Universidade Federal de Pernambuco – UFPE
Centro de Informática - CIn

Mecanismo de Segurança para Redes Ad Hoc Móveis

Recife, 28 de fevereiro de 2014

Universidade Federal de Pernambuco – UFPE
Centro de Informática - CIn

Mecanismo de Segurança para Redes Ad Hoc Móveis

Aluno: Gregório Patriota Correia
Orientador: Djamel Fawzi Hadj Sadok
Coorientador: Thiago Gomes Rodrigues
Centro de Informática da Universidade Federal de Pernambuco
Graduando em Engenharia da Computação

Recife, 28 de fevereiro de 2014

Universidade Federal de Pernambuco – UFPE
Centro de Informática - CIn

Recife, 28 de fevereiro de 2014

Dedicatória

Cada minuto e cada segundo ouvi palavras de motivação, quando eu menos acreditei em mim todos ainda bateram de frente e afirmaram que eu iria conseguir. Os que não tinham noção da complexidade do trabalho e os que sabiam bem mais que eu, todos esses me motivaram bastante. Dedico a todos eles cada letra deste trabalho, cada cabelo branco que ganhei e cada fio de cabelo que perdi. Dedico ao meu pai, mãe e irmãos. Dedico aos meus amigos de trabalho. Mas dedico especialmente a minha avó, mulher forte do sertão. Mesmo ela que um dia me perguntou se "consertar computador dava dinheiro", mas nunca desacreditou de mim, que Deus a tenha.

"Mas nem sempre a fraqueza que se sente, quer dizer que a gente não é forte."

-Gabriel, O Pensador.

Acrônimos

AES	Advanced Encryption Standard
CBC-MAC	Cipher Block Chaining Message Authentication Code
CMAC	Cipher-Based Message Authentication Code
3DES	Triple Data Encryption Standard
DES	Data Encryption Standard
HMAC	Keyed-Hashing for Message Authentication Code
HTR	Heterogeneous Technologies Routing
IEEE	Institute of Electrical and Electronics Engineers
IPv4	Internet Protocol (version 4)
IPv6	Internet Protocol (version 6)
LTE	Long Term Evolution
MAC	Message Authentication Code
MANET	Mobile Ad hoc Network
MD5	Message Digest 5
MPR	Multipoint Relaying
ns-3	Network Simulator 3
OFDM	Orthogonal Frequency-Division Multiplexing
RC4	Rivest Cipher 4
S-HTR	Secure Heterogeneous Technologies Routing
SHA	Secure Hash Algorithm
TC	Topology Control
UDP	User Datagram Protocol
WiMAX	Worldwide Interoperability for Microwave Access

Índices de Figuras

Figura 1: Vulnerabilidades relatadas ao CERT.org 15
Figura 2: Incidentes reportados ao CERT.br de 1999 a setembro 2013 15
Figura 3: Formato do Pacote HTR 18
Figura 4: Formato da mensagem HELLO 19
Figura 5: Formato da mensagem Data encapsulada na mensagem de HELLO 19
Figura 6: Formato da mensagem TC 20
Figura 7: Cabeçalho da mensagem de controle do S-HTR 24
Figura 8: Estrutura do HMAC 25
Figura 9: Distribuição dos nós no cenário 31
Figura 10: Comparação entre os tempos de convergência para o cenário de 25 elementos 35
Figura 11: Comparação entre os tempos de convergência para o cenário de 49 elementos 35
Figura 12: Comparação entre os tempos de convergência para o cenário de 64 elementos 36
Figura 13: Comparação entre os tempos de convergência para o cenário de 81 elementos 36
Figura 14: Comparação entre as cargas de dados de controle para um cenário de 25 elementos 36
Figura 15: Comparação entre as cargas de dados de controle para um cenário de 49 elementos 37
Figura 16: Comparação entre as cargas de dados de controle para um cenário de 64 elementos 37
Figura 17: Comparação entre as cargas de dados de controle para um cenário de 81 elementos 37
Figura 18: Comparação entre a perda de pacotes de controle para um cenário de 25 elementos 38
Figura 19: Comparação entre a perda de pacotes de controle para um cenário de 49 elementos 38
Figura 20: Comparação entre a perda de pacotes de controle para um cenário de 64 elementos 38
Figura 21: Comparação entre a perda de pacotes de controle para um cenário de 81 elementos 39

Índices de Tabelas

Tabela 1: Tempo médio exigido para busca completa da chave 21
Tabela 2: Simbologia do esquema de HMAC 25
Tabela 3: Status dos Ataques em um ambiente com S-HTR 27
Tabela 4: Configuração de tecnologias 30
Tabela 5: Configuração do modelo de propagação 30
Tabela 6: Modelo de energia do chip de rádio 31
Tabela 7: Configuração dos parâmetros de simulação 32
Tabela 8: Benchmark da biblioteca Crypto++ 32
Tabela 9: Tamanho dos resumos gerados pelas respectivas funções Hash 33
Tabela 10: Benchmark completo da biblioteca Crypto++ 43
Tabela 11: Funções de Hash e suas informações 44
Tabela 12: Valor da constante a de uma amostra de tamanho n, Teste de Shapiro-Wilk 46
Tabela 13: Valores críticos da estatística W de Shapiro-Wilk 48
Tabela 14: Estatísticas do Tempo de Convergência para uma rede de 25 nós 50
Tabela 15: Estatísticas da Sobrecarga de Pacotes para uma rede de 25 nós 50
Tabela 16: Estatísticas da Perda de Pacotes para uma rede de 25 nós 50
Tabela 17: Estatísticas do Tempo de Convergência para uma rede de 49 nós 50
Tabela 18: Estatísticas da Sobrecarga de Pacotes para uma rede de 49 nós 51
Tabela 19: Estatísticas da Perda de Pacotes para uma rede de 49 nós 51
Tabela 20: Estatísticas do Tempo de Convergência para uma rede de 64 nós 51
Tabela 21: Estatísticas da Sobrecarga de Pacotes para uma rede de 64 nós 51
Tabela 22: Estatísticas da Perda de Pacotes para uma rede de 64 nós 51
Tabela 23: Estatísticas do Tempo de Convergência para uma rede de 81 nós 52
Tabela 24: Estatísticas da Sobrecarga de Pacotes para uma rede de 81 nós 52
Tabela 25: Estatísticas da Perda de Pacotes para uma rede de 81 nós 52

Índices de Equações

Equação 1: Métrica utilizada pelo protocolo HTR, HTRScore 17
Equação 2: Função genérica de Hash 22
Equação 3: Estatística W para o teste de normalidade de Shapiro-Wilk 34
Equação 4: Duas amostras independentes pareadas 34
Equação 5: Hipóteses para o Teste de Wilcoxon 34

Agradecimentos

Esse trabalho pode render bons frutos para a academia. Não conseguiria desenvolver algo desse valor sem a ajuda da equipe com quem trabalho e da influência dos que me orientam.

Agradeço ao Grupo de Pesquisa em Redes e Telecomunicações (GPRT) e a minha equipe de trabalho que acima do profissionalismo foram amigos: Thiago Rodrigues, Josias Junior e Rodrigo Melo. Agradeço a minha família que sempre me apoiou, ao meu orientador Djamel Sadok, à professora Judith Kelner e a Eduardo Feitosa.

Mas devo um agradecimento especial a Josias Junior pela paciência em me ensinar toda a estatística necessária contida neste trabalho e a ainda a maior paciência de todas de Thiago Rodrigues que desde o desenvolvimento do sistema à conclusão da escrita fez-se presente ajudando para que eu tivesse neste trabalho os melhores resultados possíveis.

Aos meus amigos de estudo, pessoas com quem quero sempre manter a convivência, devo a eles muito mais que agradecimentos neste trabalho. A todos eles, deixo aqui meu simples e sincero: Muito obrigado.

Pai e mãe, outo de mina. Irmão e irmã foi presente de Deus. Serei eternamente grato a eles também.

Sumário

Resumo 13
1 Introdução 14
1.1 Contextualização 14
1.2 Motivação 14
1.3 Heterogeneous Technologies Routing 16
1.3.1 Funcionalidades 16
1.3.2 HTRScore 17
1.3.3 Formato dos Pacotes 17
1.3.4 Tipos de Mensagens 18
1.3.4.1 Mensagem de HELLO 18
1.3.4.2 Mensagem de TC 20
1.4 Segurança 20
1.4.1 Cifras Simétricas 21
1.4.2 Funções de Hash 22
1.4.3 MAC 22
1.4.4 HMAC 23
1.4.5 CMAC 23
2 Objetivos 24
2.1 Descrição do Problema 24
2.2 Arquitetura 24
3 Estado da Arte 26
3.1 Ataques 26
3.1.1 Flooding Attack 26
3.1.2 Sleep Depravation 26
3.1.3 Impersonation Attack 26
3.1.4 Black Hole Attack 26
3.1.5 Node Isolation Attack 26
3.1.6 Routing Table Poisoning Attack 26
3.1.7 Wormhole Attack 27
3.1.8 Location Disclosure Attack 27
3.1.9 Rushing Attacks 27
3.1.10 Blackmail Attack 27

3.1.11 Snare Attack 27
3.1.12 The Invisible Node Attack 27
3.2 S-HTR contra os Ataques 27
3.2.1 Ataques Mitigados 28
3.2.1.1 Flooding Attack 28
3.2.1.2 Black Hole Attack 28
3.2.1.3 Routing Table Poisoning Attack 28
3.2.1.4 Rushing Attack 28
3.2.2 Ataques Não Mitigados 28
3.2.2.1 Impersonate Attack 28
3.2.2.2 Node Isolation Attack 28
3.2.2.3 Wormhole Attack 28
3.2.2.4 Location Disclosure Attack 28
3.2.2.5 Snare Attack 28
3.2.2.6 The Invisible Node Attack 28
3.2.3 Ataques que Não se Enquadram a Natureza do HTR 29
3.2.3.1 Sleep Depravation 29
3.2.3.2 Blackmail Attack 29
4 Metodologia 30
4.1 Ambiente 30
4.2 Cenário 31
4.3 Métricas 32
4.3.1 Tempo de Convergência 32
4.3.2 Sobrecarga das Mensagens 33
4.3.3 Perda de Pacotes 33
4.4 Testes de Hipótese 33
4.4.1 Teste de Shapiro-Wilk 34
4.4.2 Teste de Wilcoxon Signed-Rank 34
5 Resultados 35
5.1 Tempo de Convergência 35
5.2 Sobrecarga das Mensagens 36
5.3 Perda de Pacotes 38
5.4 Discussão 39

6 Conclusão 41
7 Trabalhos Futuros 42
8 Apêndice A 43
9 Apêndice B 46
10 Apêndice C 50
11 Bibliografia 53

Resumo

A expansão das redes de computadores proveu grande integração e conectividade entre computadores existentes em várias partes do mundo. Essa alta conectividade trouxe a tona a necessidade de sistemas para prover segurança aos dados que trafegam nesta rede. Segurança tornou-se um requisito essencial para todo e qualquer sistema em rede, para as redes ad hoc não é diferente. Cada vez mais com maior aplicabilidade, redes ad hoc têm grandes restrições, o que inviabiliza a aplicação de mecanismos comuns de segurança, restrições tais como banda de passagem de tráfego e autonomia energética.

O *Heterogeneous Technologies Routing* (HTR) foi um protocolo desenvolvido para redes ad hoc que não oferecem mecanismo seguro. Este protocolo provê um roteamento eficiente de dados com o menor consumo energético, porém seus dados trafegam em texto claro (originais) entre os elementos que compõem a rede, ou seja, sem oferecer segurança.

Este trabalho tem como foco avaliar propor um mecanismo de segurança para o *framework* para redes ad hoc móvel HTR e avaliar o impacto causado por um sistema de segurança a mesma. Este mecanismo desenvolvido foi denominado de S-HTR. O objetivo foi encontrar a melhor forma de prover integridade e autenticidade aos dados com o menor custo de banda e processamento possível no HTR, usando um mecanismo baseado em *Keyed-Hashing for Message Authentication Code* (HMAC) foi avaliado o impacto causado entre cinco funções de *Hash* diferentes: MD5, SHA-1, SHA-256, SHA-384 e SHA-512. Avaliando métricas como tempo de convergência da rede, sobrecarga dos pacotes e perda de pacotes, conclui-se que o esquema de HMAC com a função de *Hash* SHA-1 obteve os melhores resultados, garantindo integridade e autenticidade aos dados da rede e com o menor impacto sobre: o tempo de convergência da rede, a taxa de perda de pacotes e o aumento na sobrecarga de pacotes.

Palavras-chave: Ad Hoc; Hash; HMAC; HTR, MANET; Segurança; Simulação.

1 Introdução

1.1 Contextualização

No atual cenário tecnológico, há uma gama de tecnologias de comunicação disponíveis: Wi-Fi, Ethernet, Bluetooth, LTE, WiMAX, Infravermelho e Zig Bee [1]. Melhorias contínuas e novas tecnologias aprimoram cada vez mais a qualidade e a eficiência das comunicações e permitem que novas aplicações surjam.

Em paralelo às melhores tecnologias de comunicação, há o desenvolvimento de hardwares de grande poder computacional e mais compacto, o que nos trouxe dispositivos móveis acessíveis com grande autonomia, robustez e poder computacional. A maioria destes dispositivos agregam distintas tecnologias de comunicação.

A mobilidade destes dispositivos associado à disponibilidade de recursos de comunicação possibilitou o desenvolvimento das MANET (Mobile Ad Hoc Network)[2]. Redes com alto grau de mobilidade e adaptação a ambientes adversos, assumindo qualquer tipo de topologia.

A falta de uma estrutura física e elementos centrais que coordenem o tráfego representam aspectos positivos e negativos para as MANET´s. Representa uma vantagem quando levamos em conta o custo necessário para instalação e manutenção da infraestrutura da rede, mobilidade da rede, facilidade na construção de redes espontâneas, descentralização e robustez[3]. Porém, a falta de um elemento que faça o balanceamento do tráfego representa uma desvantagem, assim como a alta taxa de erros e a baixa taxa de transmissão[3]. Um problema crítico nessas redes é a segurança.

1.2 Motivação

Segurança da informação tornou-se um requisito essencial desde 1994, como especificado em [4]. Identificando as áreas principais que necessitavam de um mecanismo de segurança, este documento estabelecia um consenso geral de que a Internet precisava de mais e melhor segurança. Os autores estabeleceram como prioridades a proteção da infraestrutura da rede contra monitoração e controle não autorizado do tráfego da rede, e a necessidade de proteger o tráfego de usuário final, usando mecanismo de autenticação e criptografia.

O *Computer Emergency Response Team Coordination Center* (CERT/CC) ilustrou os dados relatados nos últimos anos de vulnerabilidades e de incidentes. As vulnerabilidades relatadas (veja Figura 1) incluem pontos fracos de segurança nos sistemas operacionais dos computadores que sofreram ataques, além das vulnerabilidades nos roteadores da Internet e outros dispositivos de rede [5]. Já os incidentes relatados [6](ilustrado na Figura 2), incluem ataques de negação de serviço, falsificação de IP e outras diversas formas de comportamento malicioso [7].

Apesar de existirem vários esquemas de criptografia disponíveis, nenhum deles é diretamente aplicado às redes MANET's [8].

As redes ad hoc móveis possuem grandes restrições de energia e de banda, a implementação de um sistema seguro tradicional nestas redes pode provocar um grande impacto negativo, diminuindo o tempo de vida dos dispositivos na rede e causando congestionamento dos enlaces.

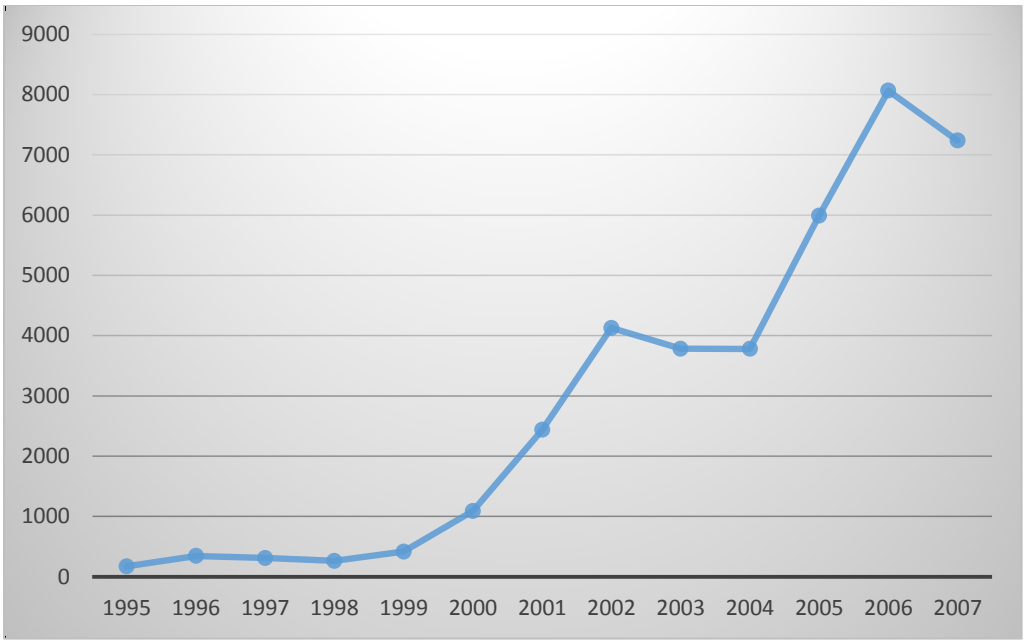

Figura 1: Vulnerabilidades relatadas ao CERT.org

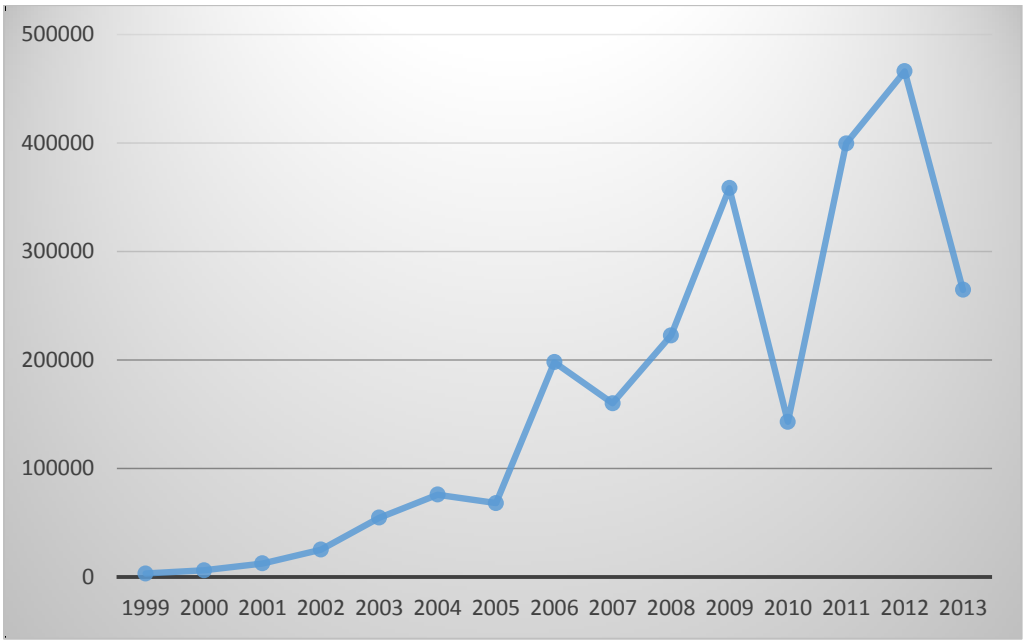

Figura 2: Incidentes reportados ao CERT.br de 1999 a setembro 2013

Um sistema é considerado seguro quando respeita dois conceitos fundamentais, como descritos em [7]:

- O custo de cripto-análise nunca deve ser menor que o custo da informação;
- O tempo para que seja executada a cripto-análise nunca deve ser menor que o tempo de vida da informação.

O desafio está em desenvolver um sistema que alcance os conceitos fundamentais da segurança focando em redes ad hoc móveis.

1.3 Heterogeneous Technologies Routing

Desenvolvido para redes ad hoc, o *Heterogenous Technologies Routing* (HTR) [9], [10] é um protocolo de roteamento proativo que faz uso de múltiplas interfaces de rede de tecnologias diferentes em um único elemento da rede. Os elementos de uma rede ad hoc HTR apresentam um endereçamento único para as múltiplas interfaces de comunicação, permitindo assim uma única identificação dentro da rede. O HTR adota um método de distribuição de endereçamento único proposto em [11], este método tem como objetivo garantir que o endereço IP distribuído é único dentro da rede. Em [12] o HTR alcançou melhores tempos de convergência, esta meta foi alcançada a partir de testes que buscaram o melhor intervalo entre as mensagens de controle HELLO. Um modelo do HTR com uma abordagem *multipath* foi desenvolvido em [13] para encontrar melhores caminhos no processo de roteamento que foi proposto nos trabalhos anteriores.

O HTR herda alguns conceitos do protocolo OLSR como o *Multipoint Relaying* (MPR) [14], as mensagens de controle HELLO e TC e o algoritmo de cálculo de melhor caminho, o *Djikstra* [15]. O MPR foi uma solução para controlar a inundação de tráfego causado pelas mensagens de controle (HELLO e *Topology Control* - TC). Entretanto, o HTR estabelece métricas adicionais como informações de qualidade de enlace, compatibilidade de dispositivos ou papeis humanos, custo de envio de um bit por determinada interface, para selecionar um nó MPR e como ponderador do algoritmo de melhor caminho *Djikstra*. Este ponderador foi chamado de HTRScore e é responsável pelo controle energético dos nós da rede, pois o algoritmo de roteamento escolherá sempre o nó que tem o maior HTRScore para rotear o pacote e à medida que a energia dos nós forem variando, outros nós serão escolhidos. Esta estratégia proporciona um balanceamento energético entre os nós da rede HTR proporcionando um maior tempo de vida da rede.

1.3.1 Funcionalidades

O HTR opera de acordo com um conjunto bem definido de funcionalidades, em ordem, para tornar possível o roteamento. Essas funcionalidades especificam o comportamento dos nós que participam da MANET e executam o HTR como seu protocolo de roteamento. As principais funcionalidades do HTR são:

- **Endereçamento do Nó**: este protocolo utiliza apenas um único endereço IP para identificar o nó na rede, em virtude de um nó poder operar com múltiplas interfaces de rede. O HTR pode ser usado tanto com IPv4 quanto IPv6.
- **Sensoriamento do enlace**: esta funcionalidade mantém atualizada a informação de quais enlaces existem entre um nó e seu vizinho. Isto acontece através da emissão periódica da mensagem de controle HELLO.
- **Detecção de vizinhos**: esta funcionalidade permite que os nós descubram ou deduzam a existência de vizinhos por meio de informações trocadas como parte da operação de sensoriamento de enlace. Através também dessa funcionalidade as informações são armazenadas em um repositório chamado de *Link-Neighbor*.
- **Seleção e sinalização de MPR**: a seleção de um nó para MPR tem como objetivo selecionar um conjunto de vizinhos que garante que as mensagens em *broadcast* sejam retransmitidas para todos os nós com dois saltos de distância. A seleção correta desses nós é um elemento fundamental nas funcionalidades do HTR.

- **Difusão das mensagens de *Topology Control***: As mensagens *Topology Control* são enviadas em *broadcast* dentro da rede a fim de fornecer informações suficientes de cálculo de rota para cada nó da rede.
- **Cálculo de Rota**: esta funcionalidade permite que cada nó possa computar sua tabela de roteamento, de acordo com as informações de qualidade de enlace obtidas, através das mensagens de trocas que ocorrem periodicamente.

1.3.2 HTRScore

Uma das diferenças entre o HTR e outros protocolos de redes ad hoc, é que o HTR tem como foco manter a rede operando pelo maior tempo possível. Isto pode ser alcançado pelo uso da métrica HTRScore que é definida usando fatores como estado do enlace e eficiência energética. Esta métrica é usada para melhorar o cálculo de rota que também é aplicável ao conjunto de cálculos do MPR.

HTRScore também introduz a probabilidade de perda de pacotes, um parâmetro que avalia a estabilidade do link, um otimização que beneficia caminhos com alta taxa de emissão de pacotes. A Equação 1 é a definição matemática do HTRScore.

$$HTRScore(i,j) = \frac{e_{i,j}^{\alpha}}{(1 - \rho_{i,j})^{\beta}} x \frac{\mathrm{E}_i^{\gamma}}{R_i^{\theta}}$$

Equação 1: Métrica utilizada pelo protocolo HTR, HTRScore

Onde i é o nó fonte; j é o nó destino; $e_{i,j}$ é a energia de transmissão necessária para o nó i transmitir uma unidade de informação para o vizinho j; $\rho_{i,j}$ é a probabilidade de perder um pacote enviado pelo enlace que liga o nó i ao nó j; R_i é a energia residual do nó i; e E_i é a energia inicial da bateria do nó i.

Os parâmetros α, β, γ e θ representam fatores de peso não negativos.

1.3.3 Formato dos Pacotes

O Pacote HTR pode conter uma ou mais mensagens por pacote, mensagens essas que podem ser tanto mensagens de HELLO quanto de TC. O formato do pacote HTR é ilustrado na Figura 3. Os campos do pacote HTR são descritos a seguir:

- **Packet Lengh**: tamanho (em bytes) do pacote, incluindo o tamanho do cabeçalho.
- **Packet Sequence Number**: número de sequência incrementado por um cada vez que um novo pacote HTR deve ser transmitido (16 bits). O número de sequência é usado para identificar se um determinado pacote já foi processado.
- **Message Type**: um valor que identifica o tipo da mensagem (8 bits). Similar ao OLSR, HTR reserva o espaço 0 – 127 para os tipos de mensagens, enquanto que o espaço 128 – 255 é reservado para futuras extensões.
- **Validity**: informa quanto tempo depois da recepção um nó deve considerar válida a informação contida no pacote (8bits).
- **Message Size**: indica o tamanho da mensagem (em bytes), incluindo o cabeçalho da mensagem (16 bits).
- **Originator Address**: endereço do nó que originalmente gerou a mensagem (32 bits). Este campo não deve ser misturado com o endereço fonte do cabeçalho IP, que é trocado cada vez que a mensagem é retransmitida.

- **Time To Live (TTL)**: identificador com o máximo número de saltos que a mensagem pode ser encaminhada (8 bits).
- **Hop Count**: informa o número de vezes que uma dada mensagem foi retransmitida (8 bits).
- **Message Sequence Number**: o número de sequência é incrementado de 1 cada vez que uma nova mensagem HTR é transmitida (16 bits). O número de sequência previne a retransmissão de duplicatas.
- **Message**: pode ser tanto uma mensagem de HELLO quanto uma mensagem de TC.

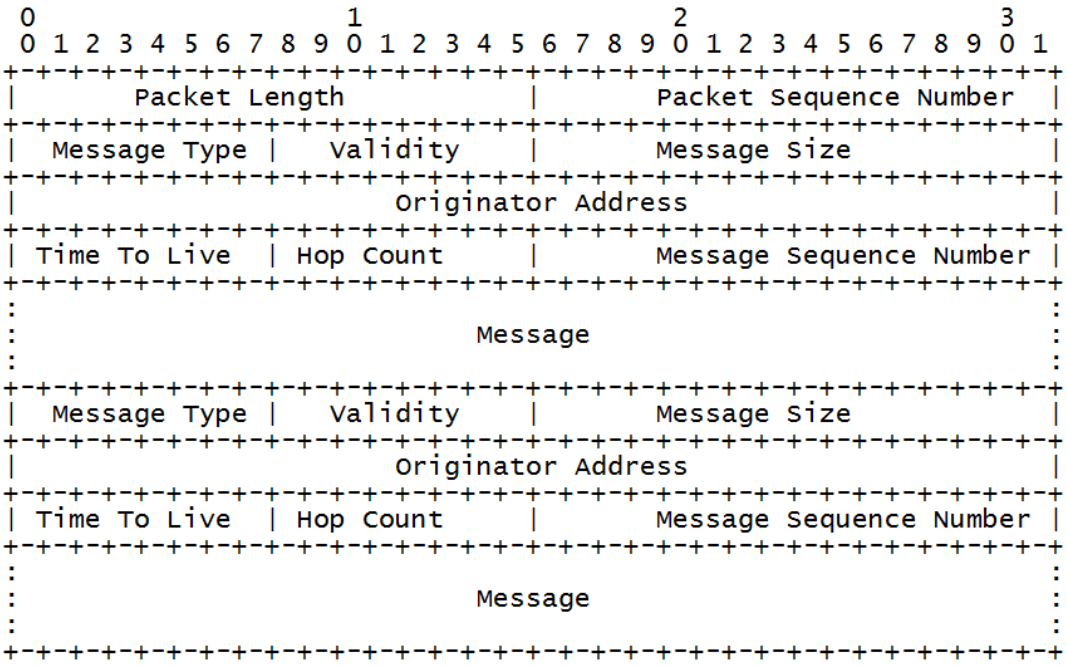

Figura 3: Formato do Pacote HTR

1.3.4 Tipos de Mensagens

HTR define apenas dois tipos de mensagens: HELLO e *Topology Control* (TC). Todas as funcionalidades do HTR estão baseadas na geração e processamento dessas duas mensagens. Entretanto, o formato dos pacotes do protocolo HTR permite uma grande variedade de mensagens para serem transmitidas, dependendo apenas da necessidade. Essas mensagens são propagadas periodicamente pelos elementos que compõem a rede, como forma de manter todos os nós atualizados sobre informações e topologia da rede. Essas mensagens são enviadas a cada intervalo de 5 segundos e as informações têm uma validade de 30 segundos dentro dos repositórios.

1.3.4.1 Mensagem de HELLO

As mensagens de HELLO são usadas para povoar as tabelas dos vizinhos de 1-salto com informações sobre vizinhos conhecidos e o estado do enlace. Tais informações são comprimidas dentro do cabeçalho e propagadas em *broadcast*. Os campos contidos na mensagem de HELLO são descritos na próxima subseção.

1.3.4.1.1 Formato HELLO

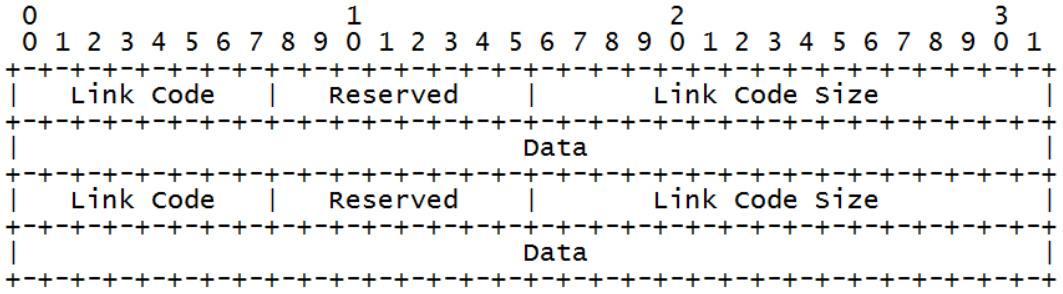

Figura 4: Formato da mensagem HELLO

- **Link Code**: este campo é uma classificação dos vizinhos de 1-salto conhecido pela origem da mensagem, e descreve o estado do enlace estabelecido entre a origem e os vizinhos de 1-salto conhecidos contidos no campo Data. Este campo pode ser dividido em dois subcampos: **Link Type** e **Neighbor Type**. O primeiro subcampo caracteriza o enlace de comunicação entre: simétrico, assimétrico ou perdido, sem comunicação. O segundo subcampo caracteriza o tipo de interação entre o vizinho pelo dado enlace, classificando entre: simétrico, assimétrico ou se é um vizinho MPR.
- **Reserved**: este campo foi reservado para uso futuro. Ele é preenchido com o valor inteiro "0".
- **Link Code Size**: indica o tamanho (em bytes) da mensagem, composto do campo Link Code até o próximo campo Link Code, ou se não há outro campo Link Code, até o fim da mensagem.
- **Data:** representa um campo variável, mostrado na Figura 5, composto por vizinhos de 1-salto que incluem as seguintes informações.
    - **Neighbor IP Address**: indica o endereço do vizinho de 1-salto da origem (32 bits para IPv4 e 128 bits para o IPv6).
    - **HTRScore**: representa o valor da métrica de energia usada para definir o MPR e para o cálculo de caminho. Representa o custo de transmissão da origem para o vizinho de 1-salto.

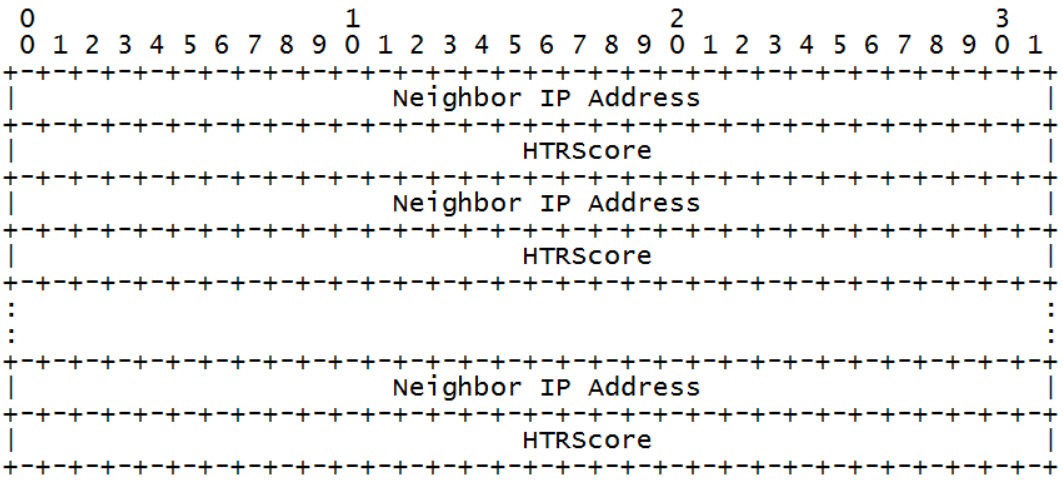

Figura 5: Formato da mensagem Data encapsulada na mensagem de HELLO

1.3.4.2 Mensagem de TC

As mensagens *Topology Control* (TC) é um mecanismo utilizado para propagar informações da topologia da rede para a rede ad hoc. O elemento fundamental nesse mecanismo é o conjunto de MPR-Selectors que é o responsável por garantir a propagação dessa informação para a rede.

1.3.4.2.1 Formato TC

O formato da mensagem TC está descrito na Figura 6.

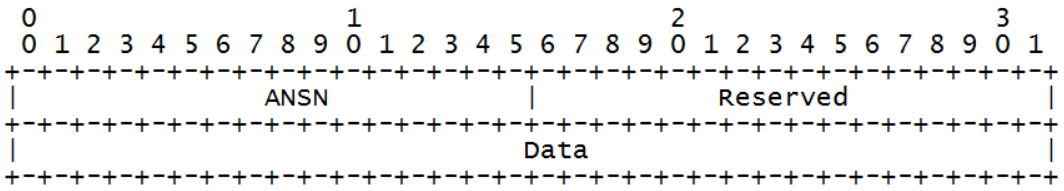

Figura 6: Formato da mensagem TC

- **ANSN**: similar no OLSR, o *Advertised Neighbor Sequence Number* (ANSN) é o campo associado ao conjunto de vizinhos que compõem o MPR-Selector. Toda vez que um nó detecta uma mudança no conjunto de vizinhos, é incrementado o número de sequência. Este número é enviado para o campo ANSN da mensagem de TC. Quando um nó recebe uma mensagem de TC, ele pode decidir, a partir do ANSN, se as informações que chegaram são mais atuais ou não quando comparadas com as que ele já possui.
- **Reserved**: este campo foi reservado para uso futuro. Ele é preenchido com o valor inteiro "0".
- **Data**: este campo utiliza a mesma estrutura do campo Data da mensagem de HELLO. Assim como na mensagem HELLO, este campo é composto pelos seguintes campos.
    - **Destination IP Address**: indica o endereço do vizinho de 1-salto da origem (32 bits para IPv4 e 128 bits para o IPv6).
    - **HTRScore**: representa o valor da métrica de energia usada para definir o MPR e para o cálculo de caminho. Representa o custo de transmissão da origem para o vizinho de 1-salto.

1.4 Segurança

Segurança da informação virou algo fundamental em todo e qualquer sistema em rede. A definição de segurança é baseada em cinco pilares:

- **Autenticidade**: garantia de que a entidade é aquela que afirma ser.
- **Confidencialidade**: proteção dos dados contra divulgação não autorizada.
- **Integridade**: garantia de que os dados recebidos estão exatamente da forma quando foram enviados por uma entidade autorizada.
- **Disponibilidade**: garantir que o sistema estará disponível para o uso de usuários autênticos, sendo limitado o acesso aos nós não autorizados.
- **Não-repúdio**: oferece proteção contra negação, por parte de uma das entidades envolvidas na comunicação, de ter participado de toda ou parte da comunicação.

As próximas subseções focam em descrever esquemas e algoritmos de segurança que serviram base para este trabalho final de curso.

1.4.1 Cifras Simétricas

A criptografia simétrica é um tipo ainda dos mais utilizado até os dias de hoje, era o único até 1970 quando foi desenvolvida a criptografia por chave pública. Seu esquema é constituído por cinco principais componentes:

- **Texto Claro**: dados originais, como entrada para o algoritmo.
- **Algoritmo de Criptografia**: algoritmo usado para executar as substituições e transformações no texto claro.
- **Chave Secreta**: entrada para o algoritmo de criptografia. É um valor independente do texto claro e do algoritmo.
- **Texto Cifrado**: mensagem embaralhada produzida como saída. Depende do texto claro e da chave secreta. O texto nesse formato é inteligível.
- **Algoritmo de Decriptografia**: é basicamente o algoritmo de criptografia executado no modo inverso. Transforma o texto cifrado em texto claro.

Os algoritmos de criptografia se dividem em Cifras de Bloco e Cifras de Fluxo. Em uma cifra de fluxo cada digito do texto em claro é encriptado com a chave, *Rivest Cipher 4* (RC4) é um exemplo de cifra de fluxo. Diferente da cifra de fluxo, a cifra de bloco encripta blocos de dados de tamanho fixo com a chave secreta, exemplo de cifras de bloco: *Advanced Encryption Standard* (AES), *Data Encrytion Standard* (DES), *Triple Data Encryption Standard* (3DES)[7].

Para garantir a segurança de um sistema baseado em criptografia simétrica é preciso um algoritmo de criptografia forte, que leve em consideração a possibilidade de um usuário malicioso (atacante) conhecer o algoritmo e amostras dos textos cifrados, e não permitir deão usuário decriptografar o texto cifrado ou obter a chave secreta.

Caso o atacante descubra a chave secreta, toda a comunicação estará comprometida. A garantia de segurança está diretamente associada à confidencialidade da chave secreta. No caso de ataque por força bruta o atacante faz uma busca por todas as possibilidades de chaves, na tentativa de conseguir uma chave a partir de um texto cifrado, e gerar uma tradução inteligível para um texto em claro. Na média, metade de todas as chaves possíveis precisa ser experimentada para se obter sucesso. Na Tabela 1 é possível verificar o tempo envolvido para vários espaços de chaves[7].

Tabela 1: Tempo médio exigido para busca completa da chave

Tamanho da chave (bits)	Número de chaves alternativas	Tempo necessário para 1 decriptografia/µs	Tempo necessário para 10^6 decriptogafia/µs
32	$2^{32} = 4.3 \times 10^9$	2^{31} µs = 35.8 minutos	2.15 milissegundos
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} µs = 1142 anos	10.01 horas
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} µs = 5.4×10^{24} anos	5.4×10^{18} anos
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} µs = 5.9×10^{36} anos	5.9×10^{30} anos
26 caracteres (permutação)	$26! = 4 \times 10^{26}$	2×10^{26} µs = 6.4×10^{12} anos	6.4×10^6 anos

O tamanho de chave de 56 bits é usado para o algoritmo DES. Já para o algoritmo AES o tamanho mínimo especificado é de 128 bits[7].

1.4.2 Funções de Hash

Uma função *Hash*[7] mapeia uma mensagem de tamanho variável em um resumo de mensagem, cujo tamanho é fixo. Também conhecida como síntese de mensagem ou valor de *hash*, a função de *Hash* é uma função unidirecional, pois a partir do resumo gerado não é possível recuperar a mensagem original. O código de *Hash* é uma função de todos os bits da mensagem e oferece uma capacidade de detecção de erros: uma mudança em qualquer bit ou bits na mensagem resulta em uma mudança no código *Hash*, isso porque esta função usa os princípios de Claude Shannon, difusão e confusão.

Um valor de *hash h* é gerado por uma função H na forma:

$$h = H(M)$$

Equação 2: Função genérica de Hash

onde M é uma mensagem de tamanho variável e H(M) é o valor de *Hash* de comprimento fixo. A finalidade de uma função de *Hash* é produzir uma 'impressão digital' de um arquivo, mensagem ou qualquer outro conjunto de dados. Uma função de *Hash* deve atender aos seguintes requisitos:

1. H pode ser aplicado a um bloco de dados de tamanho qualquer.
2. H produz uma saída de tamanho fixo.
3. H(*x*) possui implementações práticas tanto em software quanto em hardware.
4. As funções são irreversíveis, ou seja, não é possível achar a mensagem original apenas pelo *h.*
5. É computacionalmente inviável achar um $y \neq x$, tal que H(y) = H(x).
6. É computacionalmente inviável encontrar qualquer par (x,y) tal que H(x) = H(y).

1.4.3 MAC

O Código de Autenticação de Mensagem (*Message Authentication Code* – MAC)[7], também conhecido como soma de verificação criptográfica (*checksum*), é uma técnica de autenticação que envolve o uso de uma chave secreta para gerar um bloco de dados de tamanho fixo, este bloco é anexado à mensagem na transmissão em um momento em que a mensagem é conhecida como sendo correta. Na recepção a mensagem é autenticada a partir de um novo MAC gerado e comparado. Para geração do novo código, tanto o receptor quanto o elemento que transmite a mensagem, devem estar de posse da chave secreta.

Com o uso de MAC é possível garantir que:

- A mensagem recebida não foi alterada. Se um atacante alterar a mensagem, mas não alterar o MAC, então o cálculo do MAC pelo receptor será diferente do MAC recebido. Como o atacante não terá a posse da chave secreta, logo ele não poderá alterar o valor do MAC para um valor válido.
- A mensagem recebida é do emissor declarado. Apenas os nós autênticos estarão de posse da chave secreta, nenhum elemento malicioso deverá compartilhar essa chave.
- Se a mensagem inclui um número de sequência, então o receptor poderá ter a certeza que a mensagem está na devida sequência.

O algoritmo de MAC é uma função irreversível, ou seja, não é possível a partir do código gerado obter a mensagem original.

1.4.4 HMAC

Algumas funções de *Hash* não foram projetadas para uso com o MAC e não podem ser usadas diretamente para essa finalidade, pois não dependem de uma chave secreta. Surgiram várias propostas para incorporar uma chave simétrica em algoritmo de *Hash*. A técnica que obteve maior suporte foi *Hash-based MAC* (HMAC)[7].

O HMAC foi publicado na RFC2104 e escolhido como MAC de implementação obrigatória para segurança IP, é usada em outros protocolos da Internet, como SSL. Foram estabelecidos os seguintes objetivos:

- Usar, sem modificações, as funções de *hash* disponíveis, particularmente aquelas que tenham bom desempenho em software, e para as quais o código esteja ampla e gratuitamente disponível.
- Permitir a substituição fácil da função de *hash* embutida caso sejam encontradas ou exigidas funções de *hash* mais rápidas ou mais seguras.
- Preservar o desempenho original da função de *hash* sem incorrer em degradação significativa.
- Usar e gerenciar as chaves de maneira simples.
- Ter uma análise criptográfica bem compreendida quanto à força do mecanismo de autenticação com base em suposições razoáveis sobre a função de *hash* embutida.

O HMAC trata a função *Hash* como uma "*caixa-preta*", ou seja, como um módulo independente onde o esquema HMAC não precisa ter conhecimento de seu funcionamento. Isso o torna adaptável a qualquer nova função de *hash* que possa vir a surgir e facilita a troca de uma função de *hash* por outra, sem necessidade de uma reimplementação.

A força da segurança do HMAC está diretamente associada ao tamanho da chave simétrica utilizada e a segurança do algoritmo de *Hash* utilizado. Uma chave pequena pode comprometer a segurança do esquema, mas uma chave grande não aumentará mais o nível de segurança. A relação de tamanho ideal para a chave é expressa em [16].

1.4.5 CMAC

Muito adotado pelo governo e pela indústria, o CMAC também é conhecido como CBC-MAC, o CMAC na verdade é uma variação do CBC-MAC proposto por John Richard Black Jr e Phillip Rogaway. O CMAC é um esquema baseado no MAC com o uso de algoritmos de cifras de blocos, permitindo uma interface entre os mais diversos algoritmos de cifras, como por exemplo: AES, DES e triplo DES[7]. Pode ser usado para garantir autenticidade e integridade aos dados.

O CMAC é seguro sob um conjunto razoável de critérios, com a seguinte restrição:

- Somente mensagens de tamanho fixo *mn* são processadas:
    - *n* é o tamanho do bloco cifrado.
    - *m* é um inteiro positivo fixo.

Para contornar uns problemas de segurança do CBC-MAC, o CMAC tem um algoritmo que faz uso de três chaves simétricas.

2 Objetivos

No intuito de prover um ambiente seguro para o tráfego das mensagens de controle do *Heterogenous Technologies Routing* (HTR) [10] este trabalho tem como objetivo garantir como pilares de segurança integridade e autenticidade para o ambiente de rede ad hoc HTR. Foi utilizado um esquema de HMAC para alcançar tais pilares. Porém vistos as restrições deste ambiente, o sistema desenvolvido foi estruturado provendo uma interface entre as funções de *Hash* e o esquema de HMAC, sendo possível assim permutar as funções de *Hash* do esquema de HMAC.

Este trabalho de graduação (TG) tem como objetivo um esquema de segurança que ofereça integridade e autenticidade para as mensagens de controle do HTR. Tal esquema deve garantir baixo custo energético e baixo consumo de largura de banda, sem comprometer a segurança do esquema proposto.

Será feita uma análise de impacto entre cinco funções de *Hash* utilizadas no esquema de HMAC, e serão comparadas métricas como: Tempo de Convergência, Sobrecarga das Mensagens e Perda de Pacotes. As cinco funções de *Hash* comparadas entre si serão: MD5, SHA-1, SHA-256, SHA-384 e SHA-512.

2.1 Descrição do Problema

Atualmente a rede HTR parte do pressuposto que todos os nós que compõem a rede são nós confiáveis, ou seja, não há nós que praticam ações maliciosas. Desta forma todas as informações de dados de controle trafegam em texto claro, sem controles de integridade e autenticidade da informação.

Porém tal hipótese não condiz com a realidade. A expansão do número de aplicações em ambiente ad hoc acabou atraindo vários usuários maliciosos, que se aproveitam da grande vulnerabilidade deste tipo de rede para atacar nós autênticos. A ausência de uma estrutura física permite que os canais de comunicações fiquem abertos, possibilitando que um elemento não autêntico da rede possa interceptar o sinal e capturar os dados. Outras questões como ausência de um elemento central coordenador, largura de banda e estabilidade, limitam os mecanismos de criptografias que poderiam ser utilizados para proteger o canal aberto de comunicação entre os nós.

2.2 Arquitetura

O framework HTR possui dois tipos de mensagens de controle, são as mensagens de HELLO e TC. A versão segura S-HTR insere um novo cabeçalho, como ilustrado na Figura 7. Este cabeçalho além de encapsular as mensagens de HELLO e TC, é composto do resumo de uma função *Hash* entre outros campos que serão melhores descritos nas próximas subseções.

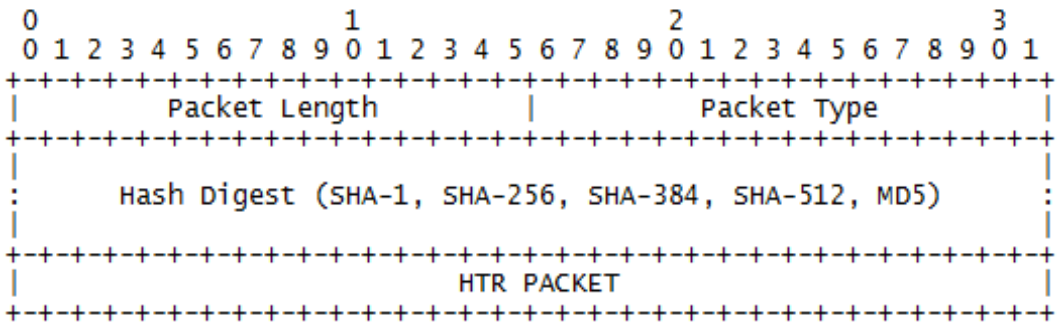

Figura 7: Cabeçalho da mensagem de controle do S-HTR

A nova sobrecarga de dados gerada pelo cabeçalho varia conforme a função de *Hash* utilizada. Os campos **Packet Length** e **Packet Type** possuem 2 bytes cada. *Packet Length* é o campo que armazena o tamanho total da mensagem. *Packet Type* é campo que informa o tipo da mensagem, ou seja, qual função *Hash* foi utilizada para geração do resumo. O tamanho do resumo correspondente a cada função de *Hash* está disposto na Tabela 9

Independente se a mensagem encapsulada é um HELLO ou um TC, os campos dessa mensagem serão usados como entrada para a função de *Hash*. Ou seja, o resumo gerado será composto por todos os campos da mensagem de controle que será encapsulada.

O tamanho do campo **HashDigest** varia conforme a função de *Hash* utilizada que pode ser de 64, 128, 256, 384 e 512 bits. Este resumo é encriptado por um protocolo de criptografia simétrica AES que utiliza uma chave previamente trocada entre os participantes para fornecer o código de autenticação da mensagem baseado em *Hash*. O esquema HMAC para as mensagens do HTR é ilustrado na Figura 8 e a simbologia está explicada na Tabela 2.

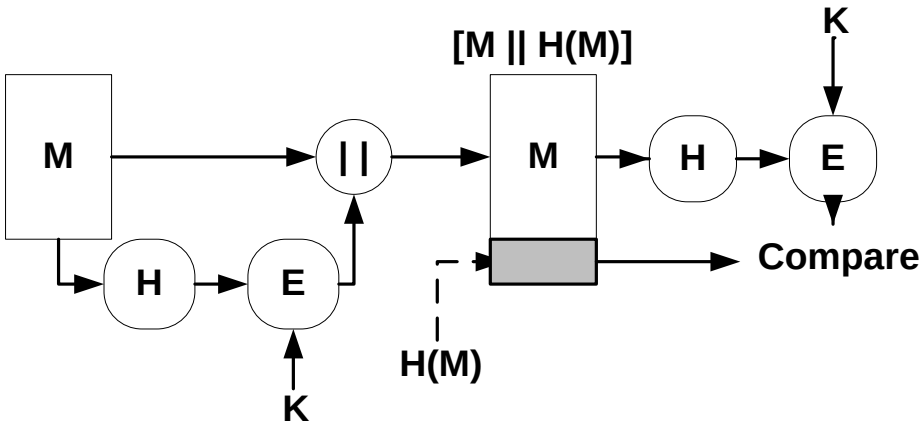

Figura 8: Estrutura do HMAC

Tabela 2: Simbologia do esquema de HMAC

Representação	Significado
M	Mensagem propagada na rede
H	Função de *Hash*
E	Esquema de criptografia simétrica
K	Chave Simétrica
H(M)	Resumo da Mensagem M
||	Elemento Concatenador

No processo de transmissão, a partir de uma mensagem ‘M’ e uma chave simétrica ‘K’ é gerado um resumo da mensagem ‘M’ pela função *Hash* em ‘H’, esta mensagem é então criptografada por pelo esquema de criptografia simétrica ‘E’ usando a chave ‘K’. O resumo é concatenado com a mensagem e então propagado para o devido destino. Na recepção da mensagem, a partir da mensagem recebida ‘M’ e de uma chave simétrica ‘K’, a mesma utilizada na etapa de transmissão, é então gerado um novo resumo da mensagem, criptografado pelo esquema de criptografia simétrico ‘E’ utilizando a chave ‘K’. O

novo resumo é comparado com o resumo recebido na mensagem. Esse processo de comparação é responsável por verificar e garantir a integridade e autenticidade da mensagem.

3 Estado da Arte

As redes ad hoc, devido a sua natureza, são mais suscetíveis a ataques maliciosos que redes estruturadas. Em geral os ataques específicos para essas redes visam sempre a tabela de roteamento dos elementos que compõem a rede. Os principais ataques a redes ad hoc, listados por [17], estão descritos nas próximas subseções.

3.1 Ataques

3.1.1 Flooding Attack

É um tipo de ataque de negação de serviço para toda a rede. Neste ataque, os protocolos de roteamento proativos são mais susceptíveis. Os atacantes sobrecarregam os nós autênticos com a criação de rotas falsas com destino para nós não existentes dentro da rede. Assim, os nós autênticos ficam sobrecarregados sem conseguir criar rotas verdadeiras para elementos autênticos, [17]–[19].

3.1.2 Sleep Depravation

Com o foco no consumo energético de um determinado nó, nesse ataque o atacante faz consecutivas requisições de rota, sejam elas falsas ou verdadeiras, sobrecarregando um elemento da rede. O nó sobrecarregado irá desperdiçar a sua energia respondendo as requisições do nó malicioso e reencaminhando seus pacotes, [17].

3.1.3 Impersonation Attack

É um ataque inerente as redes ad hoc, onde um nó malicioso finge ser um nó legítimo da rede, ou seja, o atacante se passa por um nó autêntico, dessa forma junta-se a rede e envia informações falsas de rotas para dentro da rede, mascarando outros nós autênticos, [17], [20].

3.1.4 Black Hole Attack

Trata-se de um tipo de ataque de negação de serviço, o atacante injeta rota falsa na rede afirmando que tem o menor caminho para um dado destino. Atraindo assim todo o tráfego para ele, dessa forma o atacante pode interceptar os pacotes fazendo mau uso deles ou eliminando-os, [21].

3.1.5 Node Isolation Attack

Ataque criado contra o protocolo *Optimized Link State* Routing (OLSR). O atacante tem como meta isolar um determinado nó dentro da rede, ou seja, o atacante evita que rotas sejam construídas para um determinado nó destino, [22].

3.1.6 Routing Table Poisoning Attack

Os protocolos proativos são dirigidos à tabela, em cada um dos nós da rede é mantida atualizada uma tabela com rotas para outros nós. Neste ataque, o atacante gera e envia tráfego falso ou altera o conteúdo de uma mensagem autêntica, criando entradas falsas nas tabelas de roteamento de outros nós na rede, [17].

3.1.7 Wormhole Attack

O atacante replica os dados que trafegam em um ponto da topologia da rede em outro ponto. Ou seja, os dados são inseridos em um "túnel" que liga dois pontos distintos da rede. Esse ataque envolve a cooperação de dois atacantes com o intuito de distorcer a topologia da rede ad hoc [23].

3.1.8 Location Disclosure Attack

Fazendo análise de tráfego ou com simples monitoramentos, o atacante pode descobrir a localização de um nó dentro da rede, pode descobrir também a estrutura da rede.

3.1.9 Rushing Attacks

Nas redes ad hoc as rotas entre os nós são criadas dinamicamente a partir de pacotes de requisição de rotas, os ROUTE REQUEST. No Rushing Attack, o atacante responde o mais rápido a essas requisições de rota, de forma a garantir que todas as rotas tenham como elemento intermediário este nó malicioso. As requisições que são respondidas com atraso são descartadas pelos nós, [17].

3.1.10 Blackmail Attack

Este ataque tem como alvo protocolos que implementam um sistema seguro baseado em *Blacklist*. O atacante cria uma *blacklist* própria, eliminando os nós maliciosos e/ou inserindo nós autênticos na *blacklist*. Essa nova *blacklist* é então propagada para dentro da rede e são atualizadas em todos os nós da rede.

3.1.11 Snare Attack

Em um ambiente militar, onde um nó foi fisicamente comprometido. Se um indivíduo que compõe a rede foi capturado e está de porte de um dispositivo que faz parte de uma rede ad hoc. O inimigo pode usar o dispositivo, se passando por um indivíduo autêntico da rede, interceptando o tráfego e usando-o para fins próprios.

3.1.12 The Invisible Node Attack

É um tipo de ataque passivo para protocolos que necessitam de algum tipo de identificação para qualquer funcionalidade. O atacante participa das ações sem revelar sua identidade, com o intuito de apenas coletar as a informações que trafegam na rede, [24].

3.2 S-HTR contra os Ataques

Fazendo uma análise teórica dos ataques sobre um ambiente com o S-HTR, foi definido um status para cada ataque descrito na subseção anterior. Os resultados são expressos na Tabela 3.

Tabela 3: Status dos Ataques em um ambiente com S-HTR

Ataque	Status
Flooding Attack	Mitiga
Sleep Depravation	Não se enquadra
Impersonation Attack	Não Mitiga
Black Hole Attack	Mitiga
Node Isolation Attack	Não mitiga
Routing Table Poisoning Attack	Mitiga
Wormhole Attack	Não mitiga
Location Disclosure Attack	Não mitiga
Rushing Attacks	Mitiga

Blackmail Attack	Não se enquadra
Snare Attack	Não mitiga
The Invisible Node Attack	Não mitiga

3.2.1 Ataques Mitigados

Ataques cujos riscos e impactos foram reduzidos com o S-HTR.

3.2.1.1 Flooding Attack

Apenas nós autênticos na rede irão ter posse da chave simétrica. No caso dos elementos que não possuem a chave simétrica, não conseguirão gerar pacotes com o resumo *Hash*.

3.2.1.2 Black Hole Attack

A escolha do melhor caminho no HTR é calculada através do algoritmo de Dijkstra, onde os pesos dos enlaces são ponderados pelo HTRScore.

3.2.1.3 Routing Table Poisoning Attack

O uso da função *Hash* com a chave simétrica protege o conteúdo das mensagens de controle, garantindo assim a integridade dos dados. Logo não é possível um nó não autêntico alterar ou falsificar os dados das mensagens de controle.

3.2.1.4 Rushing Attack

No *framework* HTR, a seleção de melhor rota é atribuída ao cálculo do MPR. O S-HTR mantém esta funcionalidade.

3.2.2 Ataques Não Mitigados

Ataques cujos riscos e impactos não foram reduzidos com o S-HTR.

3.2.2.1 Impersonate Attack

É necessário um esquema de troca periódica de chaves entre os nós autênticos, garantindo que apenas os nós autênticos terão acesso às chaves simétricas e assimétricas.

3.2.2.2 Node Isolation Attack

É um ataque criado especificamente contra o OLSR, logo, como o HTR herda várias características do OLSR ele também é vulnerável a este ataque. O S-HTR não implementa mecanismos para corrigir este tipo de ataque.

3.2.2.3 Wormhole Attack

Não foi encontrada uma solução tangível para resolver este tipo de ataque.

3.2.2.4 Location Disclosure Attack

Para evitar este ataque é preciso que todo o conteúdo seja criptografado. O S-HTR usa o esquema do HMAC que criptografa apenas o resumo *Hash* gerado.

3.2.2.5 Snare Attack

Há uma necessidade de uma autenticação biométrica para evitar este ataque.

3.2.2.6 The Invisible Node Attack

Ataques do tipo passivo, onde o atacante não tem influência alguma sobre a rede, são difíceis de serem detectados.

3.2.3 Ataques que Não se Enquadram a Natureza do HTR

Alguns ataques não são aplicáveis sobre a estrutura do HTR, sendo definidos por não se enquadrar ao framework, não tendo qualquer efeito.

3.2.3.1 Sleep Depravation

Este tipo de ataque tem impacto em protocolos reativos, onde as rotas são montadas no instante em que são requisitados. O S-HTR é um protocolo proativo onde os destinos são armazenados em tabelas, logo não há requisição de rota.

3.2.3.2 Blackmail Attack

Nem HTR e nem o S-HTR fazem uso de mecanismos de segurança de *Blacklist*, logo este ataque não se enquadra.

4 Metodologia

O trabalho foi desenvolvido em ambiente simulado, fazendo uso do simulador ns-3[25] e da biblioteca Crypto++[26].

O *Network Simulator 3* (ns-3) é um simulador de rede movido a eventos discretos para sistemas de internet, é um software aberto voltado para a comunidade acadêmica licenciado pela GNU GPLv2. O ambiente foi alterado para que as funcionalidades do HTR ficassem disponíveis. Ou seja, além das funções que já são inerentes ao ambiente de simulação, foram incluídas novas funcionalidades do protocolo HTR e modelos de energia das duas tecnologias 4G (*4th Generation*): LTE [27] e WiMAX [28].

Crypto++ é uma biblioteca aberta escrita em C++ licenciada pela "*Boost Software License 1.0*" na versão mais recente 5.6.2, lançada em 20/2/2013. Dispondo de vários esquemas de criptografia, como por exemplo: funções de *Hash*, criptografia de chave pública e chave privada, criptografia de curvas elípticas, códigos de autenticação de mensagens (MAC), cifras de blocos e de fluxo, entre outros, dispondo ainda de algoritmos considerados inseguros ou obsoletos. Diferentemente do ns-3, a biblioteca Crypto++ não sofreu qualquer tipo de alteração para ser utilizada. Neste contexto a sua versão mais recente será usada para prover os mecanismos de segurança nas simulações.

4.1 Ambiente

No ambiente simulado foi utilizada uma modulação OFDM [29] com uma taxa de codificação de 3/4, OFDM é um método de codificação de dados digitais para múltiplas frequências que é muito comum para comunicação digital de banda larga. Friis [30] foi o modelo de propagação utilizado para modelar as perdas de dados durante as transmissões com a frequência do canal de 2.4GHz. O modelo padrão para camada física é o padrão PHY 802.11 [31], seguindo todas as normas especificadas pela IEEE.

As informações de ambiente são encontradas na Tabela 4, enquanto que detalhamentos do modelo de propagação Friis utilizado estão na Tabela 5. Na Tabela 6 estão dispostas as informações do modelo de energia utilizado para o chip de rádio da Texas Instrument CC2420.

Os dados referentes a configuração do cenário com disposição de nós serão detalhados mais adiante. Assim como toda a descrição das métricas e dos testes estatísticos aplicados nos dados coletados.

Tabela 4: Configuração de tecnologias

Parâmetros	Wi-Fi
Modelo da Camada Física	PHY 802.11
Frequência do Canal Wireless	2.4 GHz
Modelo de Propagação Perda	Friis
Poder de Transmissão	35-39 dBm
Modulação	OFDM 64 QAM
Taxa de Codificação	3/4

Tabela 5: Configuração do modelo de propagação

Modelo de Propagação	Parâmetros Padrão	Valores
Friis	Tamanho da Onda	58.25mm
	Perda do Sistema	1

	Distância Mínima	0.5m

Tabela 6: Modelo de energia do chip de rádio

Tecnologia	Chip de Rádio	Parâmetros	Valores
Wi-Fi	**Texas Instruments-CC2420**	Voltagem	2.5V
		Corrente de Transmissão	17.4mA
		Corrente de Recepção	19.7mA
		Corrente de IDLE, CCA, BUSY, SWITCHING	0.426mA

4.2 Cenário

Os elementos que compõem a rede foram alinhados conforme a Figura 9, com uma distância de 30 metros entre eles, o protocolo utilizado na camada de transporte foi o UDP com taxa de dados de 300, 600 e 1000kbps. Os nós são estáticos na rede durante todo o processo de simulação.

Os experimentos foram executados 400 vezes para cada um dos cenários com duração máxima de 1800 segundos. Cada cenário possuía uma densidade de nó e um protocolo. A densidade foi variada de 25 nós a 81 nós na rede, enquanto que o protocolo utilizado foi permutado entre HTR, S-HTR(MD5), S-HTR(SHA-1), S-HTR(SHA-128), S-HTR(SHA-256), S-HTR(SHA-384), S-HTR(SHA-512).

Todas as informações referentes ao cenário estão expressas na Tabela 7.

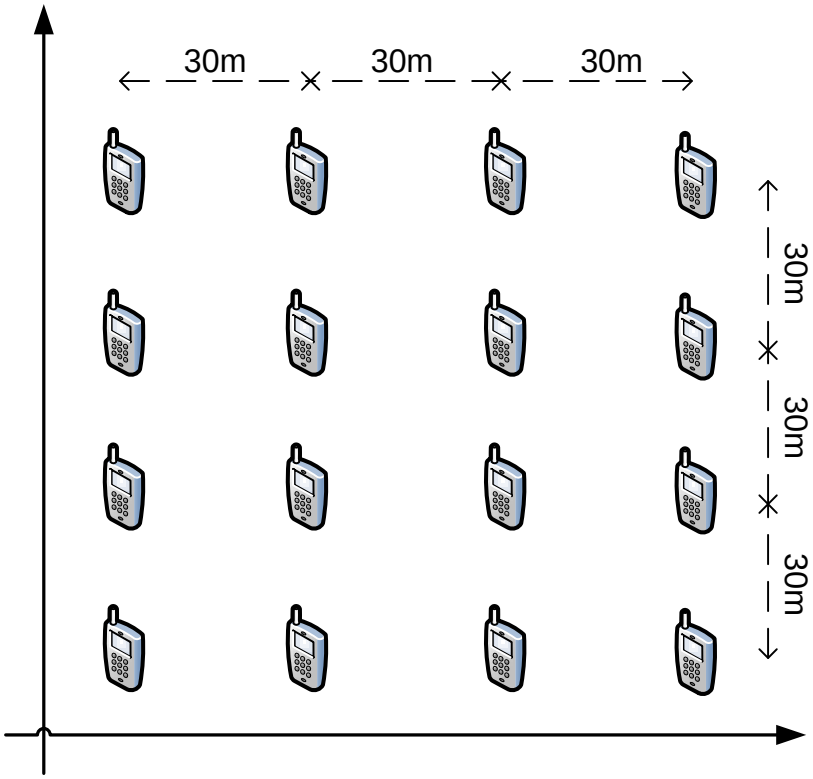

Figura 9: Distribuição dos nós no cenário

Tabela 7: Configuração dos parâmetros de simulação

Parâmetros	Valores
Simulador	ns-3
Protocolos de Roteamento	HTR / S-HTR(MD5) / S-HTR(SHA-1) / S-HTR(SHA-256) / S-HTR(SHA-384) / S-HTR(SHA-512)
Densidade de Nós na Simulação	5x5 / 7x7 / 8x8 / 9x9
Tempo Máximo de Simulação	1800 segundos
Número de Simulações	400
Protocolo de Transporte	UDP
Protocolo de Rede	IPv4
Unidade de Fragmentação IP	2048
Taxa de Dados	300 / 600 / 1000 Kbps

4.3 Métricas

Há um conjunto de métricas focadas em MANET's que avaliam a eficiência de um protocolo de roteamento. Em [2] é descrito como avaliar redes ad hoc móveis, são estabelecidas as principais métricas tanto para uma avaliação quantitativa quanto qualitativa.

Neste trabalho, há o interesse de se comparar os resultados com os já conhecidos resultados do protocolo HTR puro, por isso foi utilizados um subconjunto das mesmas métricas de avaliação do protocolo como descrito em [12], [13].

4.3.1 Tempo de Convergência

A rede converge quando a tabela de roteamento de cada nó possui rotas para todos os outros nós que compõem a rede. Ou seja, o tempo de convergência é o tempo que a rede leva para que a tabela de roteamento de cada elemento esteja devidamente preenchida.

Para cada mensagem há um resumo gerado pela função de *Hash* e associado a este resumo existe um tempo de processamento gasto para geração do mesmo. A biblioteca Crypto++ fornece um *benchmark* que foi executado e seus resultados foram coletados e utilizados na implementação da simulação, veja a Tabela 8.

Frequência da CPU do teste da plataforma é de 3.4GHz.

Tabela 8: Benchmark da biblioteca Crypto++

Algoritmo	MiB/Segundo	Ciclos por Byte
MD5	683	4.7
SHA-1	479	6.8
SHA-3-256	320	10.1
SHA-3-384	249	13.0
SHA-3-512	173	18.7

O benchmark completo pode ser visto no Apêndice A.

4.3.2 Sobrecarga das Mensagens

Apesar de não ter sido inserido tráfego na simulação da rede, os pacotes de controle geram um sobrecarga na rede. A sobrecarga das mensagens é definida neste trabalho como o fluxo médio de dados de controle ao longo da simulação.

Como discutido na seção Arquitetura, a sobrecarga do novo cabeçalho depende, principalmente, da função de *Hash* utilizada. Na Tabela 9 é descrito o tamanho de cada resumo para cada algoritmo de *Hash* utilizado no esquema de HMAC deste projeto. É importante notar que o *Hash* é apenas um dos campos que gera a sobrecarga, além dele há os campos de *Packet Lengh* e *Packet Type*. Porém o maior impacto da sobrecarga é atribuído ao campo resumo *Hash*. Uma lista com vários algoritmos *Hash* encontra-se no Apêndice A.

Tabela 9: Tamanho dos resumos gerados pelas respectivas funções Hash

Algoritmo	Tamanho da Saída (bits)
MD5	128
SHA-1	160
SHA-3-256	256
SHA-3-384	384
SHA-3-512	512

4.3.3 Perda de Pacotes

É possível verificar dentro da rede simulada a quantidade de pacotes transmitidos e a quantidade de pacotes recebidos. É sabido que nem todos os pacotes que são transmitidos dentro da rede são recebidos por algum nó da mesma. Alguns desses pacotes podem ser perdidos e/ou descartados ao longo do trajeto. Com base nos dados de transmissão e recepção de pacotes é possível verificar a proporção de perda de pacotes durante a simulação.

4.4 Testes de Hipótese

Dado um conjunto de dados amostrais é possível com o uso de ferramentas estatísticas fazer inferências sobre a população, essas ferramentas são conhecidas como Testes de Hipótese.

Uma hipótese é uma suposição feita sobre os parâmetros de uma distribuição de probabilidade. Uma dada hipótese deve ser testada, podendo ser aceita ou não. Caso ela seja aceita, provavelmente aquele conjunto de dados amostrais segue a dada distribuição formulada na hipótese, as técnicas utilizadas tentam minimizar ao máximo que uma hipótese errada seja aceita ou que uma hipótese verdadeira seja negada.

Consideramos H_0 como a hipótese nula e H_1 como a hipótese alternativa a ser testada. O teste conduz a aceitação ou rejeição de H_0.

Neste trabalho serão utilizados dois tipos de testes de hipótese para os dados amostrais, o objetivo é encontrar os dados que seguem uma distribuição normal (gaussiana). Para as métricas de Perda de Pacotes e Sobrecarga das Mensagens foi utilizado um teste de hipótese paramétrico, porém a métrica de Tempo de Convergência teve sua hipótese negada neste tipo de teste, sendo necessário aplicar um teste de hipótese não paramétrico. O teste Wilcoxon Signed-Rank[32], [33] foi o teste de hipótese não paramétrico utilizado para a métrica de Tempo de Convergência, enquanto que o teste de Shapiro-Wilk[32], [34] foi o teste de hipótese paramétrico utilizado para as demais métricas.

4.4.1 Teste de Shapiro-Wilk

Proposto em 1965 por Samuel Shapiro e Martin Wilk, o teste de Shapiro-Wilk[32] testa a hipótese nula que uma dada amostra $x_1, x_2,...,x_n$ obtida de uma população de distribuição normal. Esse teste é baseado na estatística W dada por:

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}^{n}(x_{(i)} - \bar{x})^2}$$

Equação 3: Estatística W para o teste de normalidade de Shapiro-Wilk

Onde:

- $x_{(i)}$ é o i-ésimo valor da amostra;
- $\bar{x}$ é a média das amostras;
- a_i são constanstes geradas pelas médias, variâncias e covariâncias das estatísticas de ordem de uma amostra de tamanho n de uma distribuição Normal. Seus valores tabelados são encontrados no Apêndice A.

A hipótese H_0 será rejeitada se o valor da estatística W for muito baixo, em outras palavras, formalmente definido pela desigualdade $W_{calculado} < W_{\alpha}$. Os valores críticos da estatística W estão tabelados no Apêndice A. O valor do nível de significância α é normalmente 0.05[34].

4.4.2 Teste de Wilcoxon Signed-Rank

Utilizamos o teste de Wilcoxon Signed-Rank[32], [33] para comparar se as medidas de posição de duas amostras são iguais. Em duas populações onde não temos informações quanto as suas distribuições, consideramos duas amostras de tamanho n vindas de duas populações P_1 e P_2, isto é, $x_1,...,x_n$ e $y_1,...y_n$. Vamos considerar observações pareadas, podemos assumir que temos uma amostra de pares:

$$\{(X_1, Y_1), \dots, (X_n, Y_n)\}$$

Equação 4: Duas amostras independentes pareadas

Definindo $\Delta_i = x_i - y_i$, para i = 1, 2,..., n. Assim obtemos as amostras $\Delta_1, \Delta_2,..., \Delta_n$, resultantes das diferenças de valores entre cada par.

Estabelecemos as hipóteses para realizar o Teste de Wilcoxon:

$$\begin{cases} H_0 : \Delta = 0 \\ H_1 : \Delta \neq 0 \end{cases} \qquad \begin{cases} H_0 : \Delta = 0 \\ H_1 : \Delta > 0 \end{cases} \qquad \begin{cases} H_0 : \Delta = 0 \\ H_1 : \Delta < 0 \end{cases}$$

Equação 5: Hipóteses para o Teste de Wilcoxon

Se as populações diferir em localização ou não, será testado da seguinte forma: se a hipótese nula for aceita, teremos que a mediana da diferença é nula, ou seja, as populações não diferem em localização. Já, se a hipótese nula for rejeitada, temos que as populações diferem em localização.

5 Resultados

Os dados coletados estão dispostos em gráficos, para facilitar o entendimento de leitor. Os dados dos protocolos estão quantificados em percentual em relação ao protocolo base, o HTR sem criptografia. Ou seja, o valor coletado para o protocolo HTR é definido como base, sendo atribuído 100% a este valor, os valores coletados para o protocolo S-HTR, para cada função de *Hash*, é definido em relação ao protocolo base, o HTR. Os resultados estatísticos estão tabelados no Apêndice C.

A partir das métricas definidas na seção de Metodologia, foram coletados os seguintes resultados.

5.1 Tempo de Convergência

Para o cenário de menor densidade, 25 nós, a Figura 10 nos mostra que os tempos de convergência não diferiram entre o HTR e os S-HTR. Existiu uma variação menor que 2 pontos percentuais.

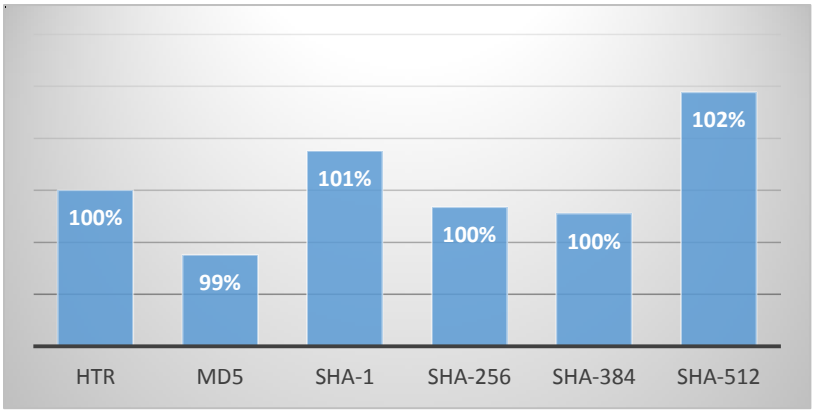

Figura 10: Comparação entre os tempos de convergência para o cenário de 25 elementos

Quando a densidade da rede cresce, este impacto torna-se notório. Para o cenário com uma densidade de 49 nós, o impacto do S-HTR em relação HTR chegar a 6 pontos percentuais em média, como ilustrado na Figura 11.

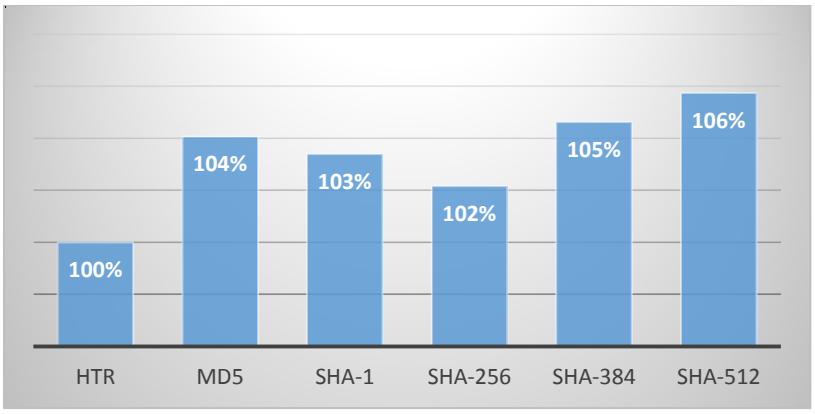

Figura 11: Comparação entre os tempos de convergência para o cenário de 49 elementos

Para o cenário de 64 nós (Figura 12) e cenário de 81 nós (Figura 13), o tempo de convergência dos protocolos com o mecanismo de segurança são visivelmente maiores que o do protocolo base. Conforme a densidade da rede for aumentando, este impacto tende, em média, a ficar cada vez maior.

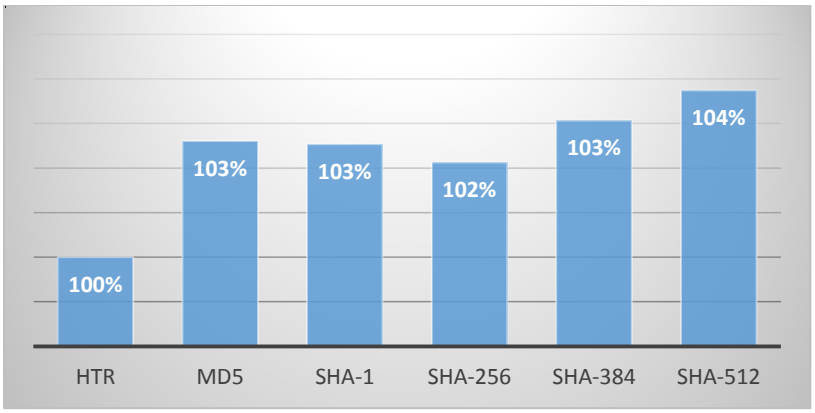

Figura 12: Comparação entre os tempos de convergência para o cenário de 64 elementos

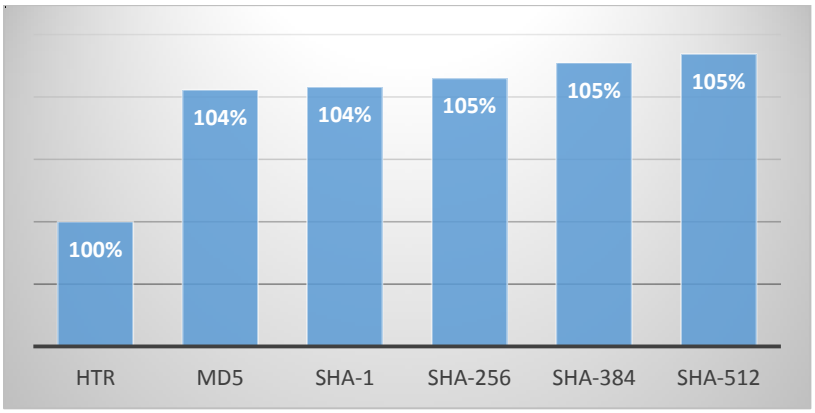

Figura 13: Comparação entre os tempos de convergência para o cenário de 81 elementos

O cenário de maior densidade, com 81 nós (Figura 13), foi o que obteve o maior impacto no tempo de convergência, como era esperado. Porém, apesar da maior densidade ter o maior impacto, ainda assim atingiu no máximo um aumento de apenas 5 pontos percentuais.

5.2 Sobrecarga das Mensagens

Já no cenário com 25 nós na rede é possível observar na Figura 14 que o impacto na rede é quase dobrado, quando comparado o protocolo com criptografia em relação ao HTR.

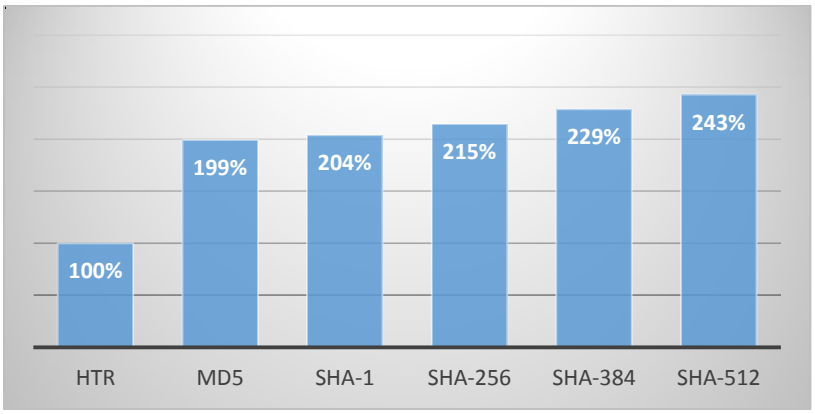

Figura 14: Comparação entre as cargas de dados de controle para um cenário de 25 elementos

Para os cenários maiores (Figura 15, Figura 16 e Figura 17) esta proporção no impacto se mantém. Ou seja, cada função *Hash* utilizada no protocolo dobra, em média, a quantidade de dados no cabeçalho da mensagem, em relação ao HTR.

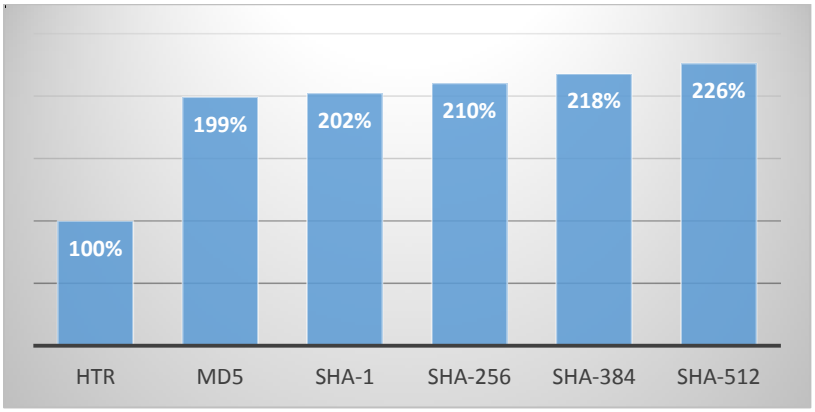

Figura 15: Comparação entre as cargas de dados de controle para um cenário de 49 elementos

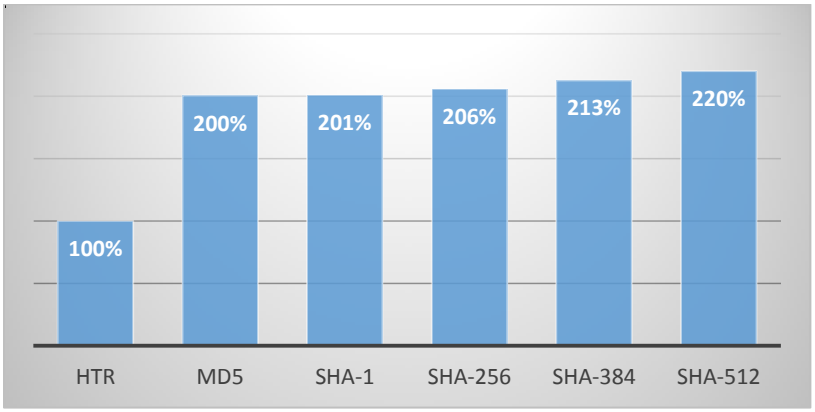

Figura 16: Comparação entre as cargas de dados de controle para um cenário de 64 elementos

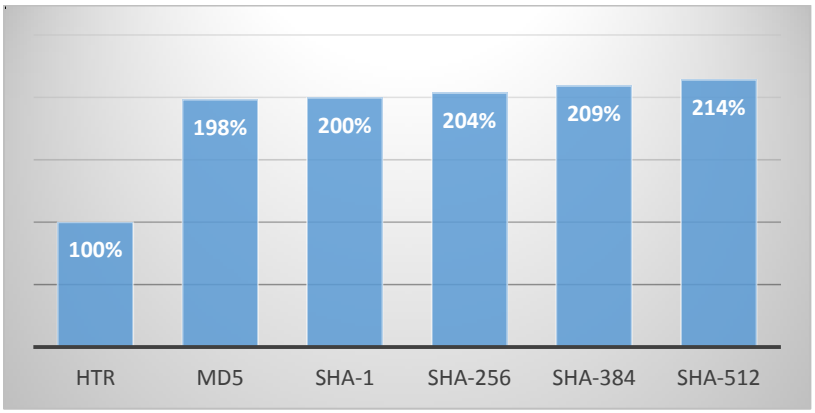

Figura 17: Comparação entre as cargas de dados de controle para um cenário de 81 elementos

Apesar de a densidade crescer, o impacto na sobrecarga da mensagem mantém uma proporção entre os protocolos.

5.3 Perda de Pacotes

Em todos os casos os protocolos seguros apresentaram um aumento na taxa de perda de pacotes. Quanto maior a densidade da rede maior é a proporção.

Em um cenário de menor densidade, 25 nós, o impacto dos protocolos seguros sobre a perda de pacotes quando comparados com o HTR, chegam até cerca de 50 pontos percentuais de pacotes perdidos, como visto na Figura 18 com a função *Hash* SHA-512.

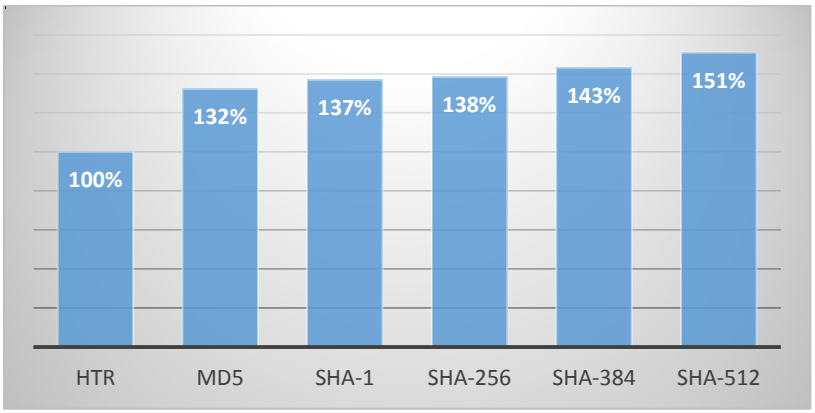

Figura 18: Comparação entre a perda de pacotes de controle para um cenário de 25 elementos

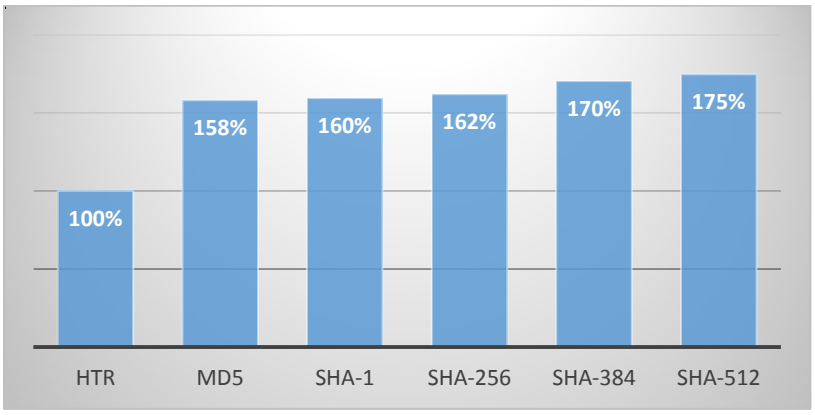

Figura 19: Comparação entre a perda de pacotes de controle para um cenário de 49 elementos

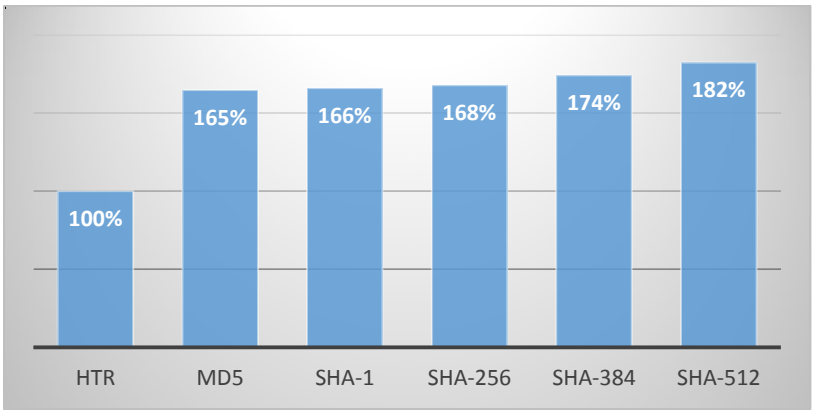

Figura 20: Comparação entre a perda de pacotes de controle para um cenário de 64 elementos

Conforme a densidade da rede aumenta o impacto dos protocolos seguros sobre a perda de pacotes aumenta também, porém conforme observamos nas figuras anteriores, o impacto causado pelos protocolos seguros, quando comparados entre si, tendem a manter uma média similar.

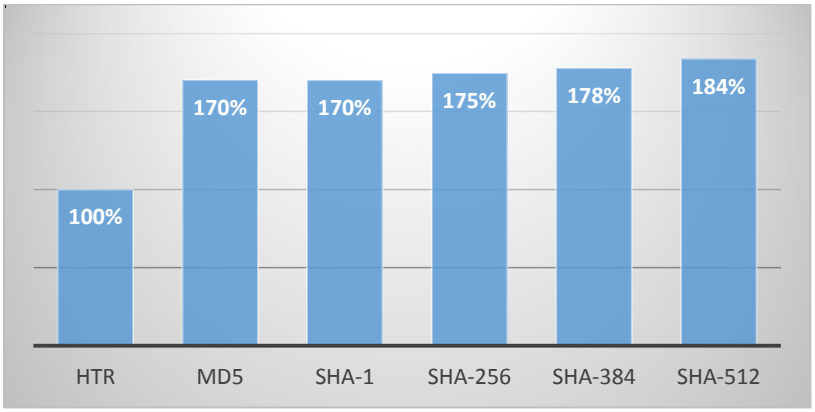

Figura 21: Comparação entre a perda de pacotes de controle para um cenário de 81 elementos

Para redes com uma alta densidade de nós, a perda de pacotes chega a quase dobrar para a função de SHA-512, como é possível ver na Figura 21.

5.4 Discussão

Conforme observado na Tabela 8, há um custo de tempo associado à geração de um resumo para cada função de *Hash*, que influi diretamente no tempo de convergência da rede. Porém para o ambiente HTR, os dados que são a entrada para as funções de *Hash* são os dados das mensagens de controle, a carga desses dados é muito pequena, na ordem de dezenas de bytes. Sendo assim o tempo de convergência dos protocolos com seus respectivos esquemas de *Hash* não diferiu tanto do protocolo HTR sem criptografia. Mas conforme observamos, à medida que a densidade da rede cresce, o impacto tende a ser cada vez maior, em média. Pelo teste de hipótese de Wilcoxon foi provado que para os cenários maiores (49 nós, 64 nós e 81 nós) a diferença entre as medianas, entre os protocolos seguros e o HTR, era muito provável de ser diferente de zero, ou seja, provando que essa ordem de impacto se manterá. Porém para o menor cenário (25 nós) foi verificado que a diferença entre as medianas provavelmente daria zero, confirmado a hipótese de que para um cenário muito pequeno o impacto será irrelevante. Ou seja, o impacto causado pelo protocolo crescerá conforme cresce a densidade da rede, como era esperado.

O novo cabeçalho que é anexado as mensagens de controle, definido na seção Arquitetura, insere uma sobrecarga referente ao tamanho do resumo gerado pela função de *Hash*, mostrado na Tabela 9. Para todos os protocolos seguros, a sobrecarga da rede é maior que o dobro em relação ao protocolo HTR sem criptografia. Em alguns casos o impacto, em relação ao protocolo base, chegou a ser de 243 pontos percentuais. Entre as métricas estudadas, esta foi a que apresentou maior impacto sobre a estrutura da rede. Mas quando comparamos o custo entre os protocolos seguros, o custo de banda é muito próximo com pouca variação de pontos percentuais. Porém dobra a sobrecarga quando comparado com o protocolo HTR sem criptografia. Isto é devido aos resumos inseridos que, conforme a Tabela 9, entre eles há apenas uma pequena diferença de tamanho.

Quando analisamos as perdas de pacotes causadas pelos protocolos seguros, observamos que para redes com uma alta densidade de nós, a perda de pacotes chega a quase dobrar, como para a função de SHA-512 (Figura 21). Porém quando comparamos o impacto entre os protocolos seguros, observamos uma linearidade entre eles. Ou seja, o impacto sobre a métrica de perda de pacotes não tem uma grande variação de impacto quando comparamos os protocolos seguros entre si.

6 Conclusão

Prover segurança é um requisito imprescindível para qualquer sistema em rede nos dias de hoje. Para uma rede ad hoc esse requisito tem mais importância devido à alta vulnerabilidade inerente a sua arquitetura. Porém prover mecanismos seguro também significa aumentar o custo, seja esse custo com relação a processamento, a banda passante, ou a memória. Faz-se necessário um mecanismo com menor custo/impacto sobre a rede. Neste trabalho foi feita uma análise do impacto de um mecanismo de segurança sobre a rede ad hoc HTR.

Com os resultados coletados é notório que as funções *Hash* que tiveram o maior impacto sobre a rede foram as de SHA-384 e SHA-512, para o tempo de validade das mensagens de controle do HTR, essas funções irão sobrecarregar a rede de forma desnecessária. As funções restantes (MD5, SHA-1 e SHA-256) mantém um mesmo patamar de impacto sobre a rede, em todas as métricas. Porém em 2004, foi anunciada que a função de *Hash* MD5 não possuía resistência a colisão[35], não é possível ignorar essa fraqueza.

A partir de testes estatísticos (Teste de Shapiro-Wilk e Teste de Wilcoxon Signed-Rank), conclui-se que a função de SHA-1 terá sempre um impacto menor ou no máximo igual ao da função de SHA-256 para todas as métricas. Para a métrica de Tempo de Convergência o teste de Wilcoxon demostrou que para cenários pequenos a diferença entre as medianas é provável que seja zero, ou seja, o impacto causado pelas duas é em média igual. Já para cenários maiores o teste de Wilcoxon nos mostrou que provavelmente em média o impacto causado pelo SHA-256 será sempre maior que o do SHA-1. As outras métricas (Sobrecarga das Mensagens e Perda de Pacotes), que apresentaram uma distribuição Gaussiana, mostraram que o impacto causado pelo SHA-256 é em média um pouco maior, sendo notório na sobrecarga das mensagens.

Desta forma, os resultados dos testes e a arquitetura do protocolo HTR nos leva a concluir que o esquema de HMAC com uso da função de *Hash* SHA-1 trará menor impacto e ainda garante de forma eficiente a integridade e a autenticidade das mensagens de controle do *framework* HTR.

Este trabalho traz como contribuição principal a escolha do melhor esquema de segurança, balanceando a questão de garantia de segurança e de menor impacto sobre a rede. Outra contribuição é sobre o impacto causado pelas funções: MD5, SHA-256, SHA-384 e SHA-512. O sistema implementado em ambiente simulado ns-3 com uso da Crypto++ também é uma contribuição secundária.

7 Trabalhos Futuros

Neste trabalho, partimos do pressuposto que os nós que compõem a rede HTR já estão na rede de posse da chave simétrica. Como proposta de trabalho futuro poderá ser implementado um esquema seguro de distribuição de chaves simétricas em ambiente inseguro.

Outra proposta de trabalho seria a extensão do protocolo para além de suportar o HMAC também suportar *Cipher-based MAC* (CMAC). O esquema de HMAC, utilizado no trabalho, é uma extensão do esquema de *Message Autentication Code* (MAC), mas além do HMAC há uma proposta de esquemas baseado em MAC que fornecem autenticidade e integridade, CMAC.

Uma terceira proposta seria a de criação de um modelo de energia no ns-3 para cada uma das funções de *Hash*. Dessa forma, poderia ser feita uma análise de consumo energético em cada dispositivo pelo uso de criptografia.

8 Apêndice A

Nesta seção são expressas as tabelas completas referentes a informações de criptografia. A **Erro! Fonte e referência não encontrada.** mostra o *benchmark,* disponível pela própria biblioteca Crypto++, executado em uma plataforma de teste com frequência de CPU igual a 3.4GHz. Já a **Erro! Fonte de eferência não encontrada.**, disponibiliza informações mais completas[36] quanto as mais diversas funções *Hash*.

Tabela 10: Benchmark completo da biblioteca Crypto++

Algoritimo	MiB/Segundo	Ciclos por Byte	Microsegundos para Configurar Chave e VI	Ciclos para Configurar Chave e VI
AES/GCM	1078	3.0	0.415	1410
AES/CCM	551	5.9	0.264	898
AES/EAX	545	6.0	0.342	1164
GMAC(AES)	1766	1.8	0.416	1413
VMAC(AES)-64	9638	0.3	0.523	1777
VMAC(AES)-128	5461	0.6	0.599	2038
HMAC(SHA-1)	479	6.8	0.224	762
Two-Track-MAC	284	11.4	0.010	34
CMAC(AES)	683	4.7	0.126	430
DMAC(AES)	683	4.7	0.365	1239
CRC32	507	6.4		
Adler32	1721	1.9		
MD5	683	4.7		
SHA-1	479	6.8		
SHA-256	278	11.7		
SHA-512	371	8.7		
SHA-3-224	341	9.5		
SHA-3-256	320	10.1		
SHA-3-384	249	13.0		
SHA-3-512	173	18.7		
Tiger	595	5.4		
Whirlpool	164	19.8		
RIPEMD-160	291	11.1		
RIPEMD-320	320	10.1		
RIPEMD-128	434	7.5		
RIPEMD-256	492	6.6		
Panama-LE	2028	1.6	0.620	2108
Panama-BE	1067	3.0	0.920	3126
Salsa20	1067	3.0	0.125	425
Salsa20/12	1679	1.9	0.147	500
Salsa20/8	2354	1.4	0.147	498
Sosemanuk	1600	2.0	0.370	1257
MARC4	457	7.1	1.175	3996
SEAL-3.0-LE	833	3.9	22.195	75462

WAKE-OFB-LE	388	8.4	1.340	4555
AES/CTR (128-bit key)	2844	1.1	0.215	730
AES/CTR (192-bit key)	2381	1.4	0.206	702
AES/CTR (256-bit key)	2069	1.6	0.223	757
AES/CBC (128-bit key)	683	4.7	0.166	564
AES/CBC (192-bit key)	575	5.6	0.157	535
AES/CBC (256-bit key)	497	6.5	0.173	590
AES/OFB (128-bit key)	648	5.0	0.215	730
AES/CFB (128-bit key)	674	4.8	0.224	762
AES/ECB (128-bit key)	3385	1.0	0.070	237
Carmellia/CTR (128bit key)	158	20.5	0.200	679
Carmellia/CTR (256bit key)	123	26.3	0.213	725
Twofish/CTR	194	16.7	1.889	6422
Serpent/CTR	97	33.4	0.415	1409
CAST-256/CTR	116	27.9	0.908	3086
RC6/CTR	221	14.7	1.692	5754
MARS/CTR	186	17.5	1.150	3911
SHACAL-2/CTR (128-bit key)	183	17.7	0.249	847
SHACAL-2/CTR (512-bit key)	183	17.7	0.259	881
DES/CTR	83	39.0	2.128	7234
DES-XEX3/CTR	77	42.1	2.114	7187
DES-EDE3/CTR	32	101.3	6.926	23548
IDEA/CTR	98	33.2	0.224	762
RC5 (r=16)	235	13.8	1.343	4567
Blowfish/CTR	141	23.1	26.921	91533
TEA/CTR	79	41.0	0.182	620
XTEA/CTR	76	42.6	0.183	622
CAST-128/CTR	116	27.9	0.279	948
SKIPJACK/CTR	47	68.9	1.519	5164
SEED/CTR (1/2 K table)	69	47.1	0.246	837

Tabela 11: Funções de Hash e suas informações

Algoritmos	Tamanho do Resumo	Tamanho do Bloco	Tamanho da Palavra (word)	Número de Etapas
GOST	256	256	32	256
HAVAL	256/224/192/160	1024	32	160/128/96
MD2	128	128	32	864
MD4	128	512	32	48
MD5	128	512	32	64
PANAMA	256	256	32	-
RIPEMD	128	512	32	48
RIPEMD-128/256	128/256	512	32	64
RIPEMD-160	160	512	32	80
RIPEMD-320	320	512	32	80
SHA-0	160	512	32	80

SHA-1	160	512	32	80
SHA-224, SHA-256	256/224	512	32	64
SHA-384, SHA-512, SHA-512/224, SHA-512/226	512/384	1024	64	80
SHA-3	224/256/384/512	1600	64	24
SHA-3-224	224	1152	64	24
SHA-3-256	256	1088	64	24
SHA-3-384	384	832	64	24
SHA-3-512	512	576	64	24
Tiger(2)-192/160/128	192/160/128	512	64	24
WHIRLPOOL	512	512	8	10

9 Apêndice B

Nesta seção estão dispostos dados estatísticos usados na seção de Testes de Hipótese. Na Tabela 12 estão expressos os valores das constantes a_i para uma amostra de tamanho n [34], usado na Equação 3 para o Teste de Shapiro-Wilk. Na Tabela 13 estão dispostos os valores críticos para cada valor de significância α [34] usado no Teste de Shapiro-Wilk.

Tabela 12: Valor da constante a de uma amostra de tamanho n, Teste de Shapiro-Wilk

i\n	2	3	4	5	6	7	8	9	10	11	12	13	
1	0,7071	0,7071	0,6872	0,6646	0,6431	0,6233	0,6062	0,5888	0,5739	0,5601	0,5475	0,5359	
2			0,1677	0,2413	0,2806	0,3031	0,3164	0,3244	0,3291	0,3315	0,3325	0,3325	
3					0,0875	0,1401	0,1743	0,1976	0,2141	0,2260	0,2347	0,2412	
4							0,0561	0,0947	0,1224	0,1429	0,1586	0,1707	
5									0,0399	0,0695	0,0922	0,1099	
6											0,0803	0,0539	
i\n	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	
1	0,5251	0,5150	0,5056	0,4968	0,4886	0,4808	0,4734	0,4643	0,4590	0,4542	0,4493	0,4450	
2	0,3318	0,3306	0,3290	0,3273	0,3253	0,3232	0,3211	0,3185	0,3156	0,3126	0,3098	0,3069	
3	0,2460	0,2495	0,2521	0,2540	0,2553	0,2561	0,2565	0,2578	0,2571	0,2563	0,2554	0,2543	
4	0,1802	0,1878	0,1939	0,1988	0,2027	0,2059	0,2085	0,2119	0,2131	0,2139	0,2145	0,2148	
5	0,1240	0,1353	0,1447	0,1524	0,1587	0,1641	0,1686	0,1736	0,1764	0,1787	0,1807	0,1822	
6	0,0727	0,0880	0,1005	0,1109	0,1197	0,1271	0,1334	0,1399	0,1443	0,1480	0,1512	0,1539	
7	0,0240	0,0433	0,0593	0,0725	0,0837	0,0932	0,1013	0,1092	0,115	0,1201	0,1245	0,1283	
8			0,0196	0,0359	0,0496	0,0612	0,0711	0,0804	0,0878	0,0941	0,0997	0,1046	
9					0,0163	0,0303	0,0422	0,0530	0,0618	0,0696	0,0764	0,0823	
10							0,0140	0,0263	0,0368	0,0459	0,0539	0,061	
11									0,0122	0,0228	0,0321	0,0403	
12											0,0107	0,0200	
13												0,0000	
i\n	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	

1	0,4407	0,4366	0,4328	0,4291	0,4254	0,4220	0,4188	0,4156	0,4127	0,4096	0,4068	0,4040	
2	0,3043	0,3018	0,2992	0,2968	0,2944	0,2921	0,2898	0,2876	0,2854	0,2834	0,2813	0,2794	
3	0,2533	0,2522	0,2510	0,2499	0,2487	0,2475	0,2463	0,2451	0,2439	0,2427	0,2415	0,2403	
4	0,2151	0,2152	0,2151	0,2150	0,2148	0,2145	0,2141	0,2137	0,2132	0,1227	0,2121	0,2116	
5	0,1836	0,1848	0,1857	0,1864	0,1870	0,1874	0,1878	0,1880	0,1882	0,1883	0,1883	0,1883	
6	0,1563	0,1584	0,1601	0,1616	0,1630	0,1641	0,1651	0,1660	0,1667	0,1673	0,1678	0,1683	
7	0,1316	0,1346	0,1372	0,1395	0,1415	0,1433	0,1449	0,1463	0,1475	0,1487	0,1496	0,1505	
8	0,1089	0,1128	0,1162	0,1192	0,1219	0,1243	0,1265	0,1284	0,1301	0,1317	0,1331	0,1344	
9	0,0876	0,0923	0,0965	0,1002	0,1036	0,1066	0,1093	0,1118	0,1140	0,1160	0,1179	0,1196	
10	0,0672	0,0728	0,0778	0,0822	0,0862	0,0899	0,0931	0,0961	0,0988	0,1013	0,1036	0,1056	
11	0,0476	0,0540	0,0598	0,065	0,0697	0,0739	0,0777	0,0812	0,0844	0,0873	0,0900	0,0924	
12	0,0284	0,0358	0,0424	0,0483	0,0537	0,0585	0,0629	0,0669	0,0706	0,0739	0,0770	0,0798	
13	0,0094	0,0178	0,0253	0,032	0,0381	0,0435	0,0485	0,0530	0,0572	0,0610	0,0645	0,0677	
14		0,0000	0,0084	0,0159	0,0227	0,0289	0,0344	0,0395	0,0441	0,0484	0,0523	0,0559	
15				0	0,0076	0,0144	0,0206	0,0262	0,0314	0,0361	0,0404	0,0444	
16						0,0000	0,0068	0,0131	0,0187	0,0239	0,0287	0,0331	
17								0,0000	0,0062	0,0119	0,0172	0,0220	
18										0,0000	0,0057	0,0110	
19												0,0000	

i\n	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**
1	0,4015	0,3989	0,3964	0,3940	0,3917	0,3894	0,3872	0,3850	0,3830	0,3808	0,3789	0,3770	0,3751
2	0,2774	0,2755	0,2737	0,2719	0,2701	0,2684	0,2667	0,2651	0,2635	0,2620	0,2604	0,2589	0,2574
3	0,2391	0,2380	0,2368	0,2357	0,2345	0,2334	0,2323	0,2313	0,2302	0,2291	0,2281	0,2271	0,2260
4	0,2110	0,2104	0,2098	0,2091	0,2085	0,2078	0,2072	0,2065	0,2058	0,2052	0,2045	0,2038	0,2032
5	0,1881	0,1880	0,1878	0,1876	0,1874	0,1871	0,1868	0,1865	0,1862	0,1859	0,1855	0,1851	0,1847
6	0,1686	0,1689	0,1691	0,1693	0,1694	0,1695	0,1695	0,1695	0,1695	0,1695	0,1693	0,1692	0,1691

7	0,1513	0,1520	0,1526	0,1531	0,1535	0,1539	0,1542	0,1545	0,1548	0,1550	0,1551	0,1553	0,1554
8	0,1356	0,1366	0,1376	0,1384	0,1392	0,1398	0,1405	0,1410	0,1415	0,1420	0,1423	0,1427	0,1430
9	0,1211	0,1225	0,1237	0,1249	0,1259	0,1269	0,1278	0,1286	0,1293	0,1300	0,1306	0,1312	0,1317
10	0,1075	0,1092	0,1108	0,1123	0,1136	0,1149	0,1160	0,1170	0,1180	0,1189	0,1197	0,1205	0,1212
11	0,0947	0,0967	0,0986	0,1004	0,1020	0,1035	0,1049	0,1062	0,1073	0,1085	0,1095	0,1105	0,1113
12	0,0824	0,0848	0,0870	0,0891	0,0909	0,0927	0,0943	0,0959	0,0972	0,0986	0,0998	0,1010	0,1020
13	0,0706	0,0733	0,0759	0,0782	0,0804	0,0824	0,0842	0,0860	0,0876	0,0892	0,0906	0,0919	0,0932
14	0,0592	0,0622	0,0651	0,0677	0,0701	0,0724	0,0745	0,0765	0,0783	0,0801	0,0817	0,0832	0,0846
15	0,0481	0,0515	0,0546	0,0575	0,0602	0,0628	0,0651	0,0673	0,0694	0,0713	0,0731	0,0748	0,0764
16	0,0372	0,0409	0,0444	0,0476	0,0506	0,0534	0,0560	0,0584	0,0607	0,0628	0,0648	0,0667	0,0685
17	0,0264	0,0305	0,0343	0,0379	0,0411	0,0442	0,0471	0,0497	0,0522	0,0546	0,0568	0,0588	0,0608
18	0,0158	0,0203	0,0244	0,0283	0,0318	0,0352	0,0383	0,0412	0,0439	0,0465	0,0489	0,0511	0,0532
19	0,0053	0,0101	0,0146	0,0188	0,0227	0,0263	0,0296	0,0328	0,0357	0,0385	0,0411	0,0436	0,0459
20		0,0000	0,0049	0,0094	0,0136	0,0175	0,0211	0,0245	0,0277	0,0307	0,0335	0,0361	0,0386
21				0,0000	0,0045	0,0087	0,0126	0,0163	0,0197	0,0229	0,0259	0,0288	0,0314
22						0,0000	0,0042	0,0081	0,0118	0,0153	0,0185	0,0215	0,0244
23								0,0000	0,0039	0,0076	0,0111	0,0143	0,0174
24										0,0000	0,0037	0,0071	0,0104
25												0,0000	0,0350

Tabela 13: Valores críticos da estatística W de Shapiro-Wilk

	Nível de significância								
N	**0,01**	**0,02**	**0,05**	**0,1**	**0,5**	**0,9**	**0,95**	**0,98**	**0,99**
3	0,753	0,756	0,767	0,789	0,959	0,998	0,999	1,000	1,000
4	0,687	0,707	0,748	0,792	0,935	0,987	0,992	0,996	0,997
5	0,686	0,715	0,762	0,806	0,927	0,979	0,986	0,991	0,993
6	0,713	0,743	0,788	0,826	0,927	0,974	0,981	0,986	0,989
7	0,730	0,760	0,803	0,838	0,928	0,972	0,979	0,985	0,988
8	0,749	0,778	0,818	0,851	0,932	0,972	0,978	0,984	0,987
9	0,764	0,791	0,829	0,859	0,935	0,972	0,978	0,984	0,986
10	0,781	0,806	0,842	0,869	0,938	0,972	0,978	0,983	0,986

11	0,792	0,817	0,850	0,876	0,940	0,973	0,979	0,984	0,986
12	0,805	0,828	0,859	0,883	0,943	0,973	0,979	0,984	0,986
13	0,814	0,837	0,866	0,889	0,945	0,974	0,979	0,984	0,986
14	0,825	0,846	0,874	0,895	0,947	0,975	0,980	0,984	0,986
15	0,835	0,855	0,881	0,901	0,950	0,975	0,980	0,984	0,987
16	0,844	0,863	0,887	0,906	0,952	0,976	0,981	0,985	0,987
17	0,851	0,869	0,892	0,910	0,954	0,977	0,981	0,985	0,987
18	0,858	0,874	0,897	0,914	0,956	0,978	0,982	0,986	0,988
19	0,863	0,879	0,901	0,917	0,957	0,978	0,982	0,986	0,988
20	0,868	0,884	0,905	0,920	0,959	0,979	0,983	0,986	0,988
21	0,873	0,888	0,908	0,923	0,960	0,980	0,983	0,987	0,989
22	0,878	0,892	0,911	0,926	0,961	0,980	0,984	0,987	0,989
23	0,881	0,895	0,914	0,928	0,962	0,981	0,984	0,987	0,989
24	0,884	0,898	0,916	0,930	0,963	0,981	0,984	0,987	0,989
25	0,888	0,901	0,918	0,931	0,964	0,981	0,985	0,988	0,989
26	0,891	0,904	0,920	0,933	0,965	0,982	0,985	0,988	0,989
27	0,894	0,906	0,923	0,935	0,965	0,982	0,985	0,988	0,990
28	0,896	0,908	0,924	0,936	0,966	0,982	0,985	0,988	0,990
29	0,898	0,910	0,926	0,937	0,966	0,982	0,985	0,988	0,990
30	0,900	0,912	0,927	0,939	0,967	0,983	0,985	0,988	0,990
31	0,902	0,914	0,929	0,940	0,967	0,983	0,986	0,988	0,990
32	0,904	0,915	0,930	0,941	0,968	0,983	0,986	0,988	0,990
33	0,906	0,917	0,931	0,942	0,968	0,983	0,986	0,989	0,990
34	0,908	0,919	0,933	0,943	0,969	0,983	0,986	0,989	0,990
35	0,910	0,920	0,934	0,944	0,969	0,984	0,986	0,989	0,990
36	0,912	0,922	0,935	0,945	0,970	0,984	0,986	0,989	0,990
37	0,914	0,924	0,936	0,946	0,970	0,984	0,987	0,989	0,990
38	0,916	0,925	0,938	0,947	0,971	0,984	0,987	0,989	0,990
39	0,917	0,927	0,939	0,948	0,971	0,984	0,987	0,989	0,991
40	0,919	0,928	0,940	0,949	0,972	0,985	0,987	0,989	0,991
41	0,920	0,929	0,941	0,950	0,972	0,985	0,987	0,989	0,991
42	0,922	0,930	0,942	0,951	0,972	0,985	0,987	0,989	0,991
43	0,923	0,932	0,943	0,951	0,973	0,985	0,987	0,990	0,991
44	0,924	0,933	0,944	0,952	0,973	0,985	0,987	0,990	0,991
45	0,926	0,934	0,945	0,953	0,973	0,985	0,988	0,990	0,991
46	0,927	0,935	0,945	0,953	0,974	0,985	0,988	0,990	0,991
47	0,928	0,936	0,946	0,954	0,974	0,985	0,988	0,990	0,991
48	0,929	0,937	0,947	0,954	0,974	0,985	0,988	0,990	0,991
49	0,929	0,938	0,947	0,955	0,974	0,985	0,988	0,990	0,991
50	0,930	0,939	0,947	0,955	0,974	0,985	0,988	0,990	0,991

10 Apêndice C

Nesta seção estão expressos os resultados estatísticos obtidos dos dados coletados, utilizando o software estatístico R.

Tabela 14: Estatísticas do Tempo de Convergência para uma rede de 25 nós

	Média	Mediana	Variância	P-value
HTR	10.9089	11.8645	4.03432	-
S-HTR (MD5)	10.7734	11.347	4.16098	0.4939
S-HTR (SHA-1)	10.9914	11.9215	4.31476	0.7013
S-HTR (SHA-256)	10.8736	11.82	3.92691	0.9082
S-HTR (SHA-384)	10.8601	11.5965	4.07412	0.9951
S-HTR (SHA-512)	11.1145	12.02	3.89412	0.1137

Tabela 15: Estatísticas da Sobrecarga de Pacotes para uma rede de 25 nós

	Média	Mediana	Variância	P-value
HTR	24.81811	24.9596	0.8880298	0.4772
S-HTR (MD5)	50.00024	49.7439	4.218547	0.2246
S-HTR (SHA-1)	51.07435	50.8863	4.066153	0.8941
S-HTR (SHA-256)	53.67764	53.54745	4.312635	0.5649
S-HTR (SHA-384)	57.0525	57.07585	4.386293	0.8471
S-HTR (SHA-512)	60.496	60.6174	4.88251	0.6214

Tabela 16: Estatísticas da Perda de Pacotes para uma rede de 25 nós

	Média	Mediana	Variância	P-value
HTR	0.03150522	0.315668	3.92×10^{-5}	0.2336
S-HTR (MD5)	0.04162181	0.417686	5.26×10^{-5}	0.8308
S-HTR (SHA-1)	0.04162181	0.417686	6.09×10^{-5}	0.1862
S-HTR (SHA-256)	0.04323088	0.0436981	5.79×10^{-5}	0.4507
S-HTR (SHA-384)	0.0454672	0.04518545	7.21×10^{-5}	0.7886
S-HTR (SHA-512)	0.04841239	0.04761505	6.88×10^{-5}	0.627

Tabela 17: Estatísticas do Tempo de Convergência para uma rede de 49 nós

	Média	Mediana	Variância	P-value
HTR	13.5826	13.9315	2.57424	-
S-HTR (MD5)	14.1353	14.06	2.30167	7.79×10^{-5}
S-HTR (SHA-1)	14.0437	14.127	2.70231	2.86×10^{-5}
S-HTR (SHA-256)	13.8757	14.0665	2.07677	0.00728
S-HTR (SHA-384)	14.2113	14.201	2.30024	4.67×10^{-8}
S-HTR (SHA-512)	14.3633	14.2345	1.67117	1.75×10^{-11}

Tabela 18: Estatísticas da Sobrecarga de Pacotes para uma rede de 49 nós

	Média	Mediana	Variância	P-value
HTR	86.0567	86.08225	9.178353	0.3672
S-HTR (MD5)	172.1915	171.386	33.71105	0.7332
S-HTR (SHA-1)	174.6682	174.1115	36.25775	0.5326
S-HTR (SHA-256)	180.0172	180.8895	31.41976	0.3728
S-HTR (SHA-384)	187.267	187.5995	29.90704	0.5632
S-HTR (SHA-512)	194.6257	194.4995	32.6906	0.8996

Tabela 19: Estatísticas da Perda de Pacotes para uma rede de 49 nós

	Média	Mediana	Variância	P-value
HTR	0.180764	0.179909	0.00025417	0.06106
S-HTR (MD5)	0.2874945	0.284277	0.000563596	0.122
S-HTR (SHA-1)	0.2889983	0.28711015	0.000535156	0.7903
S-HTR (SHA-256)	0.2959729	0.2915805	0.000615811	0.7146
S-HTR (SHA-384)	0.305078	0.3066005	0.000634744	0.5832
S-HTR (SHA-512)	0.3193177	0.314372	0.000608591	0.1221

Tabela 20: Estatísticas do Tempo de Convergência para uma rede de 64 nós

	Média	Mediana	Variância	P-value
HTR	15.1311	15.2405	1.81269	-
S-HTR (MD5)	15.5256	15.466	2.50775	0.00043
S-HTR (SHA-1)	15.515	15.5035	2.57142	0.00061
S-HTR (SHA-256)	15.4529	15.46	1.94983	0.0017
S-HTR (SHA-384)	15.5954	15.5595	2.35425	2.39×10^{-6}
S-HTR (SHA-512)	15.6973	15.5295	2.47284	4.43×10^{-7}

Tabela 21: Estatísticas da Sobrecarga de Pacotes para uma rede de 64 nós

	Média	Mediana	Variância	P-value
HTR	144.8212	144.633	16.87174	0.9884
S-HTR (MD5)	290.1404	289.7965	75.81012	0.9214
S-HTR (SHA-1)	292.0548	290.552	73.94997	0.5458
S-HTR (SHA-256)	297.8396	297.504	67.11343	0.456
S-HTR (SHA-384)	306.8994	307.473	78.19411	0.9423
S-HTR (SHA-512)	318.175	318.1115	80.24961	0.7812

Tabela 22: Estatísticas da Perda de Pacotes para uma rede de 64 nós

	Média	Mediana	Variância	P-value
HTR	0.3821136	0.3809455	0.000538097	0.9321
S-HTR (MD5)	0.6345774	0.628336	0.001130914	0.1039
S-HTR (SHA-1)	0.6323586	0.6326405	0.001601752	0.7375

S-HTR (SHA-256)	0.6425647	0.6398755	0.001219598	0.5329
S-HTR (SHA-384)	0.6654849	0.664198	0.00149129	0.1143
S-HTR (SHA-512)	0.6882289	0.694979	0.0011681448	0.1258

Tabela 23: Estatísticas do Tempo de Convergência para uma rede de 81 nós

	Média	Mediana	Variância	P-value
HTR	16.1147	15.866	1.65759	-
S-HTR (MD5)	16.796	16.1595	3.52052	2.17×10^{-8}
S-HTR (SHA-1)	16.8092	16.273	3.07812	5.46×10^{-9}
S-HTR (SHA-256)	16.8551	16.1575	3.54735	5.45×10^{-8}
S-HTR (SHA-384)	16.9354	16.4295	2.77726	1.54×10^{-14}
S-HTR (SHA-512)	16.981	16.535	2.9436	8.06×10^{-16}

Tabela 24: Estatísticas da Sobrecarga de Pacotes para uma rede de 81 nós

	Média	Mediana	Variância	P-value
HTR	230.3133	230.759	42.75082	0.2531
S-HTR (MD5)	458.6965	457.311	108.1617	0.4628
S-HTR (SHA-1)	461.9514	460.656	104.0678	0.9534
S-HTR (SHA-256)	470.367	469.71	131.0022	0.2876
S-HTR (SHA-384)	482.9753	482.3495	126.6112	0.3467
S-HTR (SHA-512)	492.6647	493.8125	164.3268	0.14436

Tabela 25: Estatísticas da Perda de Pacotes para uma rede de 81 nós

	Média	Mediana	Variância	P-value
HTR	0.7471615	0.7468215	0.001518058	0.5665
S-HTR (MD5)	1.270211	1.272015	0.002998972	0.786
S-HTR (SHA-1)	1.277238	1.27099	0.002846175	0.3538
S-HTR (SHA-256)	1.308756	1.30404	0.002594086	0.5607
S-HTR (SHA-384)	1.327874	1.329055	0.003013369	0.417
S-HTR (SHA-512)	1.36875	1.37308	0.003098143	0.09341

11 Bibliografia

[1] H. Chen and N. Yu, “A survey of communication technology in distribution network,” in *China International Conference on Electricity Distribution (CICED)*, 2012, pp. 1 – 8.

[2] Corson and Macker, “RFC 2501,” 1999. [Online]. Available: http://tools.ietf.org/html/rfc2501. [Accessed: 29-Jan-2014].

[3] M. Dipobagio, “An Overview on Ad Hoc Networks,” pp. 1–9.

[4] B. Branden, D. Clark, S. Crocker, and C. Huitema, “RFC 1636,” 1994. [Online]. Available: http://tools.ietf.org/html/rfc1636. [Accessed: 09-Dec-2013].

[5] “CERT.org.” [Online]. Available: http://www.cert.org/stats/#vuls. [Accessed: 09-Dec-2013].

[6] “CERT.br.” [Online]. Available: http://www.cert.br/stats/incidentes/. [Accessed: 09-Dec-2013].

[7] W. Stallings, *Criptography and Networking Security*, 4th ed. Pearson Education, 2007.

[8] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, “Security in Mobile Ad Hoc Networks: Challenges and Solutions,” *IEEE Wirel. Commun.*, no. February, pp. 38–47, 2004.

[9] J. Junior, “Specification and Evaluation of a Heterogeneous MANET Protocol,” Federal University of Pernambuco, 2009.

[10] E. Souto, R. Aschoff, J. L. Junior, R. Melo, D. Sadok, and J. Kelner, “HTR: A framework for interconnecting wireless heterogeneous devices,” in *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, 2012, pp. 645–649.

[11] D. Sadok, R. Aschoff, J. Kelner, and E. Souto, “Network Address Allocation Method,” 2009.

[12] J. Lima, T. Rodrigues, R. Melo, G. Correia, J. Kelner, and E. Feitosa, “On the tuning of wireless heterogeneous routing,” *2013 IEEE 24th Annu. Int. Symp. Pers. Indoor, Mob. Radio Commun.*, pp. 3391–3396, Sep. 2013.

[13] J. Junior, “An Energy-aware Multipath Routing Extension for Heterogeneous Ad Hoc Networks,” Universidade Federal de Pernambuco, 2013.

[14] “RFC 3626.” [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt. [Accessed: 24-Feb-2014].

[15] T. H. Cormen, C. E. Leirserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. MIT Press, 2001.

[16] Krawczyk, “RFC 2104,” *1997*. [Online]. Available: http://www.ietf.org/rfc/rfc2104.txt. [Accessed: 09-Feb-2014].

[17] S. Agrawal, S. Jain, and S. Sharma, “A Survey of Routing Attacks and Security Measures in Mobile Ad-Hoc Networks,” vol. 3, no. 1, pp. 41–48, 2011.

[18] P. Yi, Z. Dai, Y. Zhong, and S. Zhang, “Resisting flooding attacks in ad hoc networks,” *Int. Conf. Inf. Technol. Coding Comput. - Vol. II*, pp. 657–662 Vol. 2, 2005.

[19] A. Bandyopadhyay, S. Vuppala, and P. Choudhury, “A simulation analysis of flooding attack in MANET using NS-3,” *2011 2nd Int. Conf. Wirel. Commun. Veh. Technol. Inf. Theory Aerosp. Electron. Syst. Technol. (Wireless VITAE)*, pp. 1–5, Feb. 2011.

[20] D. Glynos, P. Kotzanikolaou, and C. Douligeris, “Preventing Impersonation Attacks in MANET with Multi-Factor Authentication,” *Third Int. Symp. Model. Optim. Mobile, Ad Hoc, Wirel. Networks*, pp. 59–64.

[21] U. Venkanna and R. L. Velusamy, “Black Hole Attack and their Counter Measure Based on Trust Management in Manet: A Survey,” *Proc. Int. Conf. Adv. Recent Technol. Commun. Comput.*, pp. 232 – 236, 2011.

[22] O. M. Ad and H. Networks, “Analysis the Node Isolation Attack Against,” pp. 30–35, 2006.

[23] Y. Hu, A. Perrig, and D. B. Johnson, “Packet Leashes : A Defense against Wormhole Attacks in Wireless Networks,” vol. 00, no. C, pp. 1976–1986, 2003.

[24] T. Andel and A. Yasinsac, “The invisible node attack revisited,” *Proc. 2007 IEEE SoutheastCon*, pp. 686–691, 2007.

[25] “ns-3.” [Online]. Available: http://www.nsnam.org/. [Accessed: 31-Jan-2014].

[26] “Crypto++,” 2004. [Online]. Available: http://www.cryptopp.com/. [Accessed: 14-Jan-2014].

[27] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution: From Theory to Practice*. John Wiley & Sons, Ltd, 2009.

[28] J. G. Andrews, A. Ghosh, and R. Muhamed, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*. 2007.

[29] R. van Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*. Artech House, 2000.

[30] F. Lassabe, P. Canalda, P. Chatonnay, F. Spies, and O. Baala, “A Friis-based calibrated model for WiFi terminals positioning,” in *World of Wireless Mobile and Multimedia Networks*, 2005, pp. 382 – 387.

[31] H. Zhu and M. Li, “A survey of quality of service in IEEE 802.11 networks,” *Wirel. Commun. IEEE*, vol. 11, no. 4, pp. 6 – 14, 2004.

[32] D. R. Anderson, D. J. Sweeney, and T. A. Williams, *Statistics for Business and Economic*, 11th ed. South-Western CENGAGE Learning, 2011.

[33] Estatcamp, “Teste Wilcoxon Pareado,” *1997*, 2011. [Online]. Available: http://www.portalaction.com.br/974-3-teste-de-wilcoxon-pareado. [Accessed: 05-Feb-2014].

[34] Estatcamp, “Teste de Shapiro-Wilk,” *1997*, 2011. [Online]. Available: http://www.portalaction.com.br/content/64-teste-de-shapiro-wilk. [Accessed: 05-Feb-2014].

[35] S. Turner and L. Chen, “RFC 6151,” 2011. [Online]. Available: https://tools.ietf.org/html/rfc6151. [Accessed: 14-Feb-2014].

[36] “Cryptographic Hash Funcion.” [Online]. Available: http://en.wikipedia.org/wiki/Cryptographic_hash_function. [Accessed: 12-Feb-2014].

Universidade Federal de Pernambuco – UFPE
Centro de Informática - CIn

Recife, 28 de fevereiro de 2014