TÉCNICAS DE MEDIÇÃO DO CONSUMO DE ENERGIA EM PLATAFORMAS COMPUTACIONAIS

TECHNIQUES FOR MEASURING THE ENERGY CONSUMPTION OF COMPUTER PLATFORMS

Lealdo Santos Neto, Edward David Moreno, Leila B.C. de Matos

Departamento de Computação, Universidade Federal de Sergipe – UFS
lealdo.neto@gmail.com, edwdavid@gmail.com, leila@ifs.edu.br

Resumo

Este artigo apresenta quatro técnicas para estimar e medir o consumo de energia de aplicações executando em plataformas computacionais. As quatro técnicas são: (i) medição do nível de descarga da bateria para um notebook usando comandos do próprio Linux; (ii) estimativa teórica com base nas características do processador, em um PC tipo desktop; (iii) simulação de uma arquitetura usando a ferramenta Sim-Panalyzer e (iv) medição do consumo real usando um osciloscópio. Nos testes foram usados os algoritmos SHA e Blowfish do benchmark Mibench. A técnica que oferece maior precisão é a experimental, pois consegue medir a real corrente drenada pelo sistema enquanto executa uma aplicação.

Palavras-chave: sistemas Embarcados; consumo de Energia; algoritmos criptográficos; avaliação de Desempenho; agoritmos de hashing.

Abstract

This paper presents four techniques to estimate and measure the energy consumption of applications running on computing platforms. The four techniques are: (i) measuring the level of discharge of the battery for a laptop using commands in Linux; (ii) theoretical estimation based on the characteristics of the processor in a PC desktop; (iii) architectural simulation using a simulation tool such as Sim-Panalyzer and (iv) to measure the real consumption using an oscilloscope. We have tests and measurements for crypto graphics algorithms from MiBench benchmark: SHA, Blowfish and AES. The technique that offers increased accuracy is the experimental with oscilloscope, since it can measure the actual current draw by the system while running an application.

Keywords: embedded Systems; Energy consumption; cryptographic algorithms; Pperformance Evaluation; hashing Algorithms.

1 INTRODUÇÃO

Esta é a era onde a informação e o conhecimento desempenham um papel estratégico, e passaram a ser supervalorizados. Neste novo cenário, os dispositivos computacionais são de extrema importância, já que são os principais meios de criação e transmissão das informações.

Estes dispositivos computacionais são em sua maioria sistemas embarcados, que são dispositivos de propósito específicos e devem apresentar desempenho satisfatório com restrições de custo, tamanho e consumo de energia [1].

Como tais dispositivos são usados para a transmissão de informações valiosas, se torna necessário cada vez mais a implementação de algoritmos de segurança para os mesmos. Estes algoritmos podem ser do tipo *hash* ou algoritmos de criptografia, com chaves simétricas ou assimétricas.

Diante do crescente uso dos algoritmos de segurança e da restrição no consumo de energia de alguns dispositivos, neste artigo é estudado e mensurado o desempenho e consumo de energia dos algoritmos SHA, Blowfish e AES presentes no MiBench [5], que é um conjunto de 35 aplicações otimizadas para sistemas embarcados.

O artigo está organizado em quatro seções, a seção 2 apresenta as metodologias usadas no trabalho, a seção 3 é dedicada aos resultados experimentais, na seção 4 temos as conclusões com sugestões de trabalhos futuros.

2 TÉCNICAS USADAS PARA MEDIÇÃO

Nesta seção são explicados os métodos empregados neste trabalho para a estimativa e medição do consumo de energia dos algoritmos SHA e blowfish em diferentes plataformas (notebook, PC desktop e simulação arquitetural de um microprocessador ARM).

As técnicas usadas neste trabalho são: (i) medição do nível de descarga da bateria para um notebook usando comandos do próprio Linux; (ii) estimativa teórica com base nas características do processador, em um PC tipo desktop; (iii) simulação de

uma arquitetura usando a ferramenta Sim-Panalyzer e (iv) medição do consumo real usando um osciloscópio.

2.1. Medição de Descarga de Baterias

Esta técnica de medição é semelhante a aquela usada em [4]. Quando o tempo de execução de uma aplicação é muito rápido não é possível medir a descarga da bateria usando os comandos próprios do Linux e por esse motivo recomenda-se usar um *Shell script* para que o mesmo seja executado um determinado número de vezes e verificar o nível de carga da bateria após esse número de execuções. Assim, o consumo de energia de cada ciclo de execução do algoritmo é calculado usando o consumo de energia total dividido pela quantidade de ciclos que o algoritmo foi executado. Também é calculado o tempo de execução de cada algoritmo dividindo o tempo de processo total pelo número de execuções feita pelo script.

Após a aquisição dos dados são feitos os seguintes cálculos para a obtenção do consumo do dispositivo ao executar o algoritmo:

a) $Tempo\ por\ execução = \frac{Tempo\ de\ execução\ total\ (s)}{Quantidade\ de\ execuções}$;

b) $Consumo\ de\ bateria\ por\ execução = \frac{Consumo\ total\ da\ bateria\ (Ah)}{Quantidade\ de\ execuções}$;

c) $Corrente\ média = \frac{Energia\ consumida\ (Ah)}{Tempo\ de\ execução\ (hora)}$;

d) $Potência\ média\ (W) = Corrente\ média \times Voltagem\ da\ bateria$;

e) $Consumo\ (J) = Potência\ média \times Tempo\ de\ execução\ (s)$.

A aplicação dessa medição foi feita em um notebook Toshiba modelo T135, com processador Intel Pentium SU4100, 3 GB de memória RAM DDR3, HDD de 320 GB e tela de 13,3 polegadas, executando o sistema operacional Linux Ubuntu 9.10.

Os testes foram realizados em grupos de 50, 100, 200, 500 e 1000 execuções para cada arquivo em cada algoritmo, e a medição do descarregamento da bateria é feita utilizando a diferença do nível da carga da bateria, apresentada no arquivo **/proc/acpi/battery/BAT1/state** do sistema operacional Linux, ao iniciar e ao terminar a execução de cada algoritmo. Para a medição do tempo de processo utilizado pela

bateria foi utilizado o comando **"time"**, presente no Linux, ao processar o conjunto de execuções de cada algoritmo. Para a execução desses conjuntos foi escrito um Shell Script para cada um dos algoritmos em que são passados parâmetros como a quantidade de execuções e o arquivo de entrada, e os parâmetros da aplicação.

Visando obter uma estimativa um pouco mais precisa do consumo energético do algoritmo foi medido, utilizando a mesma metodologia, o consumo do dispositivo nas mesmas condições de funcionamento com o processador ocioso, para tal processo foi escrito um *Shell script* que faz com que o processador fique ocioso por determinado tempo. O *script* utilizado segue o padrão:

```
#!/bin/sh
FILENAME=/home/lealdo/Documentos/TCC/Algoritmos/blowfishT/Output/ResBF_$3_$2_$1.txt
if [ $# -ne 3 ]
then
   echo "O Shell Script deve ter como parametro de entrada o número de execuções e o arquivo de
entrada"
  exit 1
fi
getInfo()
{
      echo "Execuction: #"$num >> $FILENAME
echo "Present rate(mW) : " `cat /proc/acpi/battery/BAT1/state | grep "present rate" | awk '{ print $3 }'`
>> $FILENAME
echo "Present Voltage(mV): " `cat /proc/acpi/battery/BAT1/state | grep "present voltage" | awk '{print
$3}'` >> $FILENAME
echo "Remaining capacity(mWh): " `cat /proc/acpi/battery/BAT1/state | grep "remaining" | awk '{print
$3}'` >> $FILENAME
}
getInfo
for num in $(seq $1)
do
  echo "Execuction: #"$num
      ./bf     $3     $2     /home/lealdo/Documentos/TCC/Algoritmos/blowfishT/Output/out_$2_$num.txt
1234567890abcdeffedcba0987654321
done
getInfo
```

Este script escrito possui como parâmetros a quantidade de testes desejados, e os parâmetros do algoritmo. Por exemplo, caso o processo é de encriptação ou decriptação, então o arquivo a ser criptografado e as características da chave e etc. No script é definida a função **getInfo** que captura as informações da bateria. A função básica do script é capturar as informações da bateria, em seguida executar o algoritmo. Por exemplo, para executar o programa Blowfish em encriptação 100 vezes

para o arquivo entrada1 o script é chamado da seguinte forma: **./scriptbf.sh 100 entrada1 E**.

2.2. Estimativa usando as Características do Processador

Para a aplicação desta técnica baseada em uma estimativa do consumo de energia requere-se apenas a quantidade de ciclos de *clock* necessários para a execução do algoritmo e as características do processador (como voltagem, velocidade máxima e corrente máxima) [4, 6]. Após a aquisição destes dados são feitos os seguintes cálculos para a obtenção do consumo do processador para a execução do algoritmo:

a) $Tempo\ por\ execução\ (s) = \frac{Número\ de\ ciclos\ necessários}{Velocidade\ do\ processador\ (\frac{ciclos}{s})}$;

b) $Potência\ do\ processador\ = Voltagem * Corrente\ (A)$;

c) $Corrente\ média = \frac{Energia\ consumida\ (Ah)}{Tempo\ de\ execução\ (hora)}$;

d) $Consumo\ (J) = Potência\ do\ processador\ \times Tempo\ por\ execução$;

Para o cálculo da estimativa usando este método foram usadas as características de m processador Intel Celeron 2.2 GHz, que apresenta a voltagem de 1.1 V e corrente máxima de 35 A.

2.3. Simulação usando o Sim-Panalyzer

O Sim-Panalyzer [2] é um simulador de consumo de energia baseado no simulador de processadores SimpleScalar [3], que simula uma arquitetura computacional completa (com CPU, cache e hierarquia de memória). Com base nesse modelo é possível simular programas reais executando sobre tais plataformas.

Como o resultado é tido com base em uma simulação de arquitetura detalhada, o Sim-Panalyzer consegue modelar de forma detalhada tanto a potência dinâmica quanto a de fuga total do processador alvo, além de fornecer detalhadamente o consumo de cada componente simulado (exemplo: cachê de instruções, cachê de dados, unidades de ponto flutuante, barramentos, unidades de I/O, entre outros), conforme se observa na Tabela 1.

Tabela 1 - Áreas de consumo apresentadas no Sim-Panalyzer

Áreas de Consumo	Significado
aio	Endereço da unidade de I/O
dio	Dados da unidade I/O
irf	Registradores de inteiro
fprf	Registradores de ponto flutuante
il1	cache de instruções nível 1
dl1	cache de dados nível 1
dl2	cache de dados nível 2
itlb	Tabela de instruções do TLB
dtlb	Tabela de dados do TLB
btb	Branch Target Buffer
bimod	Previsor bimodal
ras	Retorno da pilha de endereço
logic	Circuito lógico aleatório
clock	Relógio gerador de clock do sistema
uarch	Microarquitetura, a maneira com que a ISA é implementada em um processador

Os parâmetros para a geração dos perfis de cada componente do computador é dado como entrada para o simulador Sim-Panalyzer que juntamente com o SimpleScalar gera os padrões de consumo de energia. Segundo AUSTIN et al [2], o Sim-Panalyzer foi escrito originalmente com base na arquitetura de instruções (ISA) da família de processadores ARM, obtendo excelentes resultados em simulações deste tipo.

A aplicação do Sim-Panalyzer é feita de forma simples, necessitando apenas executar o simulador passando como parâmetro a arquitetura a ser simulada, juntamente com as características do sistema, e o programa compilado para a arquitetura ARM, juntamente com os parâmetros para a execução do mesmo.

Para se obter o consumo total do algoritmo é necessário apenas somar consumo energético de cada componente simulado pelo Sim-Panalyzer, obtendo

então o consumo geral do dispositivo simulado.

Durante a execução dos testes, usaram-se configurações e parâmetros de uma arquitetura ARM, que é usada e configurada como padrão no Sim-Panalyer.

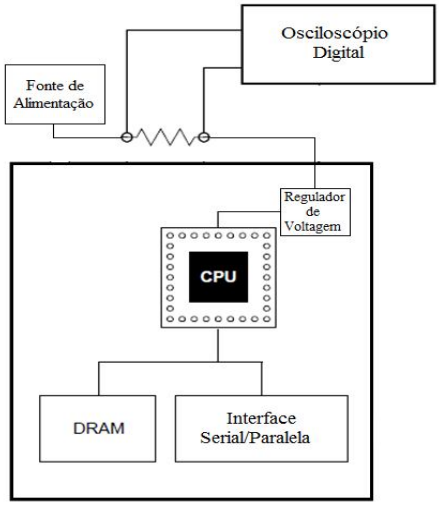
Existem outras ferramentas para realizar a medição, tais como [9]: Netsim, PowerScope, Jouletrack, SNU Energy Explorer WEB; mas a maioria delas não está disponível ou são de uso restrito.

2.4. Medição Real usando um Osciloscópio

A medição real do consumo energético foi aplicada em um desktop com Intel Celeron 2.2 GHz, 512 Mb de memória RAM, HDD de 40 Gb e placa mãe Foxconn 650M02-G-6, e sistema operacional Linux Ubuntu 9.10, usando um osciloscópio da marca Owon modelo PDS5022S.

Para a medição do consumo de energia foi usado o osciloscópio para medir a corrente consumida pelo processador ao executar o algoritmo desejado. Para medir a corrente, foi inserido um resistor de 0.333 Ohm conectado em série com o cabo de alimentação ATX12V [6, 7] da placa mãe e medida a diferença de voltagem da entrada e saída do resistor de *shunt* que foi inserido em série no circuito da placa principal, conforme se observa na Figura 1.

Figura 1 - Esquema utilizado para as medições com o Osciloscópio



Com esses dados pode-se calcular a corrente e potência consumidas pelo processador, da seguinte forma:

a) $Corrente\ (A) = \frac{Variação\ de\ Voltagem\ (V)}{Resistência\ (Ohm)}$;

b) $Potência\ (W) = Corrente(A) \times Voltagem\ de\ saída\ (V)$;

O cálculo do consumo do dispositivo durante a execução do algoritmo pode ser feito ao analisar o gráfico gerado pelo osciloscópio e multiplicar o tempo gasto para cada execução do algoritmo pela potência média usada pelo mesmo, calculada com as variáveis: voltagem e corrente reais consumidas pela plataforma.

3 RESULTADOS DOS TESTES

Os resultados obtidos ao aplicar as técnicas de medição de consumo de energia explicadas na seção 2 são apresentados a seguir, e correspondem à média após ter realizado varias execuções.

Os arquivos de entrada usados como parâmetro foram os arquivos texto entrada1 e entrada2, de tamanhos 100 KB e 500 KB, respectivamente, um arquivo de música Money.mp3 de 4,5 MB e um arquivo de vídeo vídeo.MPG de 46,8 MB. Foram usadas 3 chaves, denominadas chave1, chave 2 e chave3 que representam os

tamanhos 32, 256 e 448 bits, usadas como parâmetros para o algoritmo criptográfico Blowfish.

3.1. Medição de Descarga da Bateria

Os resultados dos testes podem ser observados na Tabela 2, neste caso as medições são obtidas usando a descarga da bateria. É possível observar um consumo médio de 12,2 W de potência ao executar o algoritmo SHA na plataforma alvo. Ao analisar os resultados é observado um consumo por byte de 2,46 µJ para o algoritmo SHA.

Tabela 2 - Resultados do algoritmo SHA

Entrada	Tempo (s)	Potência (W)	Consumo (J)
Entrada1	0,061	11,904	0,724
Entrada2	0,079	11,730	0,930
Money.mp3	0,234	11,216	2,619
Vídeo.MPG	1,375	13,935	19,157

Para o algoritmo Blowfish, o resultado foi uma potência média 14,05 W e um consumo de 1,85 µJ por byte. A Tabela 3 contém os resultados dos testes.

Tabela 3 - Resultados do algoritmo Blowfish

Entrada	Tempo (s)	Potência (W)	Consumo (J)
Entrada1	0,021	11,348	0,482
Entrada2	0,07	13,243	0,933
Money.mp3	0,493	15,866	7,82
Vídeo.MPG	4,886	15,750	76,799

3.2. Estimativa usando as Características do Processador

O cálculo da estimativa de consumo do processador Intel Celeron 2.2 GHz, usando como base sua freqüência de operação e potência máxima estimada pelos fabricantes, que é de 2.2 GHz e 38,5 W, respectivamente.

Ao efetuar os cálculos foi feita a estimativa de consumo mostrada na Tabela 4 para o algoritmo SHA e na Tabela 5 os dados para o Blowfish. A estimativa de consumo de energia por byte ao executar o algoritmo SHA é de 0,322 µJ.

Tabela 4 - Resultados - Estimativa do SHA

Entrada	Ciclos	Tempo (s)	Consumo (J)
Entrada1	2000000	0,000909	0,035
Entrada2	10000000	0,004545	0,175
Money.mp3	80000000	0,036364	1,4
Vídeo.MPG	8,6E+08	0,390909	15,05

Tabela 5 - Resultados - Estimativa do Blowfish

Entrada	Ciclos	Tempo (s)	Consumo (J)
Entrada1	4000000	0,001818	0,07
Entrada2	30000000	0,013636	0,525
Money.mp3	430000000	0,195455	7,525
Vídeo.MPG	4,44E+09	2,018182	77,7

3.3. Simulação Usando o Sim-Panalyer

Os resultados da simulação usando o Sim-Panalyzer são apresentados nas Tabelas 6 e 7, para os algoritmos SHA e Blowfish respectivamente. A potência média do dispositivo simulado é de 4 W, e o consumo por byte é de 1000 µJ para o SHA e 6200 µJ para o Blowfish.

Tabela 6 - Dados simulação do algoritmo SHA

Entrada	Instruções	Tempo (s)	Consumo (J)
Entrada1	4451010	25	102,61
Entrada2	22221949	124	511,87

Tabela 7 - Dados simulação do Blowfish

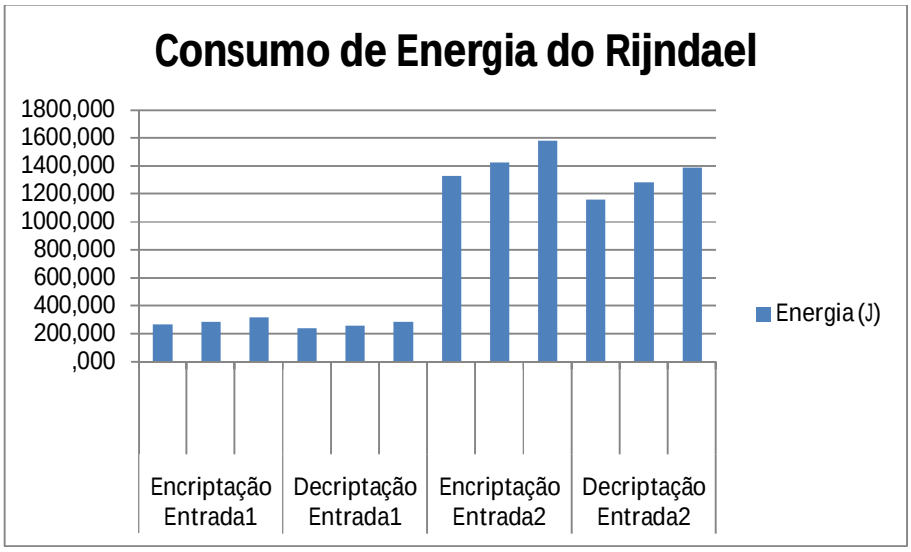
Entrada	Instruções	Tempo (s)	Consumo (J)
Entrada1	17351171	164	644,13
Entrada2	86037267	798	3131,08

Importante destacar que neste caso os dados, tanto do tempo de execução quanto do consumo, são maiores, pois a arquitetura simulada é simples e possui

menos potencia computacional. Lembrar que quando o tempo de execução é maior, então a energia aumenta.

No uso desta metodologia fizemos também testes com o algoritmo criptográfico simétrico AES, melhor conhecido como Rijndael, e foi possível perceber as mudanças no consumo de energia para diferentes tamanhos de arquivos e chave usados. No caso específico usaram-se chaves de 128, 192 e 256 bits e arquivos de entrada de tamanhos diferentes, relacionadas na Figura 2 como chave1, chave 2 e chave 3. Importante observar que para configuração do algoritmo (entrada do arquivo e tamanho da chave) se obtém diferente consumo de energia. Na Figura 2 se observa também que o processo de cifrar demanda mais energia do que o processo de decifrar.

Figura 2 - Gráfico do custo energético para execução do algoritmo rijndael.



3.4. Medição Real usando um Osciloscópio

Para a aplicação deste método foram usados três resistores em paralelo, com 1 ohm de resistência e capacidade de 20w de potência cada, resultando em um resistor equivalente de 0,333 ohms e 60 w de potência. Este conjunto de resistores foi ligado em série no cabo de alimentação da placa mãe (ver Figura 3). Após a montagem foi conectado cada um dos canais do osciloscópio da marca Owon, modelo PDS5022S, nas

extremidades do conjunto de resistores para medir a queda de voltagem provocada pela passagem de corrente.

Figura 3 - Resistores em série com o fio de alimentação da placa mãe

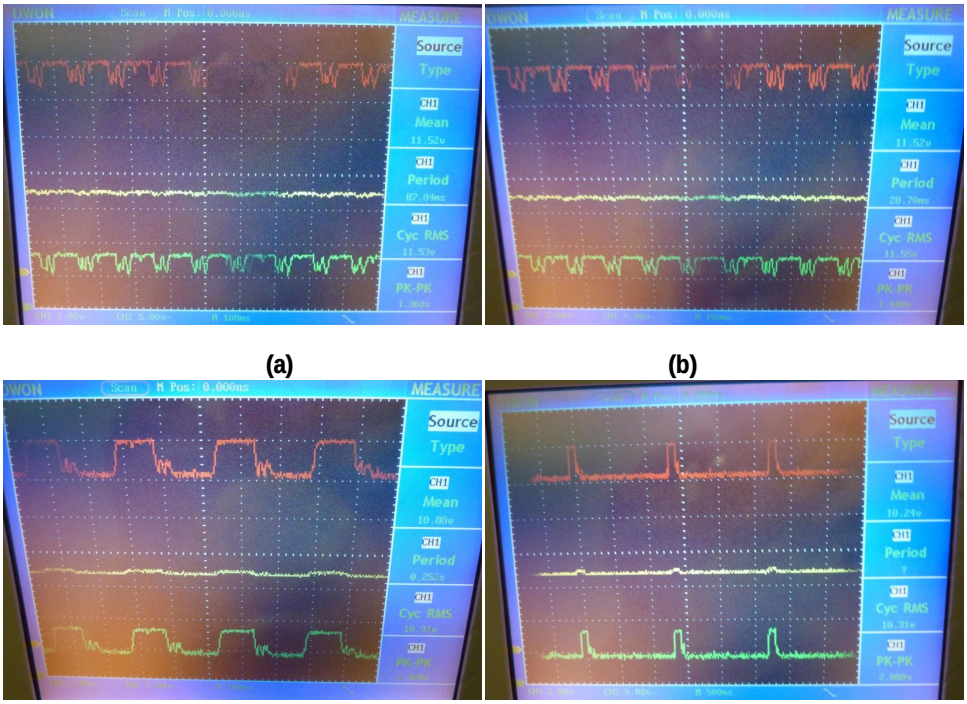
Para efeito comparativo, inicialmente foi medido o consumo do sistema ocioso que apresentava uma queda de 0,5V entre as extremidades do resistor, que significa uma passagem de corrente de 1,5A e uma potência de 17,25W para o sistema ocioso. Com o osciloscópio conectado também foi observado um aumento significativo do consumo de energia com a movimentação do mouse, que resultou em uma queda de 1V entre as extremidades do resistor, que resulta em uma corrente de 3A e uma potência de 33W.

A voltagem medida na entrada do conjunto de resistores com o sistema ocioso foi de 12V, com uma queda média da mesma para 11,68V com o sistema com carga. Estes valores foram usados como parâmetros para o cálculo do consumo de energia nesta metodologia.

No caso do algoritmo SHA foram observados dois diferentes patamares de características, sendo divididas entre as entradas pequenas (entrada1 e entrada2) e as entradas grandes (Money.mp3 e vídeo.MPG). Para as entradas pequenas a potência média usada pelo processador foi de 30,9 W, tendo um aumento para a média de 49,44W ao executarem o mesmo algoritmo para as entradas grandes.

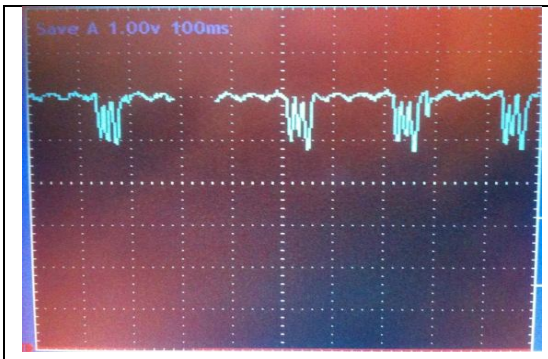
Em testes iniciais foram usados os dois canais do osciloscópio um na entrada e outro na saída da corrente no conjunto de resistores, representados pelo o gráfico amarelo e o vermelho, respectivamente, na Figura 4. O gráfico apresentado em verde é dado pela subtração das voltagens de entrada pela de saída, resultando no gráfico de queda da voltagem. Porém para efeito de cálculo do consumo, o gráfico da subtração é semelhante ao gráfico de saída, já que o gráfico de entrada tem pouca variação durante a execução. Assim foram feitas novas medições usando apenas a voltagem de saída do conjunto de resistores buscando um gráfico mais nítido da queda de voltagem.

Figura 4 - Gráficos da variação da voltagem ao executar o algoritmo SHA-1 para as entradas: (a) entrada1, (b) entrada2, (c) Money.mp3 e (d) vídeo.MPG.



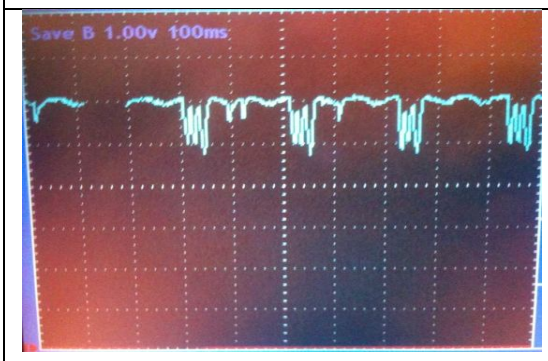
Os gráficos da Figura 5 mostram mais detalhes do consumo de energia para diferentes situações do algoritmo SHA, e as diferentes medidas que se obtém usando osciloscópio, que permite a medição real do consumo.

Figura 5 - Gráficos do Consumo de Energia do Algoritmo SHA



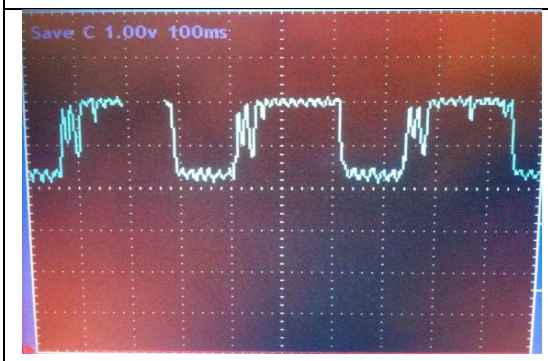
SHA-1

Arquivo	**entrada1**
Tempo entre execuções	0,16 s
Tempo por execução	0,04 s
Variação de Voltagem	0,96 V
Potência Média durante execução	30,90 W
Consumo	1,24 J



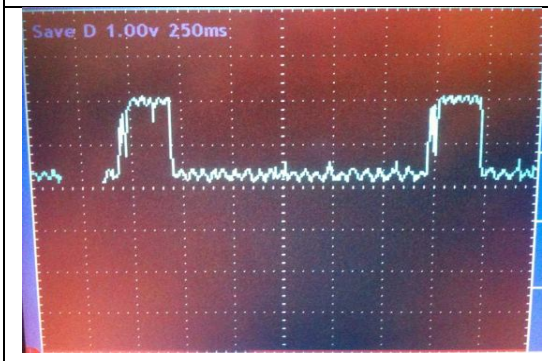
SHA-1

Arquivo	**entrada2**
Tempo entre execuções	0,16 s
Tempo por execução	0,05 s
Variação de Voltagem	0,96 V
Potência Média durante execução	30,90 W
Consumo	1,55 J



SHA-1

Arquivo	**Money.mp3**
Tempo entre execuções	0,2 s
Tempo por execução	0,13 s
Variação de Voltagem	1,6 V
Potência Média durante execução	48,43 W
Consumo	6,30 J

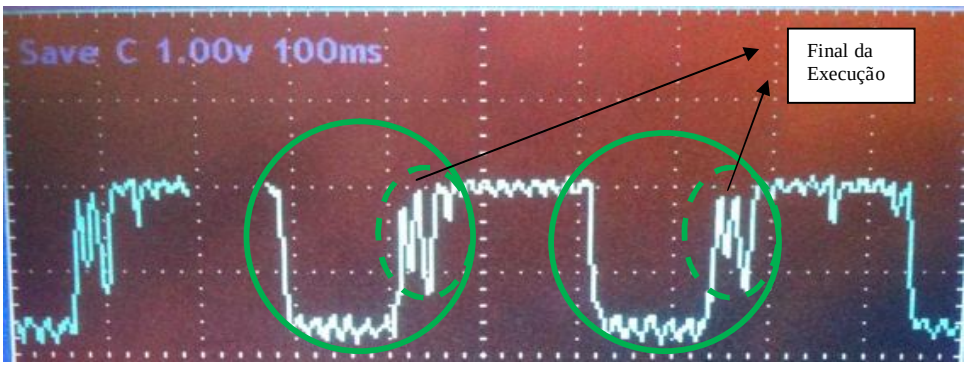


SHA-1

Arquivo	**vídeo.MPG**
Tempo entre execuções	0,25 s
Tempo por execução	1,25 s
Variação de Voltagem	1,68 V
Potência Média durante execução	50,45 W
Consumo	63,06 J

Os resultados da medição real do consumo energético com o uso do osciloscópio são obtidos a partir dos gráficos gerados pelo mesmo, como o mostrado na Figura 6, onde se observam várias execuções, e cada uma delas apresenta o mesmo comportamento no consumo uma vez que se executa o mesmo algoritmo com os mesmos parâmetros. No caso específico da Figura 6, se observa que o programa SHA demanda mais consumo no começo da execução, o qual se mantém quase constante, com variações pequenas, o que corresponde ao cálculo da mudança das constantes A, B, C, D e E do algoritmo, cada uma delas com tamanho de 32 bits, para compor a palavra especial HASH procurado da mensagem de entrada, composta de 160 bits. No final, se faz somente uma atualização do último cálculo, o que demanda menos processamento e, portanto, menor consumo de energia (ver círculo pontilhado, interno ao círculo total de consumo).

Figura 6 - Gráfico gerado para a execução do algoritmo SHA para a entrada Money.mp3



Ao executar o algoritmo SHA (ver resultados na Tabela 8) observou-se uma média de consumo de 30,9 W para as entradas menores e uma média de 49,44 W para as entradas maiores. O consumo médio por byte é de 1,3 µJ para os arquivos Money.mp3 e vídeo.MPG.

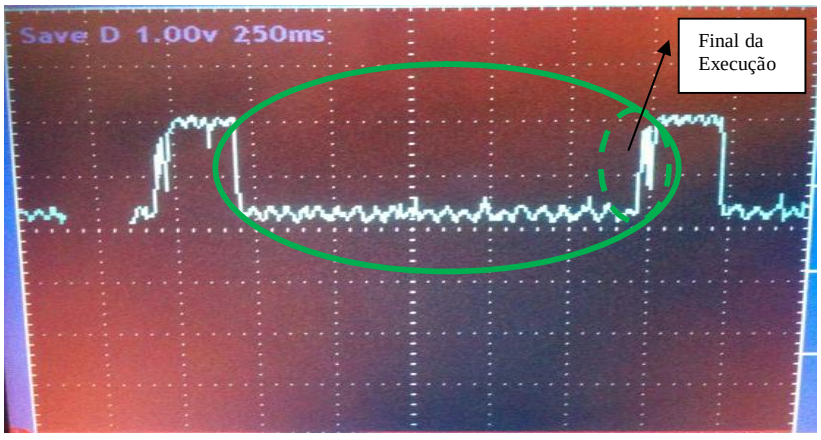
Tabela 8 - Resultados Medição - Algoritmo SHA

Entrada	Tempo (s)	Potência (W)	Consumo (J)
Entrada1	0,04	30,9	1,236
Entrada2	0,05	30,9	1,545
Money.mp3	0,13	48,43	6,296
Vídeo.MPG	1,25	50,45	63,063

A Figura 7 mostra o comportamento do consumo do SHA quando executa um arquivo de vídeo de tamanho 45.6 MB. Neste caso, observa-se um tempo de execução de 1.25 segundos e um tempo entre execuções de 0.25 s. Lembrar que a figura mostra o comportamento da voltagem, que é similar ao de corrente consumida, mas o valor deve ser dividido pelo resistor de shunt usado nos testes. Neste caso a variação de voltagem no resistor de shunt é de 1.68 V, assim a potencia media é de 50,25 W e o consumo médio é 63,06 Joules.

Neste caso é possível observar que o consumo de energia é maior, pois o arquivo de entrada é maior e então há mais processamento. Na Figura 7 se observa um tempo maior da onda de consumo, e na parte final do processamento (ver círculo menor com mensagem FINAL) há menor consumo, a onda de corrente caiu, pois há menos operações para realizar.

Figura 7 - Gráfico gerado para a execução do algoritmo SHA para a entrada Video.mpg



Para os testes com o algoritmo Blowfish foram feitos os mesmos processos utilizados para o algoritmo SHA-1. Para o algoritmo Blowfish foi observado um pequeno aumento da potência média usada pelo processador com o aumento do tamanho da entrada, porém essa diferença não supera 10% do valor médio, que é de 50W de potência.

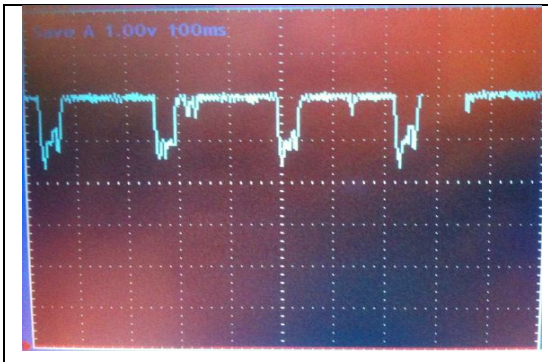
Para o algoritmo Blowfish os resultados médios são apresentados na Tabela 9, onde consta uma média de consumo de 50,15 W e um consumo médio por byte de 1,84 µJ.

Tabela 9 - Dados para o Blowfish

Entrada	Tempo (s)	Potência (W)	Consumo (J)
Entrada1	0,04	45,33	1,813
Entrada2	0,082	49,45	4,038
Money.mp3	0,525	51,45	27,009
Vídeo.MPG	5,133	54,37	279,105

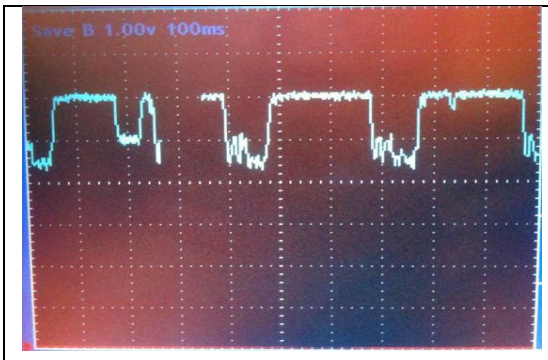
A Figura 8 mostra algumas informações e a forma de onda de corrente para o algoritmo Blowfish, permitindo medir o consumo de energia na execução do algoritmo, quando se encontra no processamento das funções de cifragem. A Figura 9 mostra os detalhes do consumo no processo de decifragem.

Figura 8 - Informações do Consumo de Energia do Blowfish – Cifragem com 32 bits



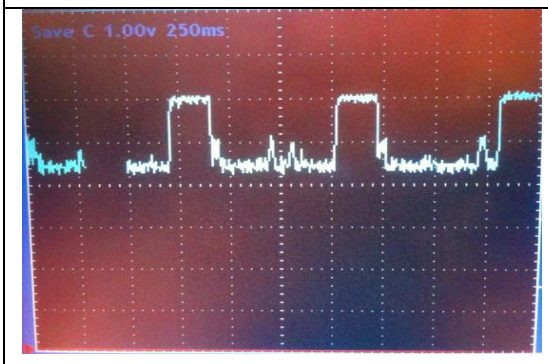
Arquivo	**entrada1**
Tempo entre execuções	0,2 s
Tempo por execução	0,04 s
Variação de Voltagem	1,48 V
Potência Média durante execução	45,33 W
Consumo	1,81 J

Blowfish – Encriptação – Chave 32 bits



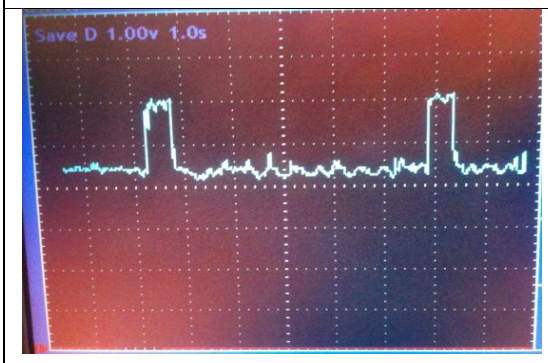
Blowfish – Encriptação – Chave 32 bits

Arquivo	**entrada2**
Tempo entre execuções	0,2 s
Tempo por execução	0,09 s
Variação de Voltagem	1,64 V
Potência Média durante execução	49,45 W
Consumo	4,45 J



Blowfish – Encriptação – Chave 32 bits

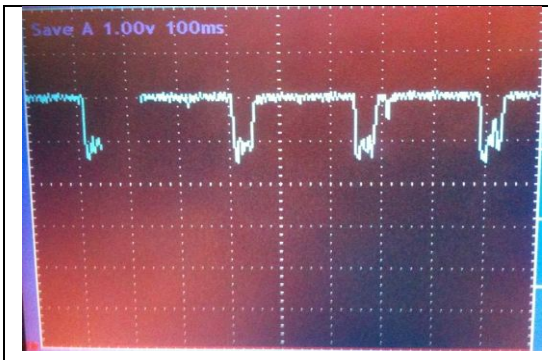
Arquivo	**Money.mp3**
Tempo entre execuções	0,2 s
Tempo por execução	0,6 s
Variação de Voltagem	1,72 V
Potência Média durante execução	51,45 W
Consumo	30,87 J



Blowfish – Encriptação – Chave 32 bits

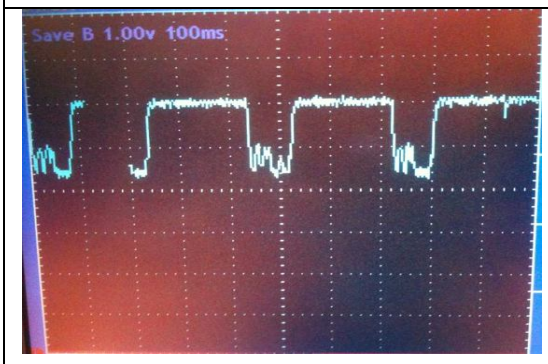
Arquivo	**vídeo.MPG**
Tempo entre execuções	0,5 s
Tempo por execução	5 s
Variação de Voltagem	1,84 V
Potência Média durante execução	54,37 W
Consumo	271,86 J

Figura 9 - Informações do Consumo de Energia do Blowfish – Decifragem com 32 bits



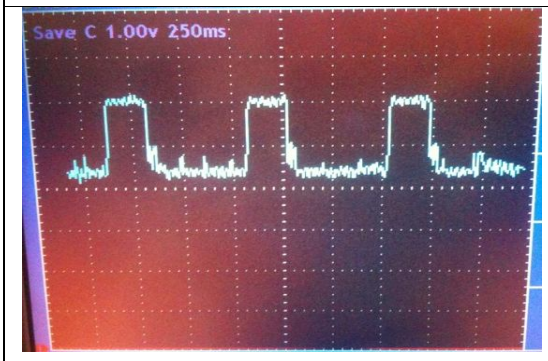
Blowfish – Decriptação – Chave 32 bits

Arquivo	**entrada1**
Tempo entre execuções	0,2 s
Tempo por execução	0,04 s
Variação de Voltagem	1,48 V
Potência Média durante execução	45,33 W
Consumo	1,81 J



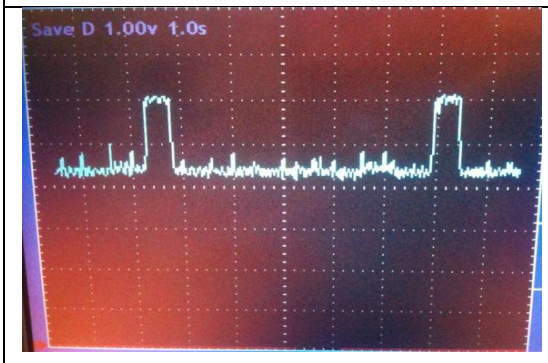
Blowfish – Decriptação – Chave 32 bits

Arquivo	**entrada2**
Tempo entre execuções	0,2 s
Tempo por execução	0,08 s
Variação de Voltagem	1,64 V
Potência Média durante execução	49,45 W
Consumo	3,96 J



Blowfish – Decriptação – Chave 32 bits

Arquivo	**Money.mp3**
Tempo entre execuções	0,2 s
Tempo por execução	0,5 s
Variação de Voltagem	1,72 V
Potência Média durante execução	51,45 W
Consumo	25,72 J



Blowfish – Decriptação – Chave 32 bits

Arquivo	**vídeo.MPG**
Tempo entre execuções	0,5 s
Tempo por execução	5,1 s
Variação de Voltagem	1,84 V
Potência Média durante execução	54,37 W
Consumo	277,29 J

4 CONCLUSÃO E TRABALHOS FUTUROS

Com a finalização do trabalho, foi possível concluir a real importância do consumo energético, visto o grande aumento no uso de dispositivos em tarefas de processamento do dia a dia da sociedade moderna, que cada vez mais usam sistemas embarcados para os mais diversos tipos de processamento. Além do tempo necessário na execução de tais tarefas, o consumo de energia desses dispositivos, em tarefas de processamento e comunicação se torna também relevante.

Para a apresentação deste tralho foram estudados a título de conceituação os sistemas embarcados, conceitos de segurança de dados, o algoritmo MiBench e diversas metodologias e ferramentas para medição do consumo de energia de dispositivos computacionais. Estes estudos resultaram em um embasamento necessário para a aplicação das metodologias e a análise dos resultados.

Nos experimentos iniciais realizados, usando um osciloscópio, foi possível observar que uma simples movimentação do mouse pode causar um aumento de até 90% do consumo em relação ao sistema ocioso [8]. Qualquer tarefa ou processamento exige uma energia para a realização. Em termos de computação, a execução de qualquer instrução demanda uma quantidade de energia específica para essa tarefa e plataforma. Dessa maneira, se exemplifica que o consumo de energia é um fator importante na viabilidade dos projetos de sistemas embarcados, além da consciência ambiental globalmente difundida, que faz com que sejam cada vez mais necessárias medidas de redução do consumo de energia.

Este trabalho teve como objetivo a análise do consumo de energia dos algoritmos SHA e Blowfish, presentes no MiBench em diferentes plataformas computacionais, usando quatro técnicas de medição usadas na literatura.

As metodologias empregadas buscaram abranger diferentes métodos de medição para dispositivos distintos, resultando também em diversos padrões de consumo. Os resultados apresentados pelas diferentes metodologias são apresentados a seguir:

- A primeira metodologia foi aplicada em um *notebook* desenvolvido com tecnologias de baixo consumo, que resultaram no dispositivo com melhor relação entre desempenho e consumo de energia.
- A segunda metodologia, puramente teórica é possivelmente a mais falha, já que os dados de consumo do processador e a quantidade de ciclos necessários para a execução de um algoritmo são calculados com base puramente teórica.
- A terceira metodologia apresenta como resultado da simulação um detalhado relatório do consumo para os parâmetros usados, porém como foi usado o processador com configurações padrão do Sim-Panalyzer a arquitetura simples do mesmo fez com que apesar da menor potência utilizada, apresentasse o pior desempenho entre as metodologias simuladas, já que o tempo de execução foi muito grande.
- A quarta metodologia empregada apresenta uma medição experimental e real do consumo energético da placa mãe contendo um processador de desktop. A medição foi feita usando um osciloscópio. Nesta metodologia são observados os picos de consumo de cada algoritmo e o real tempo de execução dos mesmos. Esta foi a metodologia que teve os resultados mais próximos do consumo real do processador.

Após os testes foi observado que o tamanho da chave e os processos de encriptação (cifrar) e decriptação (decifrar) são diferentes e possuem consumo diferente, mas neste caso não apresentaram variação superior a 10% no consumo de energia para o algoritmo Blowfish.

Também foi observado que as arquiteturas desenvolvidas para baixo consumo apresentam resultados satisfatórios como pode ser observado com a medição da descarga da bateria.

Outra conclusão importante foi observada com o resultado da simulação por meio do simulador Sim-Panalyzer, onde o dispositivo simulado apresenta uma arquitetura muito simples e mesmo apresentando a menor potência dentre os

dispositivos testados, foi o que apresentou o maior consumo de energia, pois o tempo de execução foi o mais elevado.

Ao medir o consumo com o uso do osciloscópio se torna possível observar picos de consumo dentro do algoritmo, e daí a idéia de no futuro analisar e encontrar os trechos de código responsáveis por esse comportamento.

Outros trabalhos futuros sugeridos são aplicar as todas as metodologias usando o mesmo sistema computacional, para assim estabelecer a correlação entre o método e os resultados obtidos. Além disso, aplicar as metodologias empregadas neste artigo para todos os algoritmos do MiBench e outros benchmarks.

REFERÊNCIAS

[1] SANGIOVANNI-VINCENTELLI, A. L. ; MARTIN G., Platform-based design and software design methodology for embedded systems, **IEEE Design & Test of Computers**, v. 18, no. 6, p. 23-33, Nov. 2001.

[2] AUSTIN, T., MUDGE, T., GRUNWALD, D. **Sim-Panalyzer**. Disponível em: http://www.eecs.umich.edu/~panalyzer . Acesso: em jul. 2010.

[3] AUSTIN, T. et al. SimpleScalar: An Infrastructure for Computer System Modeling. **IEEE Computer**, v. 35, p. 59-67, Fevereiro de 2002.

[4] COSTA, Ricardo A. G. **Desempenho e Consumo de Energia de Algoritmos Criptográficos do MiBench em Sistemas Móveis**. UEA - Amazonas. Nov. 2007.

[5] GUTHAUS, M. et al. MiBench: A free, commercially representative embedded suite.. In: WWC-4. IEEE INTERNATIONAL WORKSHOP ON, 2001. **Anais...** 2011**.**

[6] LIN, C. et al. **Energy Efficiency Measurement for Multimedia Audio Decoding on Embedded Systems**. Tunghai University, 2006.

[7] LEE, I. et al. Web-Based Energy exploration tool for embedded systems. **IEEE Design & Test of Computers**, 2004.

[8] NETO, Lealdo Santos. **Estudo e Medição do Consumo de Energia de Aplicações Embarcadas.** 2010. 90 f. Monografia (Trabalho de Conclusão de Curso) - Departamento de Computação, Universidade Federal de Sergipe, Sergipe, 2010.

[9] NETO, Lealdo Santos; Marco T Chella; Edward D. Moreno. Estudo e Medição do Consumo de Energia de Algoritmos Criptográficos do MiBench. In: WORKSHOP DE INICIAÇÃO CIENTÍFICA DO WSCAD, 2010. Petrópolis**. Anais...** Petrópolis: Brasil, p. 28-32, 2010.