Análise do desempenho de algoritmos criptográficos em dispositivos móveis

Thiago H. Silva[1], **Douglas G. Macharet**[1], **César F. Teixeira**[1]

[1] Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

{thiagohs,doug,cesar}@dcc.ufmg.br

Abstract. *With the considerable increase in the number of mobile devices, especially cell phones, there is also a demand for more sophisticated services, and many of these services require cryptographic protection. Considering that, this work has as objective the analysis of key encryption algorithms, when applied to mobile devices. The chosen algorithms are the most representatives in each category of cryptographic algorithms: symmetrical, asymmetrical and hash. With the help of experiments in real devices we were able to identify the algorithms with the better performance in each category. It was also shown that the frequency of the processor alone does not explain the performance of an algorithm, and that the architecture of the manufacturers has also significantly influence.*

Resumo. *O aumento considerável no número de dispositivos móveis e a demanda por serviços mais sofisticados fazem com que, a cada dia, seja maior a exigência de segurança nestes dispositivos. Assim sendo, esse trabalho tem como objetivo a análise dos principais algoritmos de criptografia, quando aplicados em dispositivos móveis. Os algoritmos escolhidos são os representantes mais utilizados das categorias de algoritmos criptográficos: simétricos, assimétricos e de código de verificação. Com o auxílio de experimentos em dispositivos reais foi possível identificar os algoritmos de melhor desempenho em cada categoria. Foi mostrado também que a frequência do processador não explica sozinha o desempenho de um algoritmo, e que as arquiteturas dos fabricantes também influenciam significativamente.*

1. Introdução

É notável o crescente aumento no número de dispositivos móveis a cada ano. Com isso, também ocorre uma maior demanda por serviços voltados para tais dispositivos, serviços esses que necessitam de uma segurança cada vez maior. Como exemplo podemos citar, bancos online, compra de conteúdo (músicas, imagens, sons, etc), email, entre outros. Isso faz com que seja necessário um maior entendimento sobre o comportamento dos atuais algoritmos de criptografia em tais aparelhos.

Recentemente pode-se observar que tem crescido o interesse no estudo de métodos criptográficos voltados para sistemas embarcados, especialmente dispositivos móveis [Kocher et al. 2004], [Argyroudis et al. 2004], [Potlapally et al. 2003]. Esse é um tema relevante porque sistemas embarcados usualmente possuem várias limitações, demandando assim estudos para melhor lidar com tais restrições.

Normalmente, quando existe a necessidade de se proteger determinado tipo de dado, são usados algoritmos criptográficos que já possuem eficiência comprovada em computadores convencionais ou servidores de alta capacidade. Entretanto, tais algoritmos utilizam de forma considerável os processadores desses dispositivos. Isso se deve ao fato de se basearem em várias operações aritméticas complexas, operações as quais a maioria dos dispositivos móveis pode não estar preparada a executar de forma eficiente. Apesar do progresso e melhoria da capacidade dos dispositivos móveis, especialmente os celulares, categoria de dispositivos analisadas neste trabalho, eles ainda permanecem possuindo recursos limitados, como baixa capacidade de processamento e memória, que são essenciais dependendo do conteúdo que se deseja criptografar.

Dessa forma, um algoritmo que executa em pouco tempo em um computador convencional pode ser extremamente lento em um aparelho de celular, e em alguns casos, devido a algumas limitações, o dispositivo pode ser incapaz de executar o algoritmo, por exemplo, por falta de memória. Como exemplo disso podemos citar o caso do aparelho Motorola V3 que foi incapaz de executar o algoritmo AES (*Advanced Encryption Standard*) para arquivos maiores que 50 Kbytes, apresentando uma exceção de falta de memória. Logo não é apenas necessário assegurar que o dado esteja seguro, mas também deve-se levar em consideração as limitações do dispositivo onde o algoritmo de criptografia está sendo executado.

Com isso em foco, é essencial que seja realizada uma avaliação das técnicas de criptografia utilizadas atualmente para proteção de dados nos dispositivos móveis. É importante que tais técnicas sejam avaliadas de forma a assegurar um grau razoável de proteção aos dados e ao mesmo tempo respeitando as limitações dos dispositivos. É nesse contexto que este artigo está inserido, possuindo como principal objetivo, comparar e avaliar vários tipos de algoritmos criptográficos aplicados em telefones celulares, de forma a determinar as melhores abordagens, isto é, aquelas que oferecem o melhor desempenho.

Assim, é discutido o comportamento dos principais algoritmos de criptografia quando executados aparelhos celulares. Os algoritmos criptográficos analisados representam os algoritmos mais utilizados das categorias: algoritmos simétricos, assimétricos e de geração de código de verificação (mais detalhes na seção 2.1). Os celulares utilizados: Motorola A1200, V3, V220 e Nokia N95, 7373 não foram escolhidos aleatoriamente e tentam representar o mercado mundial de celulares (mais detalhes na seção 3.3).

Através de experimentos em ambientes reais, calculamos os tempos de execução dos algoritmos estudados. Como resultado de nossas análises foram identificados os melhores algoritmos criptográficos de todas categorias analisadas, bem como, constatou-se que a frequência do processador não é o único fator impactante no desempenho desses algoritmos, e que as arquiteturas, que estão relacionadas aos fabricantes, também influenciam no resultado final.

O resto deste trabalho está dividido da seguinte forma: Na seção 2, discutimos os trabalhos relacionados e abordamos as categorias de algoritmos de criptografia utilizadas neste trabalho. Na seção 3, apresentamos a metodologia de avaliação utilizada neste trabalho. Na seção 4, apresentamos e discutimos os resultados encontrados. E por fim na seção 5, abordamos as conclusões e considerações finais.

2. Trabalhos Relacionados

Nenhum artigo foi encontrado na literatura que tenha realizados os mesmos testes e análises deste trabalho, logo, acreditamos que esse seja o primeiro trabalho com tais características.

O trabalho realizado por [Wong et al. 2001] realiza uma avaliação do desempenho dos algoritmos criptográficos em uma plataforma Palm [url 2007f]. Os algoritmos testados foram implementados na linguagem C, e um computador de mão foi utilizado nos testes. É interessante ressaltar que, nenhum algoritmo específico de chave pública (mais detalhes na seção 3) foi testado. Deve-se enfatizar também que a análise foi realizada em apenas um aparelho, com boa capacidade computacional equivalente a um computador pessoal com baixa capacidade, assim não representando da melhor forma os dispositivos móveis, e por consequência impedindo que conclusões mais genéricas acerca dos algoritmos fossem obtidas. Podemos dizer que esse trabalho é o que mais se assemelha ao que está sendo proposto neste artigo. Ele difere do nosso trabalho pois analisamos as principais categorias de algoritmos criptográficos, utilizando dispositivos móveis mais representativos e realizamos análises mais completas dos resultados obtidos

Uma outra análise de desempenho de protocolos de segurança em dispositivos móveis também pode ser encontrada em [Argyroudis et al. 2004]. Foram avaliados os protocolos SSL (*Secure Sockets Layer*), S/MIME (*Secure Multipurpose Internet Mail Extensions*) e IP-SEC (*IP Security Protocol*), concluindo que o tempo necessário para se realizar as operações necessárias era pequeno, não afetando o desempenho geral. Entretanto essas análises foram realizadas em um dispositivo móvel que possui desempenho equivalente ao de um computador convencional simples, sendo ainda um representante com baixa expressividade na categoria de dispositivos portáteis.

Em [Potlapally et al. 2003] é mostrado que um fator crítico no uso de sistemas embarcados está ligado ao consumo de energia. Nesse artigo, foi realizada uma análise com relação ao consumo de energia de vários algoritmos criptográficos em computadores de mão. Ao final são discutidos melhoramentos que poderiam ser feitos visando a implementação de algoritmos de segurança mais eficientes do ponto de vista do consumo de energia.

A popularização de dispositivos móveis, assim como o desenvolvimento de aplicações para esses aparelhos, é um fenômeno recente, o que faz com que as pesquisas nessa área ainda estejam em desenvolvimento. Devido a esse fato, poucos estudos sobre esse tema são encontrados na literatura. Isso aumenta a relevância do trabalho aqui apresentado.

2.1. Algoritmos Criptográficos

A criptografia é uma técnica utilizada para se comunicar de forma segura, e vem sendo utilizada há muitos anos. Atualmente, impulsionada com o advento da Internet, a necessidade de se estabelecer canais de comunicação seguros é indispensável. Criptografia baseia-se em técnicas matemáticas para tornar uma mensagem inteligível apenas àquele ao qual ela foi dirigida. Além disso, criptografia pode ser utilizada para garantir que determinado texto ou informação realmente partiu de quem se presume ser o remetente. Ou seja, criptografia também pode ser usada na realização de assinatura de mensagens.

O campo da criptografia moderna pode ser dividido em diversas áreas de estudo, sendo que os principais algoritmos são analisados neste artigo. Esses algoritmos podem ser classificados nas categorias: Algoritmos de Chaves Simétricas e Algoritmos de Chaves Públicas. Além desses dois, um terceiro conjunto denominado Algoritmos de Geração de Código de Verificação, que não utilizam chaves, também foi analisado, pois são uma classe importante devido a sua ampla utilização.

Algoritmos de chaves simétricas referem-se a métodos de criptografia em que ambos emissor e receptor compartilham a mesma chave. Os estudos modernos de algoritmos de chaves simétricas englobam cifras de blocos e cifras de fluxo. Algoritmo de chave simétrica que utilizam cifra de bloco recebem como entrada um bloco contendo o texto puro e retornam um bloco de mesmo tamanho com o texto criptografado. Como geralmente o conteúdo que se deseja criptografar contém múltiplos blocos, a mensagem pura tem que ser subdividida em blocos de tamanhos específicos para ser criptografada e reagrupada quando se deseja reaver o conteúdo criptografado. Os algoritmos AES e RC5 (*Rivest Cipher 5* [Rivest 1995]), avaliados neste trabalho, são exemplos de algoritmos de chave simétrica que utilizam cifra de blocos. Além de cifra de bloco existe uma outra técnica denominada cifra de fluxo. Um algoritmo de chave simétrica que utiliza cifra de fluxo é o RC4 (*Rivest Cipher 4*). Esse algoritmo recebe como entrada a mensagem pura completa, criptografa *byte* a *byte* essa mensagem, e retorna todo conteúdo criptografado de uma vez.

Já nos algoritmos de chave pública (ou assimétricos), o receptor possui uma chave privada e uma pública. A chave pública é distribuída para todas as pessoas com as quais se deseja trocar dados. Sua chave privada é guardada em segredo. O emissor que deseja comunicar com esse receptor em específico aplica a chave pública do receptor em um conjunto de dados que deseja enviar. O receptor, então, recebe os dados codificados aplica sua chave privada à mensagem codificada, para torná-la compreensível novamente. Como exemplo de algoritmo assimétrico, que utiliza o procedimento descrito acima, temos o RSA. Dentro dos conjuntos de algoritmos de chaves públicas, temos também os algoritmos denominados "acordo", DH (*Diffie-Hellman* [Rescorla 1999]) e ECDH (*Elliptic Curve Diffie-Hellman*), que utilizam um protocolo de estabelecimento de chaves. A técnica utilizada no DH e ECDH também possibilita dois usuários quaisquer estabelecerem a mesma chave de forma segura, mas o procedimento para isso difere da forma utilizada no algoritmo RSA (a sigla corresponde aos últimos nomes de seus criadores *Ron Rivest, Adi Shamir, and Leonard Adleman*).

Além de criptografia, os algoritmos de chaves públicas também podem realizar assinaturas de mensagens, ou seja, garantir que uma mensagem realmente partiu do remetente esperado. Para tal, um emissor usa sua chave privada para criptografar a mensagem ou o código de verificação da mesma. Essa etapa é a assinatura da mensagem. Então, o receptor aplica a chave pública do emissor, para descodificar a mensagem ou o código de verificação, e a partir disso é capaz de dizer se a mensagem partiu ou não do emissor esperado. Como exemplo desses algoritmos podemos citar o DSA (*Digital Signature Algorithm* [FIPS 1994]) e sua variante utilizando curvas elípticas, ECDSA (*Elliptic Curve DSA*).

Por fim, existem algoritmos que não propriamente criptográficos mas servem de apoio para realização da criptografia, por exemplo, certificando a integridade de uma mensagem. Esses algoritmos utilizam operações matemáticas, sobre o conjunto de dados a ser enviado, e geram um número (código de verificação) correspondente. Esse número é enviado juntamente com a mensagem. O receptor, então, ao receber ambos (mensagem e código de verificação), gera o número correspondente à mensagem que recebeu, e verifica se o número que ele gerou e o recebido são idênticos. Se forem, acredita-se que o dado não sofreu alterações. Para gerar o número, são geralmente utilizadas técnicas de geração de código de verificação. Nessa categoria podemos citar os algoritmos MD5 (*Message-Digest algorithm 5* [Rivest 1992]), SHA1 (*Secure Hash Algorithm* [D. Eastlake and Jones 2001]), entre outros.

Nesse trabalho serão avaliados os algoritmos que constam na tabela 1. Como pode ser observado, a categoria de algoritmos assimétricos possue somente um representante, o RSA. Isso se deve à falta de algoritmos dessa categoria tendo utilização expressiva atualmente. Assim, especialmente na categoria de algoritmos assimétricos não iremos contrastar algoritmos, e sim somente apresentaremos os dados coletados com relação à execução do RSA.

Tabela 1. Algoritmos escolhidos para cada categoria.

Hash	Simétricos	Assinatura	Troca de chaves	Assimétricos
MD5	AES	DSA	DH	RSA
SHA1	RC5	ECDSA	ECDH	
SHA256	RC4			

2.2. Outros Estudos

Na literatura é possível ainda encontrar outros estudos sobre criptografia em sistemas embarcados e móveis. Esses estudos não realizam análises de algoritmos de criptografia, como proposto aqui neste trabalho, mas possuem uma abordagem descritiva da arquitetura dos sistemas criptográficos, exploram seus problemas e desafios, ou propõem alguma melhoria em hardware ou software. Alguns trabalhos são: [Kocher et al. 2004], [Ravi et al. 2002] e [Potlapally et al. 2002].

3. Metodologia de Avaliação

3.1. Objetivos

As nossas análises possuem os seguintes propósitos:

- Identificar os algoritmos em cada categoria que apresentam melhor desempenho.
- Checar a hipótese de que a arquitetura do celular, ou seja, arquitetura do processador, características da memória e outros, são fundamentais na classificação dos aparelhos por seus respectivos desempenhos em todos os algoritmos testados. Assim, deseja-se mostrar que a freqüência do processador, isoladamente, não é capaz de prever qual dispositivo apresentará melhor desempenho que outro no conjunto de algoritmos testados.
- Analisar a hipótese de que o fabricante do dispositivo, mais especificamente no caso deste trabalho Nokia ou Motorola, poderia explicar uma parcela significativa do tempo de execução.

3.2. Parâmetros e Métricas

A métrica que será analisada é o tempo de execução total dos algoritmos, que foi medido com um ambiente de testes descrito na seção seguinte. O termo "total" significa que, o tempo de execução será a soma do tempo de execução do processo principal, por exemplo, na categoria de algoritmos simétricos o tempo total gasto é a soma dos tempos gastos para a criptografia e descriptografia.

Existem diversos parâmetros que interferem no tempo de execução. O primeiro é o algoritmo, já que algoritmos diferentes realizam operações diferentes, que exigem tempos de execução diferentes. O segundo parâmetro é o tamanho da entrada, que será fornecida a cada algoritmo, para a realização da criptografia. Vale ressaltar que para os algoritmos de "acordo" (DH e ECDH), excepcionalmente, o tamanho da entrada não é um parâmetro, já que o algoritmo estabelece uma chave para as futuras trocas de dados, ao invés de realizar a criptografia sobre os dados de entrada. Portanto, sempre que for feita referência ao tamanho da entrada para essa classe de algoritmos o que será variado é o tamanho da chave de criptografia utilizada. O terceiro parâmetro é o aparelho celular em si, pois aparelhos diferentes possuem arquiteturas diferentes, que podem beneficiar ou punir determinadas operações de determinados algoritmos.

O terceiro parâmetro, o celular, encapsula diversos outros parâmetros que precisariam ser analisados isoladamente. Por exemplo, o tamanho da memória, taxa de transferência da memória, frequência do processador, arquitetura do processador e outros. Entretanto, pelo fato de um dispositivo móvel não ser configurável, por exemplo, não é possível trocar a memória de um aparelho, a análise de cada um desses parâmetros seria muito difícil. Este trabalho então irá tratar o aparelho como um parâmetro único, porém, algumas de suas características serão discutidas quando necessárias.

Nosso projeto experimental considera como fatores todos os parâmetros descritos. Isso se deve ao fato de todos serem importantes na determinação do tempo de execução.

3.3. Metodologia

Os algoritmos utilizados nos testes fazem parte de uma biblioteca de criptografia bastante conhecida, a BouncyCastle [url 2007b]. Essa biblioteca possui implementações em Java e C# e já é utilizada há vários anos, gerando assim uma maior confiança na correta implementação dos algoritmos.

Como realizamos experimentos reais, foi desenvolvido um programa em J2ME para a realização de testes. Esse programa executa em um dispositivo real, permitindo a escolha do algoritmo e do tamanho do dado de entrada, apresentando em seguida informações relativas à execução do algoritmo. A informação de execução que utilizamos em nossas análises é o tempo de execução total. Os tempos "totais"registrados em cada categoria de algoritmos pelo programa são:

- Algoritmos simétricos: Soma do tempo para criptografia e de descriptografia.
- Algoritmos de "acordo": Soma do tempo para criar a chave pública e de criar a chave compartilhada.
- Algoritmos assimétricos de assinatura: Soma do tempo para assinar e verificar a assinatura de um conteúdo.
- Algoritmos de geração de código de verificação: Tempo para criar o código de verificação.

Na figura 1 podemos observar as etapas de realização de um teste, onde primeiramente é escolhido o tamanho do arquivo, escolhe-se então o algoritmo a ser executado, e em seguida as informações obtidas são exibidas.

Figura 1. Execução do programa coletor de informações.

Classificamos os aparelhos atuais em três categorias, denominadas: Baixa, Média e Alta. A categoria Baixa representa aparelhos com processamento de 0 a 50 MHz. A categoria Média representa os aparelhos de 51 a 150 MHz. Já a categoria Alta representa os aparelhos com processamento maiores que 151 MHz.

Para a escolha dos aparelhos que pertenceriam a cada categoria foi realizada uma análise na página da Internet Amazon.com [url 2007a] e foi possível assim identificar os aparelhos mais populares, fazendo com que os testes realizados tivessem uma representatividade maior dos aparelhos presentes no mercado.

A tabela 2 exibe os aparelhos selecionados. Foram utilizados apenas aparelhos da Nokia e Motorola pois, de acordo com [url 2007c], essas empresas são as que possuem maior representatividade no mercado.

Tabela 2. Relação de aparelhos selecionados em cada categoria.

Baixa (0 a 50 MHz)	**Média** (51 a 150 MHz)	**Alta** (acima de 150 MHz)
Motorola V220 (16,2 MHz)	Nokia 7373 (72,2 MHz) Motorola V3 (66,1 MHz)	Motorola A1200 (320 MHz) Nokia N95 (332 MHz)

Para obtenção das frequências dos processadores foram utilizadas as informações fornecidas pelos fabricantes Motorola [url 2007d] e Nokia [url 2007e], bem como o projeto [url 2007g], que tem a finalidade de disponibilizar a frequência do processador de telefones celulares, sendo útil para telefones antigos.

Para os tamanhos de entrada foram escolhidos tamanhos típicos de arquivos multimídia ou de texto. Os tamanhos em *bytes* dos arquivos a serem utilizados nos testes são mostrados na tabela 3. Excepcionalmente, para os algoritmos de "acordo", para os quais o tamanho da chave criptográfica é o parâmetro, foram utilizadas chaves de 512, 768 e 1024 bits.

Tabela 3. Tamanho em *bytes* dos arquivos escolhidos para os testes.

Texto	Jpeg	Mid
23	1233	15223
500	2487	
	5024	
	7873	
	53278	

Nosso experimento contou com a realização de 400 testes diferentes. Todos esses testes foram repetidos três vezes.

4. Resultados

4.1. Comparação pareada de algoritmos

O conjunto de experimentos descritos nesta seção visa determinar o desempenho relativo dos diferentes algoritmos considerados em cada classe (tabela 1), para diferentes aparelhos e tamanhos de entrada.

A seguir, os algoritmos serão comparados e será dado um percentual para representar o quão melhor um algoritmo foi em relação ao outro. Esse percentual significa que, se o algoritmo mais ineficiente estiver sendo usado, a substituição do mesmo pelo mais eficiente, representa um ganho no percentual dado. Por exemplo, o algoritmo A for 10% mais eficiente que o algoritmo B cujo tempo de execução foi de 100ms. Então, se substituirmos B por A, haverá um ganho de 10%, pois o tempo de execução de A foi de 90ms.

Em relação aos algoritmos simétricos, o RC4 apresentou melhor desempenho que o RC5 e o AES, em ambos os casos com 90% de confiança. Além disso, a substituição do algoritmo RC5, o segundo mais eficiente, pelo RC4, representa um ganho de 65.7% no tempo de execução. Dentre os algoritmos de geração de código de verificação, o SHA1 foi 11% mais eficiente que o MD5, com confiança de 40%. Entretanto, se retirarmos o N95 da análise, no qual, diferentemente do que ocorreu nos outros aparelhos, o MD5 foi significativamente superior ao SHA1, provavelmente devido a características únicas da arquitetura do N95, então a confiança sobe para 60% e o SHA1 apresenta-se 15% mais eficiente que o MD5. O SHA256 teve pior desempenho que o SHA1 e que o MD5, ambos com 90% de confiança.

O algoritmo DH, na classe de algoritmos de "acordo", foi superior ao ECDH (versão do DH utilizando curvas elípticas), com confiança de 90%. O resultado contradiz a literatura, que afirma que a adição de curvas elípticas a um algoritmo resultaria em melhora no tempo de execução em relação à versão sem curvas elípticas. Já na classe de assinatura, o DSA foi 10% superior ao ECDSA, com 40% de confiança.

À essa confiança não se pode atribuir muito significado, mas o DSA apresentou em alguns casos, uma tendência a ser superior ao ECDSA (versão do DSA com curvas elípticas), o que poderia sugerir outro contra-exemplo à literatura.

A partir dessa análise, pode-se concluir que existem algoritmos que foram significativamente melhores que outros, em uma mesma classe. Assim, a hipótese de que a escolha do algoritmo é relevante ao desempenho é válida, portanto, o algoritmo deve sim ser escolhido de maneira cuidadosa, pois uma escolha incorreta pode ter um severo impacto no desempenho. Além disso, os algoritmos utilizando curvas elípticas não apresentaram o desempenho inicialmente esperado com relação à suas implementações sem curvas elípticas.

4.2. Execução do RSA

Como o RSA, representante da categoria de algoritmos assimétricos, não possue um algoritmo similar com grande utilização atualmente foi decidido não realizar comparações de algoritmos nessa classe. Especialmente para essa categoria de algoritmos assimétricos apenas iremos apresentar os dados relativos à execução do RSA obtidos com os aparelhos utilizados. Esses resultados podem ser úteis para se ter uma visão do comportamento desse algoritmo em diversas classes de aparelhos celulares.

Na figura 2 pode-se observar os tempos de execução total do RSA obtidos com os aparelhos celulares Motorola A1200, Nokia N95 e 7373. Como pode-se observar na figura 2, apesar do Nokia 7337, um aparelho da classe Média, apresentar uma tendência de maior crescimento do tempo de execução total, comparado com os aparelhos da classe Alta, ele apresentou um bom desempenho.

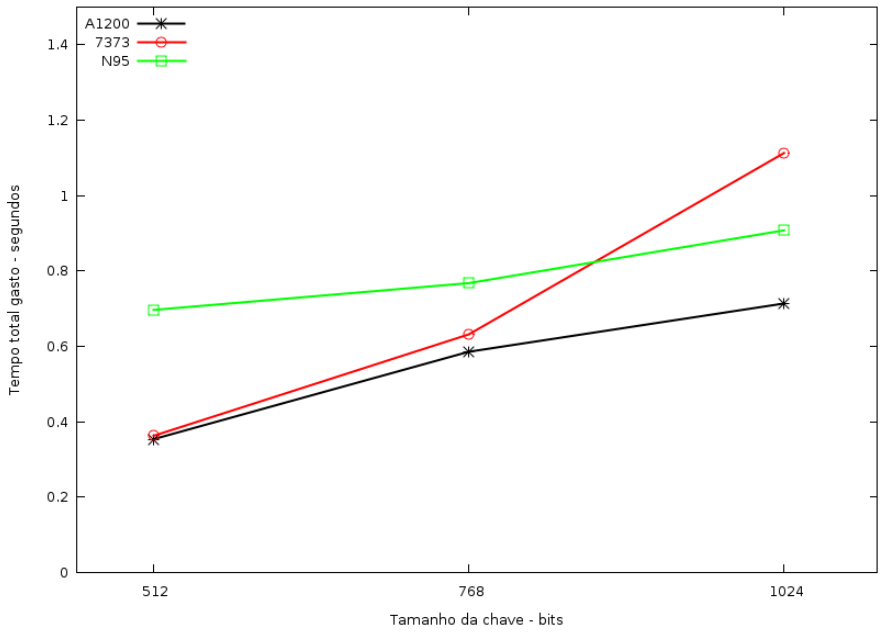

Figura 2. Tempos de execução do RSA nos aparelhos A1200, N95 e 7373

O restante dos resultados da execução total do RSA, para os outros aparelhos utilizados, estão disponíveis na figura 3. A separação dos resultados em dois gráficos foi necessária para uma melhor visualização dos mesmos, uma vez que, alguns tempos obtidos foram bastantes diferentes uns dos outros. Nota-se o elevado tempo gasto por esses aparelhos, isso pode inviabilizar várias aplicações, principalmente as que necessitam de respostas rápida.

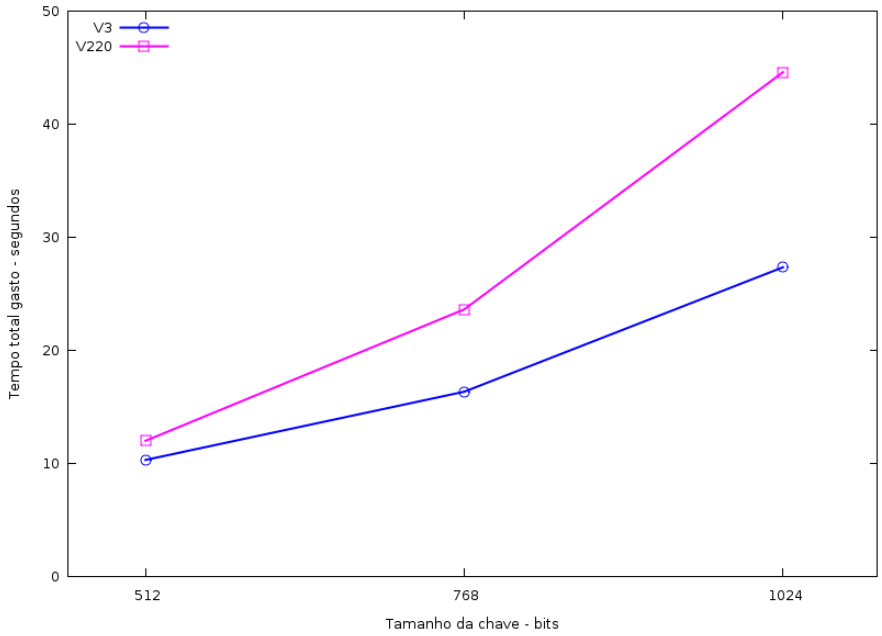

Figura 3. Tempos de execução do RSA nos aparelhos V3 e V220

4.3. Regressão Linear

Baseando-se nos testes pareados realizados anteriormente pôde-se avaliar os algoritmos que obtiveram melhor desempenho na faixa de tamanhos de arquivo utilizadas neste trabalho. Mas, não se pode afirmar sobre o comportamento caso o tamanho da entrada seja superior aos dados coletados, afinal, apesar de este trabalho ter utilizado entradas usuais, mesmo assim entradas maiores e incomuns podem sim ocorrer. Portanto, como efeito colateral dessa análise, seria interessante determinar qual a tendência de comportamento dos algoritmos analisados.

Para abordar essa questão optamos por realizar regressões para cada combinação de aparelho com algoritmo. Com isso, deseja-se saber como esses algoritmos, em cada celular, se comportam com o aumento da entrada. Dessa forma, é possível prever se um algoritmo, possui uma taxa de crescimento maior que outro, denunciando que para entradas maiores aquele algoritmo tende a se tornar mais ineficiente, e portanto, menos recomendado.

Com o objetivo de validar essa abordagem, inicialmente verificamos se regressões lineares fazem sentido, pois o crescimento no tempo de execução pode não ser linear em função da entrada. Para essa verificação, foram criados gráficos para cada combinação aparelho com algoritmo variando-se o tamanho da entrada, em seguida calculado o R^2 para todas curvas.

Para ilustrar as regressões realizadas, citaremos duas classes de algoritmos para o aparelho Nokia 95. A figura 4 exibe os resultados encontrados para os algoritmos simétricos, obtidos para o N95. Os R^2 das regressões lineares para esses algoritmos, nesse aparelhos são: AES 0,90; RC4 0,99; RC5 0,94. Apesar da regressão linear do AES explicar 90%, a inspeção visual da figura 4 na curva representada pelo AES pode sugerir uma falta de linearidade. De qualquer forma, na sua categoria, o RC4 ainda apresenta forte tendência de superioridade.

A figura 5 apresenta os dados para os algoritmos de geração de código de verificação obtidos no N95. Através de uma inspeção visual pode-se observar o comportamento linear, os R^2 dessas regressões também indicam isso: MD5 0,99; SHA1 0,91; SHA256 0,98.

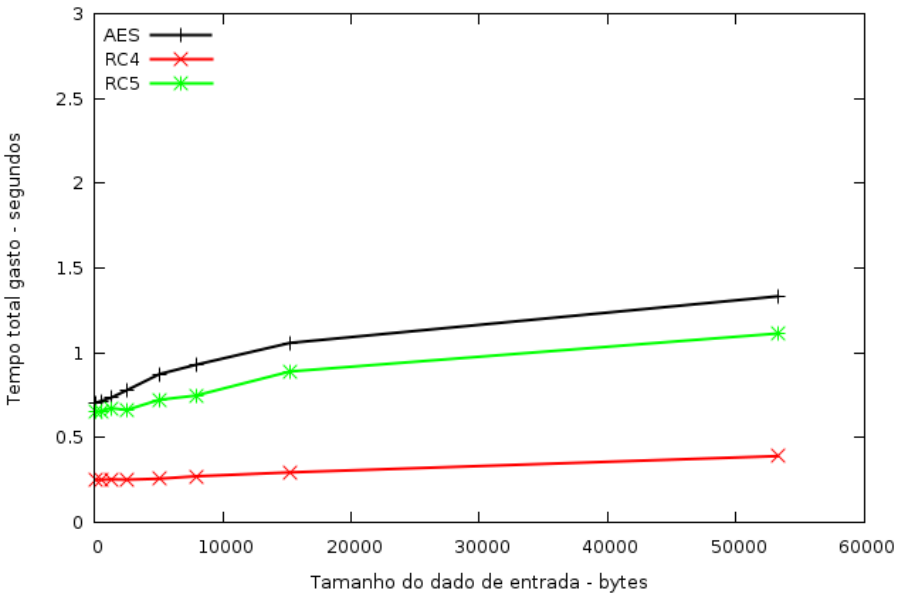

Figura 4. Algoritmos simétricos executados no N95.

A tabela 4 apresenta os coeficientes de determinação obtidos para as regressões lineares, ou seja, os parâmetros A e B da reta Ax + B obtida e o R^2 para as regressões lineares: aparelho x algoritmo. Os valores apresentados estão na ordem A; B; R^2. O termo *NS* significa não significativo, o que quer dizer que o intervalo de confiança do parâmetro incluiu o zero, assim, não se pode dizer que seja diferente de zero.

Em sua grande maioria, os valores foram acima de 95%. Entretanto, houve algumas regressões que apresentaram coeficientes baixos, como a regressão do algoritmo SHA256 no aparelho V3. Por isso, não podemos afirmar que todos algoritmos apresentam comportamento linear, porém em comparação com a regressão exponencial, a linear foi mais significativa. A tabela ainda apresenta as inclinações das retas obtidas e o ponto em que a reta intercepta o eixo y.

Em relação aos algoritmos simétricos, a regressão linear do algoritmo RC4, para todos os aparelhos, apresentou uma inclinação menor que o RC5 e AES, dessa forma, o algoritmo continuará a apresentar melhor desempenho que seus concorrentes com o crescimento da entrada a ser criptografada. Todas as análises com 90% de confiança.

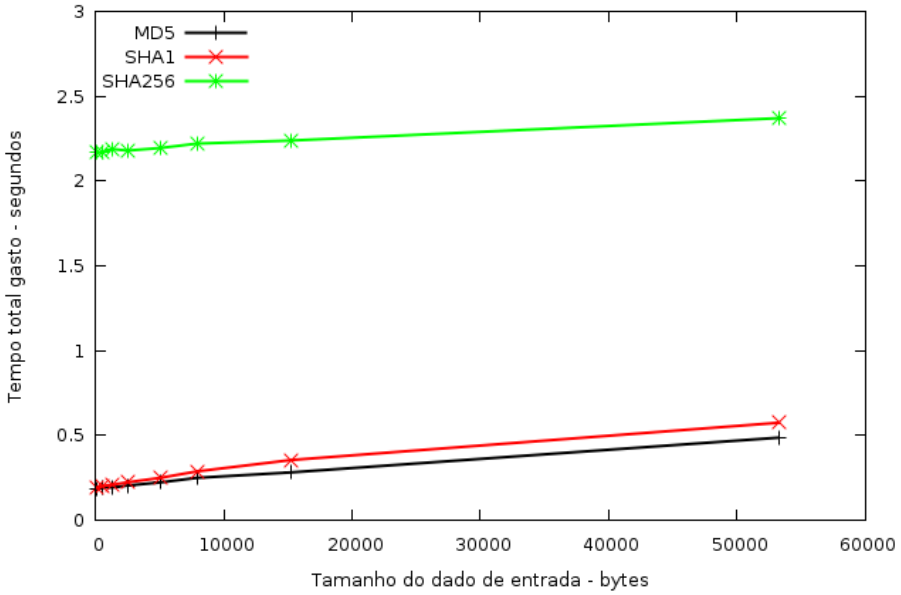

Figura 5. Algoritmos de geração de código de verificação executados no N95.

Tabela 4. Tabela de valores das regressões lineares.

	A1200	N95	7373	V3	V220
RC4	0.002; 129.2; 0.83	0.003; 250.5; 0.99	0.002; 79.88; 0.99	0.24; 349.6; 1	0.16; 332.6; 0.99
RC5	0.006; 263.1; 0.94	0.009; 671.9; 0.94	0.02; 263.1; 0.99	0.7; 905.4; 0.99	0.68; 913.9; 0.99
AES	0.008; 337.62; 0.97	0.01; 769.1; 0.90	0.03; 272.4; 0.99	0.9; 1402.5; 0.99	0.88; 1444.6; 0.99
MD5	0.002; 127.6; 0.64	0.006; 194.3; 0.99	0.004; 28.7; 0.99	0.2; 260.8; 0.99	0.18; 265.3; 1
SHA1	0.002; 111.8; 0.75	0.003; 229.1; 0.91	0.0034; 27.9; 0.99	0.19; *NS*; 0.91	0.13; 263.7; 0.99
SHA256	0.004; 2043.1; 0.94	0.004; 2176.9; 0.98	0.01; 5189.2; 0.97	0.38; 101232; 0.54	0.21; 180609; 0.99
DH	*NS*; *NS*; 0.98	*NS*; 0.4; 0.97	*NS*; 9.4; 0.97 & *NS*	*NS*; 546.03; 0.97	*NS*; 951.1; 0.97
ECDH	*NS*; 6668; 0.99	*NS*; 6672; 0.98	*NS*; 143.1; 0.99	*NS*; 8150; 0.97	*NS*; 9103; 0.99

O algoritmo SHA1, na classe de geração de código de verificação, foi melhor que o MD5 e SHA256, com 90% de confiança nos aparelhos 7373 e V220. Entretanto, SHA1 foi indistinto do MD5 no A1200 e do SHA256 para o N95.

Os algoritmos de "acordo" apresentaram regressões cuja inclinação, com confiança de 90%, incluíram o zero. Dessa forma, os algoritmos podem apresentar comportamento constante. É importante mencionar que para os algoritmos de "acordo", o essencial é a segurança relacionada à troca das chaves. Assim, foram variados os tamanhos das chaves.

O algoritmo RSA apresentou comportamento linear. Porém, como a nossa análise foi feita com poucos pontos, em todos os aparelhos foi impossível diferenciar se o comportamento esperado para o algoritmo é uma reta ou um valor constante. Além disso, houve uma grande variabilidade também em relação ao ponto de intersecção obtido pela regressão.

Como pôde ser observado, os algoritmos que apresentaram melhor desempenho nos testes pareados, tendem a apresentar melhor desempenho também para entradas superiores às utilizadas nos experimentos.

Assim, a decisão por empregar os algoritmos de melhor desempenho da seção 4.1, também parece fazer fazer sentido para entradas de tamanhos superiores. No entanto, para aumentar a confiança dessa afirmação uma maior variedade de tamanhos de arquivos são necessários no experimento.

Um resultado inusitado foi o obtido para os algoritmos de "acordo", que podem ser constantes. Isso, provavelmente se deve ao fato de os testes terem sido realizados para um pequeno conjunto de chaves (três), o que resultou em um intervalo de confiança (I.C.) mais largo. Além disso, as chaves foram escolhidas por sua popularidade, e a maior chave utilizada é apenas o dobro da menor, o que resultou em uma pequena diferença entre os tempos de execução para a maior chave e a menor.

4.4. Arquitetura é essencial

O objetivo deste experimento é tratar da hipótese de que a frequência do processador, isoladamente, não é um bom previsor do desempenho de um dado celular na execução dos algoritmos avaliados. Para tal, os diversos aparelhos serão comparados par a par, para todas as combinações de algoritmo com tamanho de entrada. Assim, pode-se saber quais tiveram um melhor desempenho. Esse desempenho será comparado com o previsto pela frequência do processador. Caso as previsões sejam diferentes podemos então assumir que a frequência do processador não é um bom previsor isoladamente.

A comparação entre o 7373 e o A1200 foi inconclusiva (I.C.: -0.133 / 0.411 com confiança de 40%), apesar de a média (0.139), ter tendido a favorecer o A1200 por ser positiva, com 90% de confiança. O A1200 deveria ter sido significativamente superior ao 7373, devido a diferença entre as frequências de processador.

Já para a comparação entre o 7373 e o N95, o teste também foi inconclusivo (I.C.: -0.429 / 0.211, com confiança de 40%), mas a média (-0.109) favoreceu o 7373 pelo fato de ser negativa, com 90% de confiança. Entretanto, o N95 deveria ter sido melhor que o 7373.

A comparação entre o 7373 e o V3 apresentou o resultado esperado. O 7373 foi significativamente superior ao V3, (I.C.: -1.032 / -0.838, com 90% de confiança). Em relação ao V220, novamente como esperado, o 7373 foi superior (I.C.: 0.842 / 1.035, com confiança de 90%).

O A1200 foi superior ao N95 (I.C: -0.667 / -0.012, com 90% de confiança). Porém, o N95 deveria ter sido superior. Por fim, a comparação entre o V3 e o V220 foi inconclusiva (I.C.: -0.103 / 0.214, com 40% de confiança), mas a média favoreceu o V3, que possui maior frequência.

Como pode ser percebido, a frequência, isoladamente, não pode ser considerada um bom previsor para o desempenho dos celulares. Ou seja, a partir apenas da frequência do processador, não foi possível inferir se um celular apresentará melhor desempenho que outro. Assim, a arquitetura do processador e do celular, como por exemplo, velocidade da memória, também podem interferir no desempenho. Além disso, o estudo apresentou como resultado colateral a classificação de desempenho dos celulares. Para a carga executada (algoritmos de criptografia), o A1200, por exemplo, foi o que apresentou melhor desempenho. Esse resultado pode servir de referência caso se deseje adquirir um celular com o propósito de realizar diversas operações que envolvam criptografia.

4.5. A influência do fabricante

Até o momento percebe-se que, a frequência do processador não é o único fator impactante no desempenho final, como visto na seção anterior. Nessa seção iremos verificar se o fabricante, juntamente com suas tecnologias usualmente utilizadas em seus aparelhos, pode ser um fator impactante no resultado final. Para tentar demonstrar essa hipótese, elaboramos um projeto fatorial 2^3.

Nesse projeto realizamos quatro conjunto de experimentos, que envolveram os algoritmos de destaque das categoria de algoritmos analisadas neste trabalho, RC4, SHA1, DH e DSA, como abordado na seção 4.1. Esse projeto conta com três fatores, fabricante (A), processador (B) e tamanho do arquivo (C) ou tamanho da segurança (C) (dependendo da categoria utilizada). Os fabricantes variados foram Nokia e Motorola. Sabemos que não é possível manipular o processador de um celular, ou seja, não conseguimos trocar o processador de um celular, como fazemos em um computador convencional. Porém, em nossos experimentos, era desejado que isso fosse possível. Na tentativa de tornar o experimento o mais real possível, dentro das limitações impostas, identificamos quatro aparelhos, dois da Motorola (A120 - 320 MHz e V3 - 66,1 MHz) e dois da Nokia (N95 - 332 MHz e 7373 - 72,2 MHz). Esses aparelhos foram agrupados em duas classes de processadores, denominadas como 325 MHz e 68 MHz, representadas respectivamente pelos aparelhos, (A1200 e N95) e (V3 e 7373). Com relação ao tamanho de arquivo, foram variados arquivos de 23 bytes e 50 Kb (para os algoritmos RC4, DSA, SHA1). E com relação ao tamanho da segurança foram utilizadas seguranças de 512bits e 1024 bits, para o algoritmo DH.

Tabela 5. Algoritmo RC4.

	23 bytes (-1)		**50 Kbytes (1)**	
	325 MHz (-1)	**68 MHz (1)**	**325 MHz (-1)**	**68 MHz (1)**
Nokia (1)	0,25 s	0,08 s	0,4 s	0,19 s
Motorola (-1)	0,1 s	0,3 s	0,2 s	13,04 s

No primeiro experimento realizado, relativo ao algoritmo RC4, encontramos que cada fator e suas interações explicavam: A = 13%, B = 14%, C = 13%, AB = 13%, AC = 14%, BC = 13%, ABC = 14%. Podemos observar que o fabricante (A) explicou uma fração significativa do resultado. Mas no resultado final precisamos levar em consideração, a diferença de frequência do processador de cada categoria. Mesmo levando isso em consideração, nota-se que o fabricante ainda é um fator relevante pois, por exemplo, no caso do Nokia 7373 e Motorola V3 a diferença da frequência do processador é aproximadamente 10% favorável ao Nokia 7373. Porém, como observado na tabela 5, para 50 Kb, considerando esses mesmos aparelhos, os tempos coletados variam por um fator 100 vezes maior. Se o desempenho fosse explicado somente pelo processador essa variação deveria ser de aproximadamente 10%.

O segundo experimento, que foi referente ao algoritmo DH, cada fator e suas interações explicaram: A = 13%, B = 14%, C = 13%, AB = 13%, BC = 13%, AC = 13%, ABC = 12%. Pode-se observar que o fabricante é bastante representativo no resultado final.

Mesmo levando em consideração a diferença da frequência dos processadores de aproximadamente 10%, entre os celulares Nokia 7373 e Motorola V3, ainda assim somente o processamento não é o único fator relevante. Como pode ser observado na tabela 6, o Nokia 7373 gastou 1,5s para realizar uma segurança de 512bits a uma informação de 256 bits, já o Motorola V3 gastou 38,08s para realizar a mesma tarefa.

Tabela 6. Algoritmo DH.

	512 bits (-1)		1024 bits (1)	
	325 MHz (-1)	**68 MHz (1)**	**325 MHz (-1)**	**68 MHz (1)**
Nokia (1)	1,4 s	1,5 s	3,09 s	6,09 s
Motorola (-1)	0,8 s	38,08 s	2,6 s	183,3 s

Já o experimento referente do SHA1, os fatores e suas interações explicaram: A = 13%, B = 14%, C = 14%, AB = 14%, AC = 13%, BC = 13%, ABC = 11%. Pode-se observar que o fabricante também tem uma parcela significativa no resultado. A diferença dos resultados obtidos pelos aparelhos 7373 e V3 é bem superior 10%, que é a quantidade da frequência dos processadores que varia. A tabela 7 apresenta os dados desse experimento.

Tabela 7. Algoritmo SHA1.

	23 bytes (-1)		50 Kbytes (1)	
	325 MHz (-1)	**68 MHz (1)**	**325 MHz (-1)**	**68 MHz (1)**
Nokia (1)	0,2 s	0,03 s	0,3 s	0,3 s
Motorola (-1)	0,07 s	0,25 s	0,2 s	11,2 s

No quarto experimento, relativo ao algoritmo DSA, os fatores, juntamente com suas interações, explicaram: A = 32%, B = 35%, C = 0%, AB = 32%, BC = 0%, AC = 0%, ABC = 0%. Observa-se que o tamanho do conteúdo não explica uma parcela significativa, já o fabricante possui alta representatividade. Levando em consideração a frequência do processador o Nokia 7373 era para ser superior ao Motorola V3 aproximadamente 10%, mas para assinar um arquivo 50 Kb o 7373 gastou 24,3s, já o V3 gastou 550,7s (tabela 8). Para assinar o mesmo arquivo o Nokia N95 gastou 7,9s e o Motorola A1200 gastou 7,5, sendo que se fosse levado em consideração somente a frequência do processador, o N95 teria que ser superior ao A1200 em aproximadamente 3%.

Tabela 8. Algoritmo DSA.

	23 bytes (-1)		50 Kbytes (1)	
	325 MHz (-1)	**68 MHz (1)**	**325 MHz (-1)**	**68 MHz (1)**
Nokia (1)	8,2 s	24,6 s	7,9 s	24,3 s
Motorola (-1)	7,5 s	549,9 s	7,5 s	550,7 s

5. Conclusões e Trabalhos Futuros

Não foi encontrado nenhum trabalho com as mesmas características ao desse na literatura, assim acreditamos que esse é um trabalho pioneiro. As análises acerca do desempenho de algoritmos criptográficos, aqui realizadas, se basearam em técnicas de estatística a fim de se obter conclusões relevantes e sólidas a respeito do tema abordado.

Este trabalho mostrou que a escolha do algoritmo tem uma importância crucial sobre o desempenho da criptografia em um dispositivo móvel. Após a realização dos testes foi possível identificar os melhores algoritmos de cada categoria sendo: SHA1, DH, DSA e o RC4.

Conclui-se também que a frequência do processador não é o único fator impactante no desempenho desses algoritmos. As arquiteturas, que estão relacionadas aos fabricantes, também influenciam no resultado final. Nas análises realizadas, com relação aos telefones inteligentes, categoria de dispositivos com frequência de processador acima de 150 MHz, a Motorola apresentou superioridade. Já na categoria Média, a Nokia foi superior. Para afirmar com maior certeza qual fabricante é melhor, são necessários testes com uma variedade maior de aparelhos.

É bastante divulgado na literatura que os algoritmos utilizando curvas elípticas devem ser mais eficientes que seus equivalentes originais. No entanto, nossos experimentos mostraram que os algoritmos utilizando curvas elípticas não tiveram o desempenho esperado, quando comparados com seus correspondentes sem curvas elípticas. Isso precisa ser melhor investigado mas, inicialmente acreditamos que a razão para essa contradição possa estar relacionada a implementação desses algoritmos. Uma vez que algumas operações, como operações com números grandes que são necessárias por exemplo no ECDH, não são fornecidas nativamente pela plataforma J2ME e tiveram que ser implementadas pelos desenvolvedores da biblioteca utilizada neste trabalho.

Novos aparelhos são lançados em um ritmo muito alto, logo, os aparelhos utilizados nesse trabalho tendem a se tornar obsoletos em um médio prazo, mesmo assim, acreditamos que as informações obtidas com as análises aqui realizadas não se tornarão defasadas na mesma velocidade em que os aparelhos celulares evoluem. Apesar disso, um próximo passo desejável seria a realização de um estudo mais genérico através de uma modelagem analítica dos dispositivos, ou seja, considerando as características de processamento, memória e outras.

Como se sabe, uma das principais limitações de dispositivos móveis está ligada ao consumo energético, logo, um outro ponto interessante a ser considerado seria a inclusão de mais um fator no projeto: o consumo de energia para se realizar a criptografia.

Este artigo pode ser usado como base para vários outros trabalhos relacionados ao tema. As informações obtidas nas análises realizadas podem ser úteis, por exemplo, na criação de novos protocolos criptográficos, que serviriam para tornar comunicações criptografadas otimizadas para os dispositivos móveis.

6. Agradecimentos

Gostaríamos de agradecer Everthon Valadão, Keyla Macharet Brasil e Itamar Viana, pela preciosa ajuda na fase de testes do projeto.

Referências

(2007a). Amazon. http://www.amazon.com.

(2007b). Bouncycastle. http://www.bouncycastle.org.

(2007c). Idc - pesquisa mundial do mercado de telefones celulares. http://www.idc.com.

(2007d). Motorola. http://www.motorola.com.

(2007e). Nokia. http://www.nokia.com.

(2007f). Palm. http://www.palm.com.

(2007g). Taste phone. https://tastephone.dev.java.net.

Argyroudis, P. G., Verma, R., Tewari, H., and O'Mahony, D. (2004). Performance analysis of cryptographic protocols on handheld devices. *Network Computing and Applications (NCA)*.

D. Eastlake, r. and Jones, P. (2001). Us secure hash algorithm 1 (sha1).

FIPS (1994). *Digital Signature Standard*. Federal Information Processing Standards Publication 186. U.S. Department of Commerce/N.I.S.T. National Technical Information Service.

Kocher, P., Lee, R., McGraw, G., Raghunathan, A., and Ravi, S. (2004). Security as a new dimension in embedded system design. *Symposium on Design Automation and Microprocessors (DAC)*.

Potlapally, N., Ravi, S., Raghunathan, A., and Jha, N. (2003). Analyzing the energy consumption of security protocols. *International Symposium of Low Power Electronics and Design*.

Potlapally, N., Ravi, S., Raghunathan, A., and Lakshminarayana, G. (2002). Algorithm exploration for efficient public-key security processing on wireless handsets. *DATE Designers Forum*.

Ravi, S., Raghunathan, A., and Potlapally, N. (2002). Securing wireless data: System architecture challenges. *International Symposium System Synthesis*.

Rescorla, E. (1999). Diffie-hellman key agreement method.

Rivest, R. (1992). The md5 message-digest algorithm.

Rivest, R. (1995). The rc5 encryption algorithm. *Dr. Dobb's Journal*, 10:pp. 146–148.

Wong, D., Fuentes, H., and Chan, A. (2001). The performance measurement of cryptographic primitives on palm devices. *Annual Computer Security Applications Conference (ACSAC)*.