

# PEARL Tools: um Conjunto de Ferramentas para Avaliação da Eficiência de Algoritmos de Criptografia em Dispositivos Móveis

Matheus P. Hernandez<sup>2</sup>, Windson Viana<sup>1</sup>, Bringel Filho<sup>1</sup>, Rossana M. C Andrade<sup>1,2</sup>  
e André Jalles Monteiro<sup>3</sup>

<sup>1</sup>Mestrado em Ciência da Computação (MDCC), <sup>2</sup>Departamento de Computação (DC),  
<sup>3</sup>Departamento de Estatística (DE) – Universidade Federal do Ceará (UFC)

{matheus, windson, bringel}@lia.ufc.br, {jalles, rossana}@ufc.br

**Abstract.** *Many applications for mobile devices (MDs) transfer and receive sensitive information that needs to be protected. However, the limited computational power of MDs imposes new challenges during the implementation of security solutions. The possibility of discovering previously if a certain cryptographic algorithm runs in a satisfactory time in a certain mobile device and the choice for the most appropriate cryptographic algorithm for each MD have become critical factors for the security solutions implementations. In this paper presents a tool set that allows the efficiency evaluation of cryptographic algorithms in MDs and the storage of the evaluation results in a web repository.*

**Resumo.** *Muitas aplicações para dispositivos móveis (DMs) transmitem e possuem acesso a informações críticas que necessitam de proteção. Contudo, o poder computacional limitado dos DMs impõe novos desafios durante a implementação de soluções de segurança. A possibilidade de descobrir previamente se um determinado algoritmo de criptografia executa em um tempo satisfatório em um DM, bem como a escolha dos algoritmos mais adequados para um determinado dispositivo, tornaram-se fatores críticos para a implementação de soluções de segurança. Este artigo apresenta um conjunto de ferramentas que permite avaliar a eficiência de algoritmos de criptografia em DMs além de construir um repositório Web com os resultados das avaliações.*

## 1. Introdução

Uma enorme diversidade de dispositivos móveis (DMs) tem surgido no mercado com a possibilidade de conexão a redes IP através das tecnologias de comunicação sem fio. Esses DMs oferecem conectividade à Internet permitindo ao usuário obter acesso a informações a qualquer momento e em qualquer lugar. Além disso, os DM fornecem aos usuários não somente o *software* do fabricante (e.g., agenda, calculadora), mas também a possibilidade de instalação de novas aplicações implementadas em plataformas de desenvolvimento suportadas por esses dispositivos, como por exemplo, J2ME (Java 2 Micro Edition) [11] e Superwaba [12]. Algumas dessas aplicações para DMs operam com informações críticas (e.g., aplicações corporativas, de *Mobile Banking* e de *Mobile Commerce*), sendo assim, elas requerem serviços de segurança, tais

como o provimento de confidencialidade e integridade. Entretanto, os mecanismos presentes nos sistemas de comunicação sem fio (e.g., Bluetooth) para a provisão de segurança ainda apresentam vulnerabilidades que permitem a ocorrência de ataques ativos e passivos [9][7]. Além disso, a maioria desses mecanismos está limitada a proteger apenas o meio sem fio, não provendo a segurança fim-a-fim desejada às aplicações.

Devido às vulnerabilidades dos mecanismos presentes nos sistemas de comunicação sem fio e do desempenho requerido pelo uso de protocolos de segurança fim-a-fim, tais como o SSL/TLS e VPNs [6], soluções de criptografia devem ser adicionadas às aplicações, como apresentadas [1][4]. Contudo, certos algoritmos de criptografia que são eficientes quando executados em processadores de alto desempenho disponíveis em computadores *desktops* podem ser ineficientes em processadores de baixo desempenho como os presentes nos DMs mencionados anteriormente. Além disso, a pequena quantidade de memória para a execução das aplicações disponibilizada pelas plataformas de desenvolvimento, reduz desempenho de algoritmos que necessitam de grande quantidade de memória durante a execução. Portanto, a limitação do poder computacional e memória desses dispositivos impõem novos desafios para a implementação de criptosistemas.

Nesse contexto, os dados resultantes da avaliação da eficiência de algoritmos de criptografia nos DMs tornam-se informações fundamentais para a escolha do algoritmo ou da combinação de algoritmos adequados para um determinado DM, bem como para identificar quais dispositivos podem executar de forma satisfatória uma combinação de algoritmos pré-estabelecida. Este artigo apresenta um conjunto de ferramentas, intitulado de PEARL Tools (i.e., a tool set for Performance EvaluAtion of cryptogRaphic aLgorithms in mobile devices), que permite a avaliação da eficiência de algoritmos de criptografia em dispositivos móveis e a construção de um repositório Web com essas informações. Este conjunto é formado basicamente por três ferramentas: o Mobile PEARL, o Web Receptor e o Web PEARL Analyser.

Diferentemente dos trabalhos que aparecem na literatura [8][5], este conjunto de ferramentas permite a avaliação da eficiência de algoritmos de criptografia em uma grande variedade de dispositivos móveis (e.g., celulares, Palms, Pocket PCs). Para isso, a ferramenta que avalia a eficiência nos dispositivos, o Mobile PEARL, é implementada em J2ME, que é uma plataforma suportada pela grande maioria dos DMs atuais e largamente utilizada por desenvolvedores de aplicações. Além disso, o Mobile PEARL, através do Web Receptor, permite transferir para um repositório Web os dados das avaliações realizadas. Esses dados podem ser consultados por desenvolvedores de criptosistemas utilizando o Web PEARL Analyser, que disponibiliza gráficos comparativos e tabelas dos resultados das avaliações armazenados no repositório.

Na próxima seção serão apresentados os trabalhos relacionados, em seguida, a seção 3 descreve a abordagem de avaliação da eficiência dos algoritmos criptográficos utilizada pelas PEARL Tools. A seção 4 descreve em detalhes as PEARL Tools, que será demonstrada no estudo de caso descrito na seção 5. Por fim, a seção 6 apresenta as conclusões e direcionamentos a trabalhos futuros.

## 2. Trabalhos Relacionados

Em [8], os autores implementaram e avaliaram bibliotecas de sistemas de criptografia para a plataforma Palm OS. Essas bibliotecas incluem implementações na linguagem C de cifradores de fluxo (SSC2, ARC4 e SEAL 3.0), de cifradores de bloco (Rijndael, DES, DESX e TripleDES), de funções hash (MD2, MD4, MD5 e SHA-1) e de operações aritméticas inteiras de precisão múltipla. Os algoritmos foram avaliados em dispositivos Palm V (processador de 16 MHz, 2MB de memória RAM) e Palm IIIc (processador de 20 MHz, 8MB de memória RAM). As avaliações consistiram em execuções desses algoritmos nesses dispositivos através da cifragem de amostras de texto de vários tamanhos (2 Kb, 50 Kb e 4Mb), tendo como resultado a quantidade de bytes cifrados (ou resumidos, no caso de funções hash) por segundo (bytes/s). Com o resultado da avaliação, os autores de [8] identificaram que o SSC2 teve melhor desempenho do que o ARC4 e o SEAL 3.0 para amostras de texto pequenas (1KB). Em amostras de texto grandes (4MB), o SEAL 3.0 executou duas vezes mais rápido que o SSC2. Da análise dos cifradores de bloco, os autores observaram que o Rijndael é quatro vezes mais rápido que o DES.

Em [5], os autores mostram a análise de desempenho de protocolos de criptografia em dispositivos móveis. A análise é aplicada aos protocolos SSL, S/MIME e IPsec, os quais são amplamente utilizados por aplicações desenvolvidas para redes fixas. Os resultados da análise mostram que o tempo necessário para executar funções de criptografia é pequeno, não causando impacto significativo no desempenho de transações móveis de tempo real. A análise foi realizada somente no dispositivo iPAC H3630 (processador 200 Mhz StrongARM, 32MB de memória RAM, sistema operacional Windows CE Pocket PC 2002). Entretanto, o dispositivo analisado tem desempenho similar ao de um *desktop*, bem como os autores não investigaram em DMs com poder computacional menor e em outras plataformas de desenvolvimento.

## 3. Uma Abordagem de Avaliação da Eficiência

A abordagem utilizada para a avaliação da eficiência de algoritmos de criptografia em DMs é dividida em três fases. A primeira consiste em coletar informações relacionadas à execução dos algoritmos no DM, chamadas de amostras. Nessa fase são coletadas informações sobre o tempo de inicialização do algoritmo e o tempo de execução para cada tamanho de entrada. Para isso, o tamanho inicial do texto de entrada (1KB por padrão) é duplicado até atingir o limite especificado para avaliação (512KB por padrão) ou o tamanho máximo suportado pelo dispositivo (e.g., 8KB para dispositivos Palm com máquina virtual Sun J2ME). A avaliação da eficiência do algoritmo torna-se mais precisa com esta variação no tamanho da entrada, permitindo a análise da variação nos tempos de execução e de inicialização de acordo com o aumento do tamanho do texto.

A segunda fase contempla a filtragem destas informações buscando identificar os dados mais significativos, bem como calcular a velocidade (i.e., bytes/s) das operações criptográficas. Essa filtragem consiste no cálculo do valor mais significativo que representa a execução de um algoritmo para um determinado tamanho de entrada, que é calculada utilizando o tamanho do texto de entrada (e.g., 1024, 2048 e 4096 bytes) e o valor da moda dos tempos de execução e de inicialização para cada tamanho. A moda é utilizada, pois é necessária uma medida rápida e próxima que represente o valor

mais típico da distribuição. Portanto, terá como resultado a velocidade que ocorre com maior frequência para cada tamanho de texto e algoritmo avaliado. Caso a moda não possa ser calculada, ou seja, caso todos os resultados possuam valores diferentes, o valor da mediana é tomado como valor mais significativo da distribuição. A terceira fase necessita da interação do usuário com a ferramenta, que consiste no envio das velocidades calculadas no DM para um servidor Web para serem avaliados mais criteriosamente.

#### 4. PEARL Tools

Para validar a abordagem apresentada na seção anterior, foram implementadas as PEARL Tools, que consiste em três ferramentas: o Mobile PEARL, o Web Receptor e o Web PEARL Analyser.

Na Figura 1 são apresentadas as ferramentas e a forma como elas interagem entre si. O desenvolvedor de criptosistemas se cadastra no site do PEARL Tools [3] e realiza o *download* do Mobile PEARL para o DM no qual será executada a avaliação (passo 1). Após a configuração dos parâmetros da simulação a avaliação é realizada e os dados das amostras são armazenados no *Samples Local Data – SLD* (passo 2). Com o término das simulações, os dados são filtrados através do cálculo da moda ou da mediana como foi detalhado na seção 3. Os dados filtrados são enviados para o Web Receptor que, após a validação, são armazenados no repositório Web de simulações (passo 3). Assim, o desenvolvedor pode utilizar o Web PEARL Analyser para gerar gráficos comparativos dos algoritmos no dispositivo e consultar tabelas contendo dados sobre a eficiência dos algoritmos nos outros dispositivos cujos resultados de simulações estão também armazenados no repositório (passo 4). A seguir as três ferramentas são descritas.

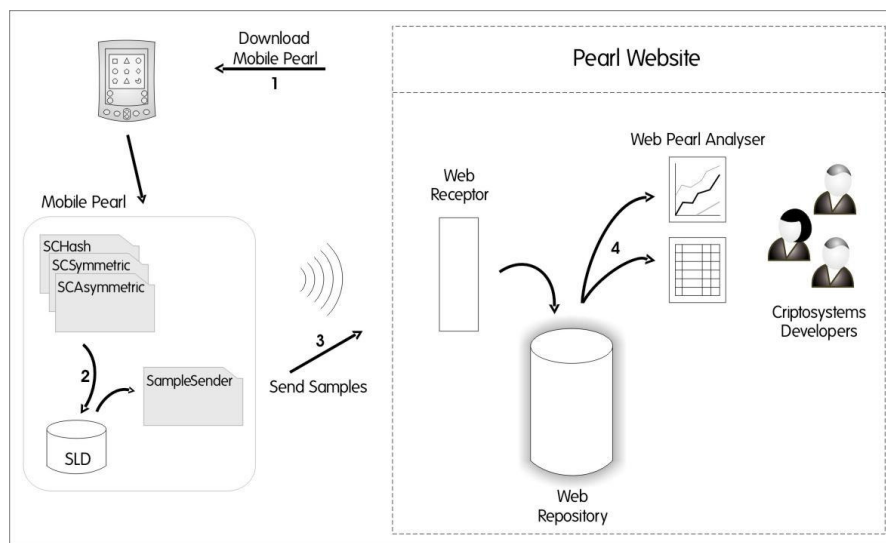


Figura 1 - Arquitetura do PEARL Tools

#### 4.1. Mobile PEARL

O Mobile PEARL é a ferramenta responsável por coletar as informações sobre a eficiência dos algoritmos nos DMs, que é uma evolução da ferramenta apresentada em [2]. Podemos citar como pontos dessa evolução a maneira como a ferramenta coleta, avalia e organiza os dados, além da mudança na abordagem utilizada. A inclusão do repositório web e do Web PEARL Analyser também complementam o progresso obtido desde a publicação dos resultados iniciais em [1].

É importante ressaltar que a eficiência de um algoritmo em uma plataforma de desenvolvimento para DMs está diretamente relacionada à complexidade do algoritmo, ao tipo de operação utilizada, à implementação do algoritmo e à quantidade de memória disponível para a aplicação que faz a avaliação. Além disso, no caso de plataformas de programação baseadas em Java, tais como J2ME e Superwaba, a implementação da máquina virtual também pode influir no resultado. Portanto, os dados obtidos da avaliação de eficiência devem ser analisados criteriosamente, caso o desenvolvedor utilize outra plataforma de programação ou até mesmo, outra implementação dos algoritmos. A ferramenta Mobile PEARL foi implementada na plataforma Java Micro Edition (i.e., J2ME) e os algoritmos de criptografia utilizados são implementados e distribuídos pelo Legion of the Bouncy Castle [9]. Para permitir a avaliação de novos algoritmos, serão necessárias modificações no código que irão requer recompilação e reinstalação da ferramenta.

O Mobile PEARL é formado por quatro MIDlets que são as unidades básicas de um programa MIDP/J2ME [10]. São eles: SampleSender, SCSymmetric, SCAsymmetric, e SCHash. O SampleSender é o MIDlet capaz de filtrar as informações das simulações contidas no SLD e enviá-las ao Web Receptor.

O SCSymmetric é responsável pelas simulações dos algoritmos de criptografia simétrica, que são divididos em duas categorias: cifradores de bloco e cifradores de fluxo. Este MIDlet permite avaliar qualquer um dos dezessete cifradores de bloco a seguir: AES, AES Fast, AES Light, Blowfish, CAST5, CAST6, 3DES, DES, IDEA, RC2, RC5 32 bits, RC5 64 bits, RC6, Rijndael, Serpent, Skipjack, e Towfish, bem como o cifrador de fluxo RC4. Alguns desses algoritmos são variações de algoritmos existentes, tais como o AES Fast e o AES Light (variações do AES). Outros são implementações diferentes do mesmo algoritmo, como Rijndael e AES. Para cada algoritmo cifrador de bloco há quatro modos: Electronic Codebook (ECB), Cipher Block Cleaning (CBC), Cipher Feedback (CFB) e Output Feedback (OFB). Sendo assim, quatro simulações são executadas para cada algoritmo e tamanho de entrada; cada uma delas aplica um dos modos de cifragem mencionados anteriormente, com a mesma chave de 128 bits.

O SCAsymmetric é responsável pelas simulações dos algoritmos de criptografia assimétrica. Foram utilizados dois algoritmos: RSA e ElGamal. E por fim, o SCHash simula as funções *hash*. No total, permite a simulação de dez algoritmos: MD2, MD4, MD5, RIPEMD128, RIPEMD160, SHA 256, SHA 384, SHA 512, SHA-1 e Tiger.

#### 4.2. Web Receptor e Web PEARL Analyser

Como descrito anteriormente, além do Mobile PEARL outras duas ferramentas compõem o PEARL Tools: o Web Receptor e Web PEARL Analyser. O Web Receptor

é uma página Web desenvolvida em PHP que recebe e valida os dados enviados pelo SampleSender, que é o MIDlet invocado pelo usuário após as fases de simulação e de filtragem das amostras.

Após a validação, o Web Receptor armazena os dados da simulação no repositório Web que poderão ser analisados pelos desenvolvedores. A análise desses dados é realizada através da ferramenta Web PEARL Analyser. Esta ferramenta disponibiliza um conjunto de consultas que permite aos desenvolvedores realizar comparações da eficiência dos algoritmos em um mesmo DM para diversos tamanhos de entrada, apresentando tabelas com os valores de desempenho em bytes/s, além de gráficos comparativos.

## 5. Estudo de Caso

Como estudo de caso, a ferramenta Mobile PEARL foi executada em um Palm m130 e nos celulares Nokia 6820 e Sony Ericsson P800 (Tabela 1). Em cada DM foram executados os algoritmos de criptografia simétrica, assimétrica e funções hash. O tamanho inicial do texto avaliado foi de 1KB sendo duplicado até atingir 256KB.

**Tabela 1- Dados Técnicos**

Dispositivo	RAM	Processador	SO	VM
Palm m130	8MB	Motorola Dragonball 33 MHz	Palm OS 4.1	IBM & Sun
Sony-Ericsson P800	32MB	32-bit RISC ARM9 @ 156 MHz	Symbian OS 7.0	Sony Ericsson
Nokia 6820	3.5MB	-----	Symbian OS 7.0	Nokia

Em cada DM foram executadas dez simulações para cada tamanho de entrada e algoritmo, correspondendo a 680 (seiscentos e oitenta) execuções de algoritmos simétricos cifradores de bloco, 10 (dez) execuções do algoritmo cifrador de fluxo RC4 e 100 (cem) execuções de funções *hash*. Os cifradores assimétricos RSA e ElGamal não tiveram tempos satisfatórios de eficiência e não serão abordados por esse motivo.

Na Tabela 2 são apresentados os tempos de execução do algoritmo de função *hash* SHA-1, do cifrador de fluxo RC4 e do cifrador de bloco DES para entradas de tamanho 8KB e 64KB. Pode-se observar através desta tabela que além do poder computacional do DM simulado e da quantidade de memória disponível, a máquina virtual utilizada também influi nos resultados dos testes. Isso demonstra que ambos, DM e máquina virtual devem ser considerados no momento da análise.

**Tabela 2 - Tempos de Execução**

Dispositivos Analisados	SHA-1 (8KB)	SHA-1 (64KB)	RC4 (8KB)	RC4 (64KB)	DES (8KB)	DES (64KB)
P800	125ms	953ms	47ms	312ms	125ms	1016ms
Palm M130 IBM	9760ms	77570ms	4750ms	37990ms	14630ms	117000ms
Palm M130 SUN	10230ms	-----	4380ms	-----	13000ms	-----
Nokia 6820	1746ms	13917ms	599ms	-----	2261ms	-----

Para todos os tamanhos de entrada simulados, o RC4 obteve menor tempo de execução, enquanto o Rijndael apresentou o maior tempo de execução dentre os algoritmos de criptografia simétrica. Em relação aos algoritmos de função *hash*, o MD4 foi o mais rápido, enquanto o MD2 o mais lento.

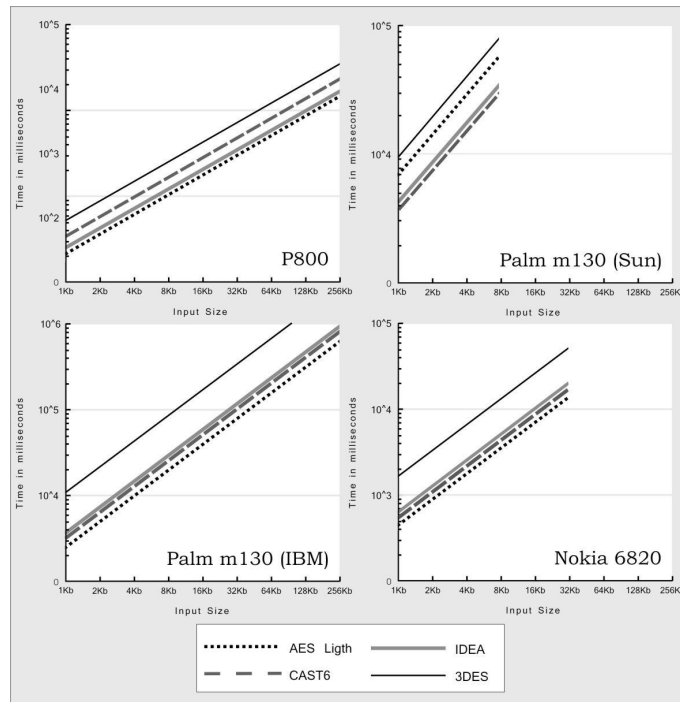


Figura 2 - Gráficos Comparativos

Na Figura 2, encontram-se gráficos gerados através do Web Pearl Analyser para os algoritmos cifradores de bloco AES Light, CAST6, IDEA e 3DES. Os gráficos dos resultados da avaliação nos dispositivos Palm m130 (utilizando a VM da Sun) e Nokia 6820 apresentam quebras devido às limitações da máquina virtual e de memória, respectivamente. Ainda com relação à Figura 2, de acordo com os gráficos dos dispositivos podemos verificar que o CAST6 foi mais lento que o IDEA no dispositivo Sony-Ericsson P800, enquanto no dispositivo Nokia 6820 ocorreu o inverso. Isso demonstra a influência exercida pelo DM utilizado na execução dos algoritmos.

## 6. Conclusão e Trabalhos Futuros

A avaliação da eficiência dos algoritmos de criptografia é um requisito para o desenvolvimento de criptosistemas eficiente em dispositivos de baixo poder computacional. A abordagem apresentada neste artigo e suas respectivas ferramentas permitem avaliar a eficiência dos algoritmos de criptografia em uma grande variedade de dispositivos móveis (e.g., Palms, celulares, Pocket PCs).

As principais contribuições deste trabalho podem ser resumidas em: (1) a ferramenta Mobile PEARL, que implementa a abordagem de simulação, permite a avaliação da eficiência de algoritmos de criptografia implementados em J2ME para vários DMs; (2) os resultados da avaliação possibilitam verificar a viabilidade da aplicação de um algoritmo em determinados DM; (3) os resultados permitem também identificar os algoritmos que possuem o melhor desempenho para um certo tamanho de entrada; (4) a ferramenta Mobile PEARL pode avaliar o desempenho de um algoritmo em diferentes VMs (i.e., máquinas virtuais) no mesmo DM, o que proporciona também a comparação do desempenho das máquinas virtuais; (5) os resultados da avaliação permitem a construção de criptosistemas otimizados para os DMs analisados; e, por fim,

(6) a infra-estrutura do PEARL Tools permite que desenvolvedores possam consultar o repositório Web sobre a eficiência dos algoritmos através do Web PEARL Analyser e também que novos desenvolvedores possam realizar simulações em outros dispositivos, aumentando a base de informações de avaliações contidas no repositório.

Como trabalho futuro, se propõe a implementação de um framework que permita o desenvolvimento de aplicações seguras para dispositivos móveis na plataforma J2ME. Por exemplo, com os resultados da avaliação apresentados neste artigo, um desenvolvedor é capaz de escolher os algoritmos a serem adicionados ao *framework* e garantir a segurança na transmissão e armazenamento dos dados no DM.

## Referências

1. VIANA, Windson, CASTRO, R. M. C., MACHADO, Javam, FILHO, Bringel, MAGALHAES, Katy, GIOVANO, Carlo. Mobis: A Solution For The Development Of Secure Applications For Mobile Device. In: ICT - International Conference on Telecommunication, 2004. Lecture Notes in Computer Science, v.3124, p.1015-1022,2004.
2. FILHO, Bringel; VIANA, Windson; CASTRO, R. M. C. PEARL: a Performance evaluator of cryptographic algorithms for mobile devices. In: The First International Workshop on Mobility Aware Technologies and Applications. Florianópolis-PR, Brasil, Outubro 2004. Lecture Notes in Computer Science. MATA Conference, v.3284, 2004.
3. Site do PEARL Tools. *Download* do Mobile PEARL e o uso do Web PEARL Analyser. Disponível em: <<http://PEARL.lia.ufc.br>>. Acesso em: 08 dez 2004.
4. W. Itani and A. Kayssi. "J2ME End-to-End Security for M-Commerce", in Proc. IEEE Wireless Communications and Networking Conference (WCNC 2003), p. 2015 - 2020, March 2003, New Orleans, Louisiana.
5. ARGYROUDIS, P. G.;VERMA, R.;TEWARI, H.;O'Mayony, D. Performance Analysis of Cryptographic Protocols on Handheld Devices. In Proceedings of 3rd IEEE International Symposium on Network Computing and Applications (NCA'04), pp 169-174, 2004.
6. NUNES, Bruno, A. A.; MORAES, Luís F. M. Avaliando a Sobrecarga Introduzida nas Redes 802.11 pelos Mecanismos de Segurança WEP e VPN/IPSec. Workshop de segurança - WSEG. In: Anais SBRC 2003.
7. SILVA, G. M.; SOUZA, J. N. Uma análise dos mecanismos de segurança de redes locais sem fio e uma proposta de melhoria. III Workshop de Segurança de Sistemas Computacionais, Workshop de segurança - WSEG. In: Anais do SBRC 2003, Natal, 2003.
8. WONG, D. S.; FUENTES, H. H.; CHAN, A. H. The Performance Measurement of Cryptographic Primitives on Palm Devices. 17th Annual Computer Security Applications Conference. New Orleans, Louisiana, p. 10-14, dec. 2001.
9. STALLINGS, William. Network security essentials: applications and standards. Prentice Hall, Inc. 2000. ISBN: 0-13-016093-8.
10. The Legion of Bouncy Castle. Cryptography API for Java. Disponível em: <<http://www.bouncycastle.org/>>. Acesso em 12 de fevereiro de 2003.
11. Java 2 Platform Micro Edition. Disponível em: <<http://java.sun.com/j2me/>>. Acesso em 2004.
12. Superwaba: The Java VM for PDAs. Disponível em: <<http://www.superwaba.com.br>>. Acesso em: 15 fev. 2005.