criadas	995,14	995,14	995,14	969
Número de mensagens entregues	687,85	261,85	344,57	789,28
Overhead	6,91	489,37	138,22	4,30

Os resultados obtidos nas simulações podem ser visualizadas nas Tabelas 1 e 2.

[3] *Warmap* é o tempo em que os nodos ficam apenas se conhecendo na rede, antes de começarem a trocar mensagens.

Tabela 2. Resultados sem o método proposto.

	Spray and Wait	Epidemic	Prophet	APRP-Ack
Número de mensagens criadas	1000	1000	1000	1000
Número de mensagens entregues	688,42	259,42	335,14	820,142
Overhead	6,94	504,99	142,19	4,29

Quantidade de Mensagens Criadas A quantidade de mensagens criadas quando o método proposto está em uso foi de 995,14 mensagens (com intervado de confiança entre 993,68 e 996,59) para o protocolo *Spray and Wait*, 995,14 mensagens (com intervalo de confiança de 993,68 e 996,59) para o protocolo Epidêmico, 995,14 mensagens (com intervalo de confiança de 993,68 e 996,59) para o protocolo *Prophet* e 969 mensagens (com intervalo de confiança entre 963,45 e 974,54) para o protocolo APRP-Ack. Nas simulações executadas sem a utilização do método proposto, 1000 mensagens foram criadas com todos os protocolos. Essa discrepância ocorre em função das mensagens só serem geradas quando o nodo enviador da mensagem conhece a chave pública de seu destinatário.

Quantidade de Mensagens Entregues Com a utilização do método proposto a quantidade da mensagens entregues foi de 687,85 (com intervalo de confiança entre 670,20 e 705,51) para o protocolo *Spray and Wait* e 789,28 (com intervalo de confiança entre 766,25 e 812,31) para o protocolo APRP-Ack. Quando o método não está em uso 688,42 mensagens foram entregues (com intervalo de confiança de 670,18 e 706,66) para o protocolo *Spray and Wait* e 820,14 para o APRP-Ack (com intervalo de confiança entre 799,41 e 840,86). Essa diferença ocorre em função de um número menor de mensagens serem criadas. Para os protocolos Epidêmico e *Prophet* foram entregues 261,85 e 344,57 mensagens, respectivamente (com intervalo de confiança entre 247,12 e 276,59 e entre 331,79 e 357,35) quando o método estava sendo utilizado. Quando o método não foi utilizado, aproximadamente 259,42 e 335,14 mensagens foram entregues (com intervalo de confiança entre 255,27 e 267,58 e entre 327,17 e 343,10). O aumento na entrega das mensgens quando o método estava em uso ocorre em função do comportamento dos protocolos Epidêmico e *Prophet*. O primeiro distribui livremente as mensagens pela rede e o segundo analisa a probabilidade de sucesso. Dessa forma, quando o método está em uso um menor número de mensagens são criadas, menos mensagens são distribuídas, e menos mensagens são armazenadas nos *buffers*, o que pode resultar em um consumo menor nos recursos da rede e aumentar a quantidade de mensagens entegues.

Overhead Quando o método proposto estava em uso o *overhead* ficou em 6,91 mensagens (com intevalo de confiança entre 6,73 e 7,08) para o protocolo *Spray and Wait*, 489,37 mensagens (com intervalo de confiança entre 457,41 e 521,33) para o protocolo Epidêmico, 138,22 mensagens (com intervalo de confiança de 126,95 e 149,49) para o protocolo *Prophet* e 4,30 mensagens (com intervalo de confiança entre 4,21 e 4,38) para o protocolo APRP-Ack. Já quando o método não estava em uso houve 6,94 mensagens de *overhead* (com intevalo de confiança entre 6,75 e 7,12) para o protocolo *Spray and Wait*, 504,99 mensagens (com intervalo de confiança de 478,60 e 531,38) para o protocolo Epidêmico, 142,19 mensagens (com intervalo de confiança entre 136,48 e 147,90) para o protocolo *Prophet* e 4,29 mensagens (com intervalo de confiança entre 4,20 e 4,37) para

o protocolo APRP-Ack.

5. Conclusão

As redes DTN são redes sem fio com características únicas como frequentes desconexões e possíveis atrasos na entrega de suas mensagens. Um de seus desafios é a segurança. Mesmo já existindo soluções aceitáveis para redes tradicionais e até mesmo para redes sem fio como em *ad hoc*, essas soluções não apresentam resultados admissíveis para DTN. Em função disso, este trabalho buscou estudar trabalhos realizados na área e apresentar uma solução para o problema de confidencialidade do nodo enviador da mensagem.

O mecanismo proposto criptografa o endereço do nodo origem da mensagem com a chave pública do nodo que deve receber a mesma. Dessa forma, o mecanismo garante que apenas o verdadeiro nodo destinatário da mensagem identifica o nodo enviador.

Acredita-se que o método não apresentou impacto negativo no desempenho da rede, um ponto muito importante em função da distribuição das chaves pela rede. Com o método proposto, poucas mensagens não foram entregues com os protocolos *Spray and Wait* e APRP-Ack, o que é bastante aceitável em função do ganho obtido com a segurança do nodo enviador. Já com os protocolos Epidêmico e *Prophet* ocorreu um aumento na quantidade de mensagens entregues e foi garantida a segurança do nodo enviador da mensagem. Mesmo acreditando que outros testes precisam ser realizados, os resultados obtidos foram bastante positivos e acredita-se que são bastante viáveis para redes DTN.

No futuro pretende-se acrescentar ao método a possibilidade de garantir também a autenticidade do nodo enviador da mensagem. Pretende-se também realizar a comparação com outros trabalhos similares, ou seja, que também utilizam a idéia de chaves para criptografia de dados.

Referências

Akyildiz, I. F., Özgür B. Akan, Akan, O. B., Chen, C., Fang, J., and Su, W. (2003). Interplanetary internet: State-of-the-art and research challenges. *Computer Networks*, 43:75–112.

Carina T. de Oliveira, Marcelo D. D. Moreira, M. G. R. L. H. M. K. C. e. O. C. M. B. D. (2007). Redes tolerantes a atrasos e desconexões. In *Minicurso do XXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos SBRC'2007*. Sociedade Brasileira de Computação.

Durst, R. C. (2002). An infrastructure security model for delay tolerant networks.

Ekman, F., Keränen, A., Karvo, J., and Ott, J. (2008). Working day movement model. In *MobilityModels '08: Proceeding of the 1st ACM SIGMOBILE workshop on Mobility models*, pages 33–40, New York, NY, USA. ACM.

Fernandes, R. d. M. S. (2009). Proposta de um mecanismo de segurança baseado em troca de chaves assimétricas para redes tolerantes a atrasos e desconexões.

Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley-Interscience, New York, NY, 5th edition.

Jain, S., Fall, K., and Patra, R. (2004). Routing in a delay tolerant network. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 145–158, New York, NY, USA. ACM.

Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L. S., and Rubenstein, D. (2002). Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. *SIGOPS Oper. Syst. Rev.*, 36(5):96–107.

Keränen, A. and Ott, J. (2007). Increasing reality for dtn protocol simulations. Technical report, Helsinki University of Technology.

Keränen, A., Ott, J., and Kärkkäinen, T. (2009). The one simulator for dtn protocol evaluation. In *Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, pages 1–10, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Lindgren, A., Doria, A., and Schelén, O. (2003). Probabilistic routing in intermittently connected networks. *SIGMOBILE Mobile Computing and Communications Review*, 7(3):19–20.

Nunes, C. M., Dotti, F. L., and Oliveira, J. B. S. d. (2010). Aprp-group: Roteamento para redes dtn com repasse baseado em agrupamento de nodos por potencial de entrega. In *XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 451–464.

Nunes, C. M., Link, E., and Dotti, F. L. (2009). Evaluating the impact of an acknowledgment strategy for aprp. In *LANC '09: Proceedings of the 5th International Latin American Networking Conference*, pages 77–86, New York, NY, USA. ACM.

Ott, J. and Kutscher, D. (2004). Drive-thru internet: Ieee 802.1 1b for "automobile"users. In *IEEE INFOCOM 2004*, volume 1, pages 362–373. IEEE.

Rivest, R., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126.

Spyropoulos, T., Psounis, K., and Raghavendra, C. (2008). Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Transactions on Networking*, 16(1):77–90.

Symington, S Farrell, H. W. P. L. (2009). Delay-tolerant networking security overview - draft-irtf-dtnrg-sec-overview-06.

Szydlo, M. (2004). Merkle tree traversal in log space and time. In *In Eurocrypt 2004, LNCS*, pages 541–554. Springer-Verlag.

Vahdat, A. and Becker, D. (2000). Epidemic routing for partially-connected ad hoc networks. Technical report, Duke University.

Warthman, F. (2003). *Delay-Tolerant Networks (DTNs): A Tutorial v1.1. Relatório técnico*. Warthman Associates, 1th edition.

Zhu, H. (2009). *Security in Delay Tolerant Networks*. PhD thesis, University of Waterloo.