UNIVERSIDADE FEDERAL DE PELOTAS
CDTec
Centro de Desenvolvimento Tecnológico

Dissertação

FLUXO DE ATAQUE DPA/DEMA BASEADO NA ENERGIA DOS TRAÇOS PARA NEUTRALIZAR CONTRAMEDIDAS POR DESALINHAMENTO TEMPORAL EM CRIPTOSISTEMAS

R o d r i g o N u e v o L e l l i s

Pelotas, 2017

RODRIGO NUEVO LELLIS

FLUXO DE ATAQUE DPA/DEMA BASEADO NA ENERGIA DOS TRAÇOS PARA NEUTRALIZAR CONTRAMEDIDAS POR DESALINHAMENTO TEMPORAL EM CRIPTOSISTEMAS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação Computação da Universidade Federal de Pelotas, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Rafael Iankowski Soares

Pelotas, 2017.

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

L542f Lellis, Rodrigo Nuevo

Fluxo de ataque DPA/DEMA baseado na energia dos traços para neutralizar contramedidas por desalinhamento temporal em criptosistemas / Rodrigo Nuevo Lellis ; Rafael Iankowski Soares, orientador. — Pelotas, 2017.

96 f. : il.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2017.

1. Ataques a canais laterais. 2. Potência. 3. DPA. 4. DEMA . 5. Cripto. I. Soares, Rafael Iankowski, orient. II. Título.

CDD : 005

Elaborada por Aline Herbstrith Batista CRB: 10/1737

Banca examinadora:

_____

Prof. Dr. Adão Antônio de Souza Jr.

_____

Prof. Dr. Júlio Carlos Balzano de Mattos

_____

Prof. Dr. Leomar Soares da Rosa Jr.

RESUMO

LELLIS, Rodrigo Nuevo. **FLUXO DE ATAQUE DPA/DEMA BASEADO NA ENERGIA DOS TRAÇOS PARA NEUTRALIZAR CONTRAMEDIDAS POR DESALINHAMENTO TEMPORAL EM CRIPTOSISTEMAS.** 2017. 96f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal de Pelotas, Pelotas.

Nas últimas décadas uma das grandes preocupações de projetistas de *hardware* dedicado a aplicações que exigem segurança e sigilo de informações tais como *smart cards* são os ataques a canais laterais (em inglês *Side Channel Attacks* – SCAs). Estes ataques permitem relacionar os dados processados em dispositivos eletrônicos com grandezas físicas tais como a potência, a emissão de radiação eletromagnética ou o tempo de processamento. Isto se torna crítico quando, por exemplo, algoritmos criptográficos são executados e a chave criptográfica pode ser revelada pelo ataque. Dentre estes ataques, os baseados nos traços de potência, conhecidos como ataque por Análise Diferencial de Potência (em inglês *Differential Power Analysis* – DPA) e na emissão de radiação eletromagnética, denominados de Análise Diferencial Eletromagnética (em inglês *Differential Electromagnetic Analysis* - DEMA) são os mais populares, e por não serem invasivos, serem eficientes e não deixarem rastros no dispositivo atacado. Por outro lado, estes ataques exigem que a aquisição dos traços de potência ou radiação eletromagnética, sejam alinhados no tempo a fim de comparar e avaliar estatisticamente as amostras relativas a execução de operações com diferentes dados. Na literatura, existem diversas contramedidas visando evitar a ação destes ataques através da inserção de aleatoriedade de execução de operações, seja através da adição de atrasos aleatórios até a execução com diferentes frequências de relógio. Da mesma forma, existem propostas de estratégias baseadas em processamento de sinais aplicadas aos traços a fim de extrair informações vazadas pela arquitetura, métodos como correlação de fase (em inglês, *Phase Only Correlation - POC*), deformação dinâmica de tempo (do inglês, *Dynamic Time Warping* - DTW) e filtros digitais são usados em fluxos de ataques para estabelecer o realinhamento de traços antes da realização de ataques. Apesar disso, estes métodos são restritos a traços processados com sinal de relógio de mesma frequência ou com pequenas variações, o que por consequência exigem um grande número de traços e seus agrupamentos por frequência de operação. Este trabalho propõe um fluxo de ataque baseado no cálculo da energia dos traços a fim de permitir o realinhamento dos traços independentemente da frequência de operação e assim potencializar a ação dos ataques DPA em arquiteturas protegidas por contramedidas com inserção de aleatoriedade no processamento. Os resultados obtidos destacam que os ataques DPA são mais efetivos quando o cálculo da energia ocorre com segmentos de tamanho aproximado a metade do ciclo médio das frequências de operação dos traços atacados. Em comparação com trabalhos anteriores, o fluxo permite uma redução, no melhor caso, de aproximadamente 93% traços para um ataque bem-sucedido, motivando o uso do fluxo proposto.

Palavras-chave: Ataques a canais laterais, Potência, DPA, DEMA, criptografia.

ABSTRACT

LELLIS, Rodrigo Nuevo. **FLUXO DE ATAQUE DPA/DEMA BASEADO NA ENERGIA DOS TRAÇOS PARA NEUTRALIZAR CONTRAMEDIDAS POR DESALINHAMENTO TEMPORAL EM CRIPTOSISTEMAS.** 2017. 96f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal de Pelotas, Pelotas.

In recent decades one of the major concerns of hardware designers dedicated to applications requiring security and secrecy of information such as smart cards are Side Channel Attacks (SCAs). These attacks allow you to relate processed data to electronic devices with physical quantities such as power consumption, electromagnetic radiation emission or processing time. This becomes critical when, for example, cryptographic algorithms are executed and the cryptographic key can be revealed by the attack. Among these attacks, by power consumption and emission of electromagnetic radiation are the most popular, known as Differential Power Analysis (DPA) and Differential Electromagnetic Analysis (DEMA). Since they are not invasive, efficient and leave no traces on the attacked device. These attacks require that the acquisition of traces of power consumption or electromagnetic radiation relating to the execution of cryptographic algorithms be time aligned in order to statistically compare and evaluate consumption or radiation samples for the execution of operations with different data. In the literature there are several countermeasures of these attacks through the randomization of execution operations either by adding random delays to the by changing clock frequencies. Similarly, there are proposals for strategies based on signal processing applied to the traces in order to extract information leaked by the architecture. Methods such as phase correlation (POC), dynamic time warping (DTW) and digital filters are used to realign traces before attacks. Nevertheless, these methods are restricted to traces processed with clock signal of the same frequency or with small variations, and require a large number of traces or their clustering frequency. This work proposes an attack flow based on the calculation of the trace energy in order to allow the realignment independently of the frequency of operation and thus enable the action of the DPA attacks in architectures with countermeasures based on processing randomization. Results show that DPA attacks are more effective when the energy is calculated in segments of approximately half the average cycle of the frequencies of operation of the traces attacked. Compared to previous works, the flow allows a reduction, in the best case, of approximately 93% traces for a successful attack, motivating the use of the proposed flow.

Palavras-chave: Side Channel Attacks, Power Consumption, DPA, DEMA, cryptography.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AES	*Advanced Encryption Standard*
AOC	*Amplitude Only Correlation*
CMOS	*Complementary Metal-Oxide Semiconductor*
DEMA	*Differential Electromagnetic Analysis*
DES	*Data Encryption Standard*
DPA	*Differential Power Analysis*
DTW	*Dynamic Time Warping*
DTFS	*Discrete Time Fourier Series*
DTFT	*Discrete Time Fourier Transform*
DVFS	*Dynamic Voltage and Frequency Switching*
EBS	*Energy Based Signal*
FFT	*Fast Fourier Transform*
FIR	*Finite Impulse Response*
FPGA	*Field Programmable Gate Array*
GALS	*Global Asynchronous Local Synchronous*
NCAF	*National Cybersecurity Agency of France*
NIST	*National Institute of Standards and Technology*
POC	*Phase-Only Correlation*
RDI	*Random Delay Insertion*
RMS	*Root Mean Square*
SBOX	*Substitution Box*
SCA	*Side Channel Attack*
SEMA	*Simple Electromagnetic Analysis*
SNR	*Signal to Noise Ratio*
SoC	*System on Chip*
SPA	*Simple Power Analysis*
SW-DPA	*Sliding Window Differential Power Analysis*
TA	*Timing Attack*
T-POC	*Threshold Phase Only Correlation*
TTL	*Transistor-Transistor Logic*

SUMÁRIO

1 INTRODUÇÃO

Para que informações sigilosas possam ser trocadas entre dois dispositivos interligados por uma rede de comunicação pública são utilizados protocolos e algoritmos criptográficos. Como exemplos de sistemas que utilizam tais recursos, temos os sistemas *online* para compras através da Internet e sistemas bancários, que implementam protocolos de segurança para troca de dados confidenciais. Os algoritmos criptográficos consistem de mecanismos para alterar o conteúdo da mensagem original a ser transmitida, também conhecida como texto claro, de maneira que essa somente possa ser interpretada, conhecendo-se uma palavra secreta chamada de chave criptográfica. Como os algoritmos de criptografia são públicos, o segredo da encriptação da mensagem fica condicionada a chave criptográfica utilizada que deve ser conhecida apenas pelos entes comunicantes. O texto claro alterado pelo algoritmo de criptografia é dito texto cifrado e pode ser transmitido com segurança.

Por outro lado, existem técnicas de criptoanálise, ciência que visam descobrir ou violar os dados criptografados explorando vulnerabilidades dos algoritmos criptográficos. Ao longo das últimas décadas, os algoritmos criptográficos tiveram uma grande evolução, aumentando suas complexidades com a finalidade de resistir às técnicas de criptoanálise. As técnicas de criptoanálise podem ser divididas em dois grandes grupos, segundo o nível de abstração em que é realizado o ataque. No primeiro grupo estão os ataques lógicos que exploram as vulnerabilidades matemáticas dos algoritmos de criptografia, analisando as relações existentes entre os textos claros e os textos cifrados para gerar expressões matemáticas capazes de predizer os bits da chave criptográfica. No segundo grupo encontram-se técnicas que investigam vulnerabilidades existentes em grandezas físicas dos dispositivos que executam os algoritmos criptográficos, como por exemplo, o tempo de execução, a potência, a emissão eletromagnética, etc. Os ataques desse tipo são chamados de ataques a canais laterais ou ocultos (do inglês *Side Channel Attacks* – SCAs).

O primeiro exemplo de SCA foi apresentado à comunidade acadêmica por Kocher (KOCHER, P. C., 1996), onde provou ser possível relacionar a chave criptográfica com o tempo de execução do algoritmo, ataque conhecido como ataque por análise de tempo (em inglês *Timing Attack* – TA). A partir de então, uma intensa área de pesquisa se desenvolveu com interesses acadêmicos e industriais levando

ao surgimento de outras formas de ataque: ataques por sondagem, ataques por indução a falhas, ataques por análise de potência ou da emissão de radiação eletromagnética.

Kocher (KOCHER, P. C.; JAFFE, J.; JUN, B., 1999) demonstra que existe uma relação entre a potência consumida por um circuito digital e os dados que estão sendo processados pelo mesmo. É possível se ver que diferentes operações aritméticas possuem diferentes traços de potência. Essa análise foi denominada pelos autores de Análise Simples de Potência (do inglês, *Simple Power Analysis* – SPA). Além disso, os autores demonstram que utilizando métodos estatísticos, pode-se estabelecer uma correlação entre os dados processados em um sistema de criptografia e o seu consumo. Esta análise foi denominada de análise diferencial de potência (do inglês, *Differential Power Analysis* – DPA). Há ainda, a análise diferencial eletromagnética (do inglês *Differential Electromagnectic Analysis* – DEMA), a qual, procede do mesmo modo que DPA, utilizando o traço de radiação eletromagnética emitida pelo dispositivo criptográfico em funcionamento.

Diversas propostas são encontradas na literatura para reduzir a ação destes ataques, conhecidas como contramedidas. Estas contramedidas podem ser divididas em três categorias segundo a estratégia adotada para evitar o ataque. A primeira consiste em introduzir ruído nas medições de potência e será a estratégia abordada com mais profundidade neste trabalho; a segunda consiste em impedir análises de correlação mascarando os dados processados e a terceira tenta obter um consumo uniforme para qualquer sequência de dados de entrada. Contudo, existem estratégias de ataques que visam anular os efeitos das contramedidas, tornando vulneráveis até mesmo os sistemas criptográficos dotados de tais proteções.

1.1 Motivação

Com os avanços da microeletrônica, tornou-se possível o projeto de sistemas integrados em um único chip (do inglês, *Systems-on-Chip* - SoCs), sistemas que integram diversos núcleos de propriedade intelectual para o processamento de aplicações de propósito geral ou específicas tais como algoritmos de criptografia, a fim de proteger o sigilo de informações. Como exemplo é possível citar os *Smart Cards*, os *SIM Cards*, etc.

Para garantir um requisito mínimo de segurança e proteção à fuga de informações, estes dispositivos passam por rigorosos testes para avaliar vulnerabilidades segundo diferentes maneiras de tentar-se violar o sigilo das informações processadas. Critérios comuns de segurança foram estabelecidos pela NCAF (*National Cybersecurity Agency of France*) na França e pelo NIST (do inglês, *National Institute of Standards and Technology*) nos E.U.A. para serem avaliados em todos os dispositivos eletrônicos, tais como os smart cards, antes de serem disponibilizados ao mercado consumidor. A imunidade aos ataques DPA e DEMA é um critério de segurança avaliado nestes dispositivos. Assim, é notória a importância do desenvolvimento de sistemas que sejam imunes a estes ataques a canais laterais.

SCA exigem que a aquisição de todos os traços de potência ou radiação eletromagnéticas durante a execução dos dispositivos sejam alinhadas no tempo. Isto significa que a eficiência dos ataques é condicionada a este alinhamento para que análises sejam feitas entre amostras referentes a execução das mesmas operações, porém com dados diferentes e assim permitir que o ataque estabeleça uma relação de dependência entre dados e a potência ou radiação eletromagnética.

Uma das estratégias adotadas para proteger sistemas criptográficos de tais ataques, consiste em criar modos de desvincular a relação existente entre as operações e dados processados com a potência dos dispositivos. Como exemplo é possível citar a proposta de Bucci (BUCCI, M.; LUZZI, R.; GUGIELMO, M.; TRIFILETTI, A., 2005) que propõem a inserção de atrasos aleatórios no caminho de dados da arquitetura do sistema, causando o desalinhamento dos traços. Outra contramedida proposta por Soares (SOARES, R. I., 2010), combina o estilo de projeto Globalmente Assíncrono e Localmente Síncrono (GALS) permitindo ilhas de processamento com diferentes domínios de relógio com o uso de pipeline como forma de inserir aleatoriedade e ruído durante a execução do sistema.

Os praticantes de criptoanálise observando as estratégias de contramedida em sistemas criptográficos propõem a adição de uma etapa de pré-processamento dos traços antes de se realizar o ataque propriamente dito. Esta etapa tem como objetivo realinhar os traços e eliminar os ruídos inseridos por contramedidas.

Neste sentido, Réal (RÉAL, D.; CANOVAS, C., 2008) e Tian (TIAN, Q.; HUSS, S. A., 2012) observaram que variações na frequência de relógio durante o processamento causam, além do desalinhamento temporal uma variação na amplitude do traço. A partir dessa constatação, os autores propõem algoritmos para

alinhamento de traços levando em consideração mudanças na amplitude do sinal. Estas propostas mostram-se sensíveis a ruídos existente nos traços, tornando sua aplicação complexa. Estes trabalhos motivaram a proposição de soluções visando o realinhamento de traços segundo variações de amplitude no presente trabalho, processo aqui denominado alinhamento vertical.

Loder (LODER, L. L., 2014) propõe um fluxo de ataque dedicado ao alinhamento de traços empregando técnicas como correlação de fase (em inglês, *Phase Only Correlation* - POC), DTW (do inglês, *Dynamic Timing Warping*), filtragem e subamostragem. Embora o fluxo de ataque mostre-se efetivo ao realinhamento de traços, a solução demanda frequências de operação com pequenas variações, exigindo uma classificação e agrupamento de traços por frequência para que as ferramentas realizem ataques bem sucedidos.

Alternativamente existe uma proposta de alinhamento de traços baseada na obtenção da energia do sinal. Le (LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J., 2007) propuseram o realinhamento do traço através do cálculo da energia em segmentos de tamanho superior ao atraso inserido pela contramedida. O método mostra bons resultados comparado a estratégias relacionadas, porém restringe-se ao realinhamento de pequenos atrasos e tão pouco é discutido o tamanho do segmento com relação a taxa de sucesso do ataque.

Neste cenário, este trabalho tem por motivação o desenvolvimento e a utilização de técnicas de processamento de sinais, como etapa de pré-processamento ao ataque DPA/DEMA, a fim de remover os desalinhamentos causados por contramedidas que inserem aleatoriedade e ruído sem a necessidade de classificação dos traços por frequência de operação. Isto permite verificar vulnerabilidades de sistemas criptográficos protegidos por tais contramedidas, frente a essas etapas adicionais de pré-processamento de sinais.

1.2 Objetivos

O presente trabalho tem como objetivo propor um fluxo de ataque DPA/DEMA baseado no cálculo da energia do sinal, capaz de realinhar traços alterados pelo processamento com sinal de relógio de frequência aleatória e deslocamentos no domínio do tempo provocados por contramedidas. O fluxo proposto contribui com a

combinação de três etapas de pré-processamento, sendo elas: extração, subamostragem e cálculo da energia dos traços.

Assim, nesta dissertação busca-se investigar técnicas de extração da assinatura dos traços, alvo dos ataques DPA/DEMA. Também, tem-se por objetivo avaliar o processo de subamostragem como pré-alinhamento horizontal dos traços (domínio do tempo). Além disso, o trabalho propõe uma avaliação do tamanho do segmento para cálculo da energia mostrando seu impacto no desempenho dos ataques DEMA. Ainda como uma etapa adicional, realizar experimentos com relação ao alinhamento dos traços no eixo vertical, ou seja, em relação a amplitude.

Como estudo de caso são utilizadas as arquiteturas GALS *pipeline* que possuem configurações que apresentam as contramedidas de inserção de atrasos aleatórios e variação na frequência de relógio. Com isto, pretende-se testar a robustez de tais sistemas ou contramedidas e verificar suas vulnerabilidades.

1.3 Organização do Trabalho

Este trabalho está dividido em sete capítulos. No Capítulo 2 é apresentado com mais detalhes o algoritmo criptográfico DES e o funcionamento do ataque DPA/DEMA. É apresentada, ainda, uma revisão sobre as contramedidas encontradas na literatura para os ataques por DPA/DEMA, destacando-se as arquiteturas GALS *pipeline*. O Capítulo 3 traz uma revisão das estratégias de pré-processamento encontradas na literatura. No Capítulo 4 é feita uma análise dos trabalhos relacionados, que utilizam técnicas de pré-processamento nos traços de potência. Já o Capítulo 5 faz a descrição das etapas do fluxo de ataques, e o Capítulo 6 apresenta os experimentos realizados durante o presente trabalho, e os resultados obtidos, juntamente com as discussões acerca destes. Por fim, no Capítulo 7 são apresentadas as conclusões deste trabalho e também sugeridas propostas para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este Capítulo apresenta uma revisão do referencial teórico necessário para compreender o contexto do problema abordado neste trabalho. Uma revisão detalhada do algoritmo criptográfico DES alvo dos ataques é inicialmente mostrado. Além disso, uma visão geral dos ataques a canais laterais bem como a análise diferencial da potência (DPA) e analise diferencial eletromagnética (DEMA) são abordadas. Do mesmo modo, uma revisão sobre propostas de contramedidas a estes ataques encontradas na literatura.

2.1 O Algoritmo Criptográfico DES

A IBM propôs o algoritmo DES (do inglês, *Data Encryption Standard*) que atendia os requisitos de projetos necessários à solicitação do NBS (*National Bureau of Standards*). Este algoritmo foi adotado como padrão de criptografia pelo NIST (em inglês, *National Institute of Standards and Technology*) em 1977.

O DES trabalha com blocos de dados de 64 bits de texto claro, utilizando uma chave criptográfica de 56 bits para realizar a encriptação dos dados, obtendo-se o texto cifrado. A chave criptográfica deste algoritmo é dita *privada*, ou seja, o DES é um algoritmo *simétrico*. Isso significa que a mesma chave é utilizada tanto na encriptação quanto na decriptação dos dados. Além disso, somente os entes envolvidos na comunicação devem conhecer a chave, sob pena de interceptação dos dados por terceiros.

O funcionamento do algoritmo consiste nas seguintes etapas: permutação inicial; divisão do bloco de dados em duas partes de 32 bits; 16 rodadas que consistem de: funções de expansão, função XOR com a chave, funções de substituição chamadas de SBOX e permutação do bloco de 32 bits; e permutação inversa ou permutação final.

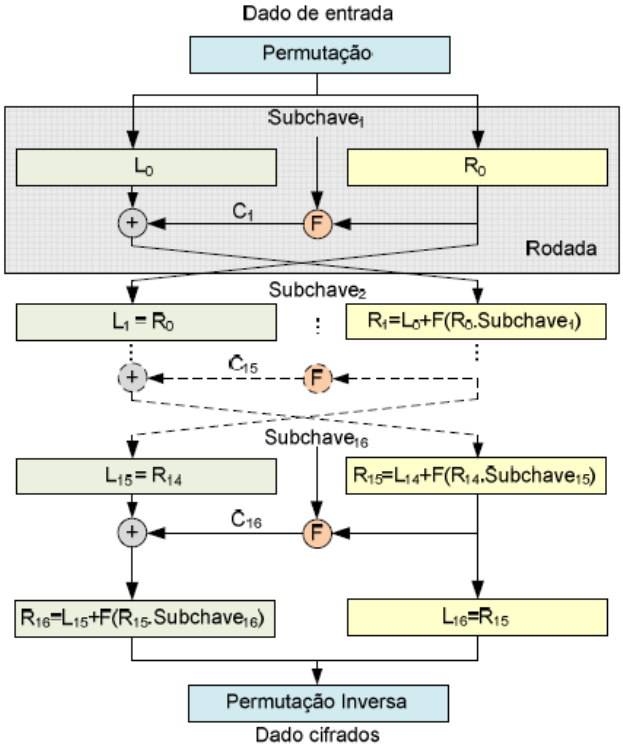

Figura 1 – Estrutura do Algoritmo DES.
Fonte: (SOARES, R. I., 2010)

Primeiramente, o bloco de 64 bits dos dados de entrada sofre uma permutação inicial, ou seja, seus bits têm as posições trocadas em relação ao arranjo inicial. Como exemplo, consideremos a seguinte matriz de permutações:

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Figura 2 – Matriz de Permutação.
Fonte: (NIST, 1999)

A matriz de permutação é percorrida da esquerda para a direita e de cima para baixo, o que significa que o primeiro bit do bloco de 64 bits vai para a 58ª posição, o segundo para a 50ª, o terceiro para a 42ª posição e assim por diante.

No final da execução do algoritmo, como mencionado anteriormente, há uma permutação final, também chamada de permutação inversa, colocando os bits

novamente nas suas posições originais. Então para o nosso exemplo, deve seguir a matriz abaixo:

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Figura 3 – Matriz de Permutação Inversa.
Fonte: (NIST, 1999)

Como vimos, o bit da 58ª posição nesta matriz estava originalmente na 1ª posição. Assim, se percorrermos a matriz de permutação da Figura 3, veremos que na sua 58ª posição consta o valor 1, ou seja, o bit que está na 58ª posição (trocado de posição pela permutação inicial) deve voltar para a 1ª posição, sua posição original. O mesmo acontece com o valor encontrado na 50ª posição desta matriz. É o valor 2, indicando que o bit que está na 50ª deve agora, ocupar a 2ª posição, ou seja, voltando a sua posição original.

As 15 rodadas seguintes executam as mesmas operações na mesma ordem. A Figura 4, simplifica o entendimento da Função F executada em cada uma das rodadas.

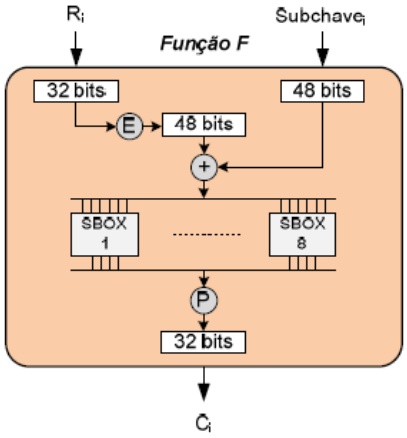

Figura 4 – Diagrama interno da Função F.
Fonte: (SOARES, R. I., 2010)

O bloco de 64 bits de dados é dividido em duas partes de 32 bits: a parte esquerda (do inglês, *Left* – L) e a parte da direita (em inglês, *Right* – R).

Como podemos ver na Figura 4, a parte da direita dos dados (R_i) passa por uma função, chamada *Função F*; que consiste numa etapa inicial de permutação e expansão, que como na permutação anterior, troca a posição dos bits, repetindo alguns deles de maneira que essa parte passe de 32 para 48 bits. Essa operação é necessária para que posteriormente possa ser realizada a função XOR desta parte com uma parte chave criptográfica (K_i). Porém, como a chave possui 56 bits, esta também deve passar por uma transformação de maneira que também fique com 48 bits. Depois de realizada a operação XOR entre R_i e a chave K_i, o resultado dessa operação é submetido às 8 caixas de substituição, as chamadas SBOX. Cada SBOX possui 6 bits de entrada e 4 bits de saída, totalizando novamente o tamanho 32 bits da parte direita (R_i) dos dados originais.

As SBOX funcionam da seguinte maneira: suponhamos que na SBOX1, ou S1, temos os seguintes bits na entrada "110011_2" e que essa SBOX é regida pela matriz mostrada na Figura 5:

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Figura 5 – Matriz relativa a SBOX1 do exemplo.
Fonte: (NIST, 1999)

Pegam-se o primeiro e o último bits da entrada, que no nosso exemplo são "11_2"; converte-se esse número binário para decimal (3_{10}). Isso corresponde à linha 3 da matriz correspondente a S1. Os 4 números binários intermediários são também convertidos, ficando-se no nosso exemplo com "$1001_2 = 9_{10}$"; que corresponde à coluna 9 da matriz. É importante que a leitura das linhas e colunas da matriz devem ser feitas de cima para baixo e da esquerda para direita, começando-se sempre pelo índice 0. Com base nisto, buscando o valor correspondente à essa posição dentro da matriz, encontramos o valor "11_{10}", que corresponde em binário o número "1101_2". Portanto, concluímos que para a S1, com entrada "110011_2" temos uma saída igual a "1101_2".

Depois disso, R_i passa por uma nova permutação chamada de *Wire-Crossing*, P-BOX ou permutação direta.

Após esses passos, encerrada a *Função F*, é realizada uma operação XOR entre R_i e o lado esquerdo (L_i) para compor o novo lado direito (R_{i+1}). Já o novo lado esquerdo (L_{i+1}) recebe o valor direto do bloco direito antes das operações (R_i).

A 16ª rodada do algoritmo DES executa as mesmas operações das 15 primeiras, com a diferença que $R_{16} = R_{15}$ e $L_{16} = L_{15} + F.(R_{15}.K_{16})$, onde *F* denota a *Função F*, descrita anteriormente.

Este comportamento, faz com que a implementação em *hardware* deste algoritmo, tenha uma assinatura de potência bastante característica, com áreas bem delimitadas, conforme podemos ver na Figura 6, o que facilita aos atacantes correlacionar os dados processados e operações realizadas pelo dispositivo atacado com seu consumo durante a execução, como veremos ao longo deste trabalho.

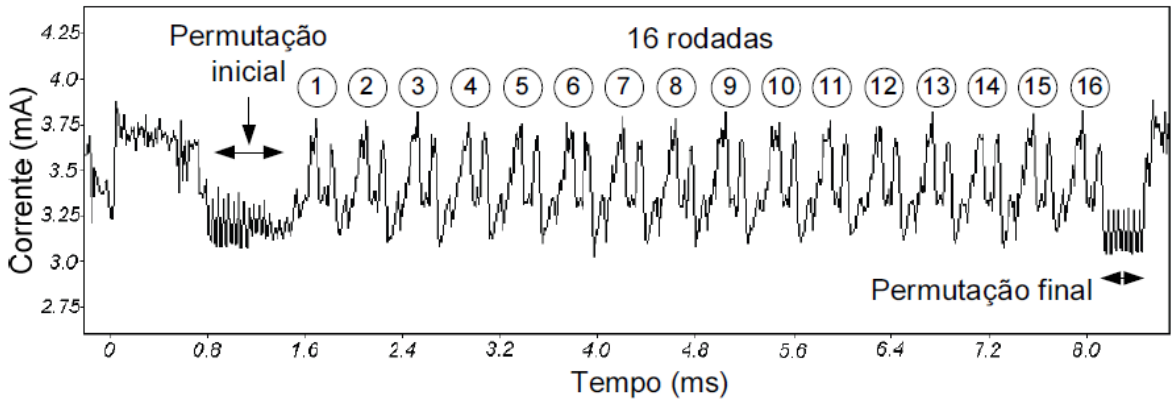

Figura 6 – Assinatura de consumo do algoritmo DES.
Fonte: (SOARES, R. I., 2010)

2.2 Potência em Circuitos Digitais

Circuitos digitais são definidos como circuitos que trabalham com apenas dois níveis de tensão para representar dígitos binários – 0's ou 1's. Sua implementação pode ser concebida através de diferentes tecnologias, como por exemplo, CMOS (do inglês, *Complementary Metal-Oxide-Silicon*). Essa tecnologia é empregada na construção dos transistores que vão formar os elementos dos circuitos digitais. Em sistemas baseados em circuitos digitais, os transistores operam chaveando, ou seja, alterando seu estado entre as regiões de corte (estado de não-condução – situação análoga a uma chave aberta) e saturação (estado de máxima condução – análogo a uma chave fechada) quando o circuito está em funcionamento. Assim, os circuitos

digitais têm seu consumo formado por duas parcelas: o consumo estático, que representa o consumo enquanto os circuitos não estão executando nenhuma operação e o consumo dinâmico originado pelo chaveamento quando da sua atividade.

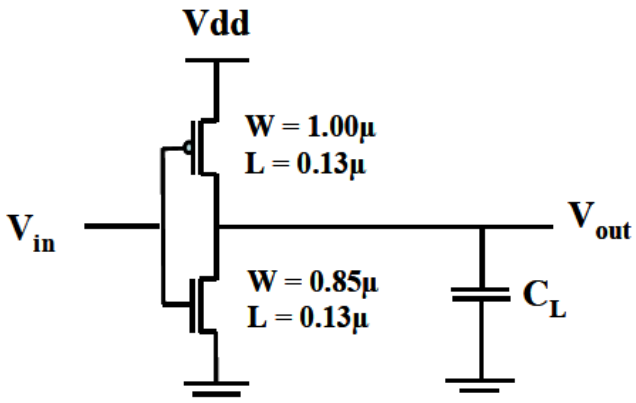

Figura 7 - Porta Lógica Inversor CMOS.
Fonte: (SOARES, R. I., 2010)

Esse chaveamento é um dos componentes importantes para a potência total dos circuitos digitais. O que nos faz perceber que uma parte da potência tem relação com os dados de entrada, os quais vão determinar as transições dos transistores do corte para a saturação e vice-versa, ou sem transição, no caso de sequências de níveis altos ou baixos. Mesmo sem a alteração dos dados de entrada, existe taxa de chaveamento devido à rede de relógio dos circuitos síncronos. Outra característica interessante da atividade de chaveamento é que o consumo de uma transição de nível alto para baixo é diferente do consumo de uma transição de baixo para alto. Pois os tempos das transições de níveis lógicos dos circuitos CMOS está associado a cargas e descargas da capacitância da rede de transistores que compõe o circuito. Tomando o inversor CMOS da Figura 7 como exemplo, percebemos que quando a entrada V_{in} está em nível lógico alto, o transistor canal P, conectado ao terminal positivo da fonte de alimentação V_{dd}, encontra-se em condução, e o transistor canal N, conectado ao terminal de terra da fonte encontra-se em corte. Supondo essa condição inicial, temos o capacitor de carga C_L carregado. Fazendo-se agora, a entrada V_{in} ir para nível lógico baixo, o cenário inverte-se, ficando o transistor canal P em corte e o transistor canal N em condução. Isso faz com que o capacitor de carga C_L seja descarregado.

Portanto, com uma variação de nível alto para baixo na entrada da porta, temos uma descarga da capacitância de carga do circuito, e fazendo a entrada variar

de um nível lógico baixo para alto, teremos uma situação de carga dessa capacitância. Com isso, devido a diferença na mobilidade dos portadores majoritários dos transistores tipo N e tipo P, temos diferentes tempos para diferentes transições dos dados no circuito. Para reduzir essa diferença nos tempos, é feita uma compensação no tamanho dos transistores, fazendo com que o consumo se torne desigual para as transições de nível lógico dos dados no circuito. Isso faz com que tenhamos uma fuga importante de informações com base na atividade de chaveamento dos circuitos integrados. Uma contramedida para o exposto anteriormente é tentar desenvolver circuitos com potência o mais uniforme possível (VAHEDI, H.; MURESAN, R.; GREGORI, S., 2006).

Outra relação muito importante se dá através do consumo e o peso *Hamming* dos dados que estão sendo processados. O peso *Hamming* (em inglês, *Hamming Weight*) é dado pelo número de bits em nível lógico alto, ou seja, ‘1’.

Nos circuitos CMOS a potência é diretamente proporcional ao número de transistores em chaveamento, ou seja, o número de transições. Assim, conhecendo-se os dados de entrada de determinado circuito CMOS, é possível modelar seu consumo segundo o peso *Hamming* das suas saídas. Consequentemente, uma análise dos traços de potência pode permitir a descoberta dos dados manipulados.

Com isto, podemos concluir que a parte dinâmica da potência dos circuitos digitais está relacionada com o tipo de transições efetuadas e com a atividade do circuito, ou ainda com o peso *Hamming* dos dados tradados.

A outra parte da potência, a parte estática, se deve às capacitâncias parasitas e a alimentação das portas lógicas do circuito. Mesmo o circuito estando sem atividade de chaveamento, ou seja, em estado de equilíbrio, existe um potência, devido ao fato dos transistores apresentarem uma corrente de fuga, mesmo estando na região de corte.

A energia estática média consumida, tanto em nível lógico alto, quanto em nível lógico baixo, seguem a Equação (1) (HALLIDAY, RESNICK e WALKER, 2012):

$$E = \frac{1}{2} C_L {V_{dd}}^2 \tag{1}$$

Tomando-se a energia média consumida total, considerando nível lógico alto e baixo, temos:

$$E = C_L {V_{dd}}^2 \quad (2)$$

Além disso, a potência estática total consumida pode ser definida como a razão entre a energia e o tempo gasto (HALLIDAY, RESNICK e WALKER, 2012) de acordo com as Equações (2) e (3).

$$P = \frac{E}{T} \quad (3)$$

$$P = \frac{C_L {V_{dd}}^2}{T} \quad (4)$$

$$P = C_L {V_{dd}}^2 f \quad (5)$$

A potência estática sempre foi insignificante em relação ao consumo total dos circuitos até algumas décadas atrás, porém nas tecnologias mais recentes vem se tornando cada vez mais significativa (KIM, N. S.; AUSTIN, T.; BLAAUW, D.; MUDGE, T.; FLAUTNER, K.; HU, J. S.; IRWIN, M. J.; KANDEMIR, M.; NARAYANAN, V., 2003). Com a evolução da tecnologia, a espessura do canal dos transistores MOS é cada vez menor fazendo com que a corrente de fuga entre os terminais de Dreno e Fonte seja maior. Para as tecnologias futuras, a tendência é que naturalmente a fuga de informações por este canal lateral sejam cada vez menores (KIM, N. S.; AUSTIN, T.; BLAAUW, D.; MUDGE, T.; FLAUTNER, K.; HU, J. S.; IRWIN, M. J.; KANDEMIR, M.; NARAYANAN, V., 2003).

Com base nas características da potência dos circuitos digitais vistas anteriormente, e inserindo-se uma parcela correspondente à atividade de chaveamento, é possível determinar o consumo total de potência através da Equação (6):

$$P = \alpha C_L {V_{dd}}^2 f \quad (6)$$

Onde α é a taxa de atividade de chaveamento da porta, C_L é a carga capacitiva, f é a frequência de operação e V_{dd} a tensão de alimentação.

Com uma rápida análise na Equação (6), podemos ver facilmente que variando-se algum(ns) dos seus parâmetros, como por exemplo, mudar a tensão de

alimentação da porta, ou então variar a frequência de relógio do circuito é possível alterar os traços de potência do circuito, o que dificulta os ataques baseados na potência. Isso será abordado posteriormente neste trabalho. Nos ataques por análise do consumo, são utilizadas as variações instantâneas de potência do circuito, ou seja, a potência dinâmica.

2.3 Ataques por Análise diferencial de Potência (DPA) e de Emissão Eletromagnética (DEMA)

Os ataques apresentados por Kocher et al. em 1999 (KOCHER, P. C.; JAFFE, J.; JUN, B., 1999), exploram a relação de dependência entre a potência de um dado circuito digital dedicado a execução de algoritmo criptográfico com os dados processados pelo mesmo. Neste trabalho, os autores mostram que existem duas formas de utilizar as informações contidas nos traços de potência dos circuitos criptográficos para realizar ataques.

A primeira forma, exige conhecimento detalhado sobre o algoritmo criptográfico e o modo como foi implementado. Em contrapartida, exige uma quantidade pequena de traços para sua execução, onde basicamente é realizada uma interpretação direta dos traços, explorando as relações da assinatura do consumo e as operações executadas sobre os dados durante o processamento. Por esta característica esse ataque é denominado SPA (do inglês *Simple Power Analysis* – Análise Simples de Potência) ou SEMA (em inglês, *Simple ElectroMagnetic Analysis* – Análise Simples de Emissão Eletromagnética), dependendo da grandeza física monitorada durante o ataque. Estes ataques são úteis quando poucos traços de potência do dispositivo atacado estão disponíveis.

A outra forma de ataque por análise da potência são os ataques DPA ou DEMA. Nestes ataques não é necessário um conhecimento detalhado do algoritmo criptográfico e de sua implementação. Porém, DPA exige uma grande quantidade de traços para a análise de potência. Outra característica interessante dos ataques DPA é que mesmo diante de perturbações elétricas durante o monitoramento e aquisição dos traços, é possível realizar ataques bem sucedidos. Seu custo de execução é relativamente baixo e por tratar-se de um ataque não-invasivo, não deixa vestígios no dispositivo atacado. Ainda, este tipo de ataque possui um bom índice de sucesso. Essas características tornaram os ataques DPA, os mais populares dentro das

criptoanálises.

A forma como os traços de potência são analisados em DPA, difere do SPA. No SPA os traços são analisados diretamente, ou seja, o atacante analisa o traço ao longo do tempo, tentando identificar padrões de consumo para correlacioná-los com os dados e as operações realizadas. Já no DPA a potência é analisada em instantes fixos de tempo comparando com os demais traços, verificando o comportamento de instruções iguais executadas com dados distintos e assim estabelecendo relações de dependências entres estes.

O ataque DPA é composto de 5 etapas: (*i*) escolher um resultado intermediário alvo, (*ii*) medir e coletar traços, (*iii*) calcular valores intermediários hipotéticos, (*iv*) aplicar modelo de consumo ao dispositivo atacado e (*v*) avaliar hipóteses de subchaves.

A primeira etapa do ataque consiste em escolher um resultado intermediário do algoritmo criptográfico alvo. Esse resultado precisa ser uma função de parte da chave secreta *k* e parte da mensagem de entrada ou saída conhecida *d* $\left(f(d,k)\right)$. Se o atacante obtiver uma função que satisfaça essa condição, esta pode ser utilizada como alvo do ataque para encontrar *k*. A mensagem conhecida d pode ser tanto uma mensagem de entrada ou um criptograma de saída, ou até mesmo outro dado intermediário que seja conhecido.

Na segunda etapa do ataque, a potência é medida para várias encriptações ou decriptações realizadas sobre um conjunto de *D* dados distintos escolhidos na etapa anterior, usando a mesma chave criptográfica. Assim, *d* é um conjunto contendo os dados *di*, escolhidos na primeira etapa, na forma $d = \{d_1, d_2, \dots d_D\}$.

Para cada encriptação ou decriptação é armazenado um traço *ti* de potência correspondente. São armazenadas *T* amostras de potência em cada traço, formando um conjunto $t_i' = \{t_{i,1} t_{i,2}, \dots, t_{i,T}\}$ para cada traço. Finalmente, esses traços são armazenados em uma matriz M_T de tamanho $D \times T$.

Nesta etapa, podemos perceber que os traços de potência devem estar bem alinhados para que o ataque DPA obtenha sucesso, pois cada coluna t_j da matriz M_T deve corresponder as mesmas operações durante a encriptação ou decriptação para que possam ser comparadas e analisadas.

Assim, para que o ataque tenha eficiência, os traços de potência analisados, devem estar alinhados horizontalmente, ou seja, no domínio do tempo conforme

vimos.

A etapa seguinte, a terceira etapa, consiste do cálculo de valores hipotéticos intermediários para todas as possibilidades de valores de *k,* relembrando que *k* são hipóteses da chave secreta. Ou seja, faz-se o cálculo de valores hipotéticos intermediários para todos os valores possíveis de chave.

As hipóteses de chave são denotadas por um conjunto $k = \{k_1, k_2, \dots, k_K\}$, onde *K* é o número total de possibilidades de chave *k*. Assim, através do conjunto de dados ***d*** e do conjunto de chaves hipotéticas ***k***, o atacante pode calcular todos os valores intermediários hipotéticos possíveis para $f(d, k)$.

$$v_{i,j} = f(d_i, k_j) \; i = 1,\dots, \text{D e } j = 1,\dots, \text{K} \tag{7}$$

Esses valores formam uma matriz denominada M_V, que como visto na Equação (7), tem tamanho $D \times K$.

Podemos observar, que cada coluna *j* da matriz M_V contém os resultados calculados para a hipótese de chave *kj*, através de $f(d_i, k_j)$. Obviamente que, se M_V possui os resultados intermediários para todas as possibilidades de chave *k*, então uma de suas colunas possui os valores intermediários reais calculados pelo sistema criptográfico durante a encriptação ou decriptação dos dados, realizados na segunda etapa.

Também é óbvio que a chave secreta é um dos elementos do vetor ***k*** descrito anteriormente. Esse elemento é denominado de k_{ck}. Assim, o ataque DPA procura descobrir, em qual coluna da matriz M_V encontram-se os mesmos valores produzidos por $f(d, k)$ durante a encriptação ou decriptação do vetor ***d***.

A quarta etapa do ataque DPA, é a etapa em que o modelo de consumo é aplicado ao dispositivo que está sendo atacado. São relacionados os traços de potência com os Pesos *Hamming* dos resultados intermediários $v_{i,j}$ anteriormente calculados, pois o Peso *Hamming* de um circuito CMOS é proporcional à potência do mesmo. Uma alternativa a este modelo é o uso da Distância *Hamming* para modelar a potência. Esta alternativa aproxima-se mais do consumo real de um circuito digital e, portanto, é mais usado na prática. A distância *Hamming* entre dados com representação binária é igual a diferença de seus Pesos *Hamming*, o que pode ser obtido logicamente pela execução de uma operação XOR. Nos ataques DPA a

distância *Hamming* é aplicada sobre os valores intermediários calculados pela função escolhida na primeira etapa, tal como mostrado na Equação (8).

$$HD = r_0 \oplus r_1 \tag{8}$$

onde r_0 e r_1 são pesos *Hamming* e *HD* é a distância *Hamming*.

Com isso, Kocher et al. em (KOCHER, P. C.; JAFFE, J.; JUN, B., 1999) associaram a cada valor intermediário hipotético calculado $v_{i,j}$ um único valor binário $h_{i,j}$ relacionado ao consumo deste valor intermediário da seguinte forma: se o consumo relacionado à este valor for alto, ou seja, $v_{i,j} = 1$, então $h_{i,j} = 1$, caso contrário, $h_{i,j} = 0$. Estes valores $h_{i,j}$ formam uma matriz chamada de M_H.

Finalmente, a última e quinta etapa do ataque DPA, tem como objetivo avaliar as hipóteses de subchaves, atividade realizada a partir das matrizes M_T dos traços de potência e M_H a matriz dos valores de consumo hipotéticos calculados a partir do modelo de potência utilizado.

Cada coluna h_j da matriz M_H é comparada com a coluna correspondente t_j da matriz M_T, ou seja, nesta etapa o atacante compara os valores de consumo hipotéticos de cada hipótese de chave com os traços de potência coletados do dispositivo atacado. Os resultados são armazenados em uma matriz M_R de tamanho $K \times T$. Cada elemento dessa matriz, denominado de $r_{i,j}$ é a comparação com base no método da diferença das médias entre as colunas h_j e t_j, proposto por Kocher em 1999.

Pode-se concluir do exposto até agora, que temos os traços de potência t_i do sistema criptográfico ao executar encriptação ou decriptação para diferentes dados de entrada. Resultados intermediários $v_{i,k}$ calculados com base nos dados de entrada, dentre os quais está o resultado obtido com a chave secreta do sistema de criptografia v_{ck}, pois todas as possibilidades de chave são utilizadas nos resultados intermediários. Disto, podemos notar que em algum momento, ou instante de tempo denominado ct, os traços de potência estão relacionados ao resultado intermediário v_{ck} executado sobre a chave secreta verdadeira do circuito criptográfico atacado.

Como os valores hipotéticos de potência $h_{i,j}$ são calculados sobre os resultados intermediários $v_{i,j}$, podemos perceber que h_{ck}, que é a potência hipotética calculada com o resultado intermediário da chave secreta verdadeira v_{ck}, está fortemente relacionado à t_{ct}, que é o traço de potência medido, no instante que está

executando a encriptação ou decriptação do resultado intermediário para a chave secreta verdadeira v_{ck}.

Devido a esta forte relação entre h_{ck} e t_{ct}, o atacante pode descobrir o índice ck da chave secreta, e por sua vez a própria chave k_{ck}, apenas verificando o maior valor da matriz resultado M_R que será $r_{ck,ct}$. Se os valores da matriz resultado M_R forem aproximados, o atacante sabe que não coletou traços suficientes para determinar a relação entre h_{ck} e t_{ct}.

Como mencionado anteriormente, Kocher et. al apresentaram como método para avaliar as chaves hipotéticas k_i, o método da diferença das médias, que no caso dos ataques DPA é utilizado para relacionar as matrizes M_H e M_T. Esse método dá-se da seguinte forma: primeiramente o atacante divide a matriz dos traços de potência medidos M_T em duas outras matrizes M_{T0} e M_{T1}, sendo M_{T0} composta pelas linhas da matriz M_T cujos coeficientes $h_{i,j}$ da matriz M_H são iguais a zero. E a matriz M_{T1}, é montada com as linhas restantes da matriz M_T. Depois disso, a média das linhas de cada uma das matrizes M_{T0} e M_{T1} é calculada, sendo m'_{0i} o vetor que contém a média das linhas da matriz M_{T0} e m'_{1i} o vetor que contém a média das linhas da matriz M_{T1}. As hipóteses de chave k_i estão corretas se houver uma diferença significativa entre os vetores m'_{0i} e m'_{1i} no mesmo instante de tempo.

A Figura 8 mostra um fluxo de execução do ataque DPA.

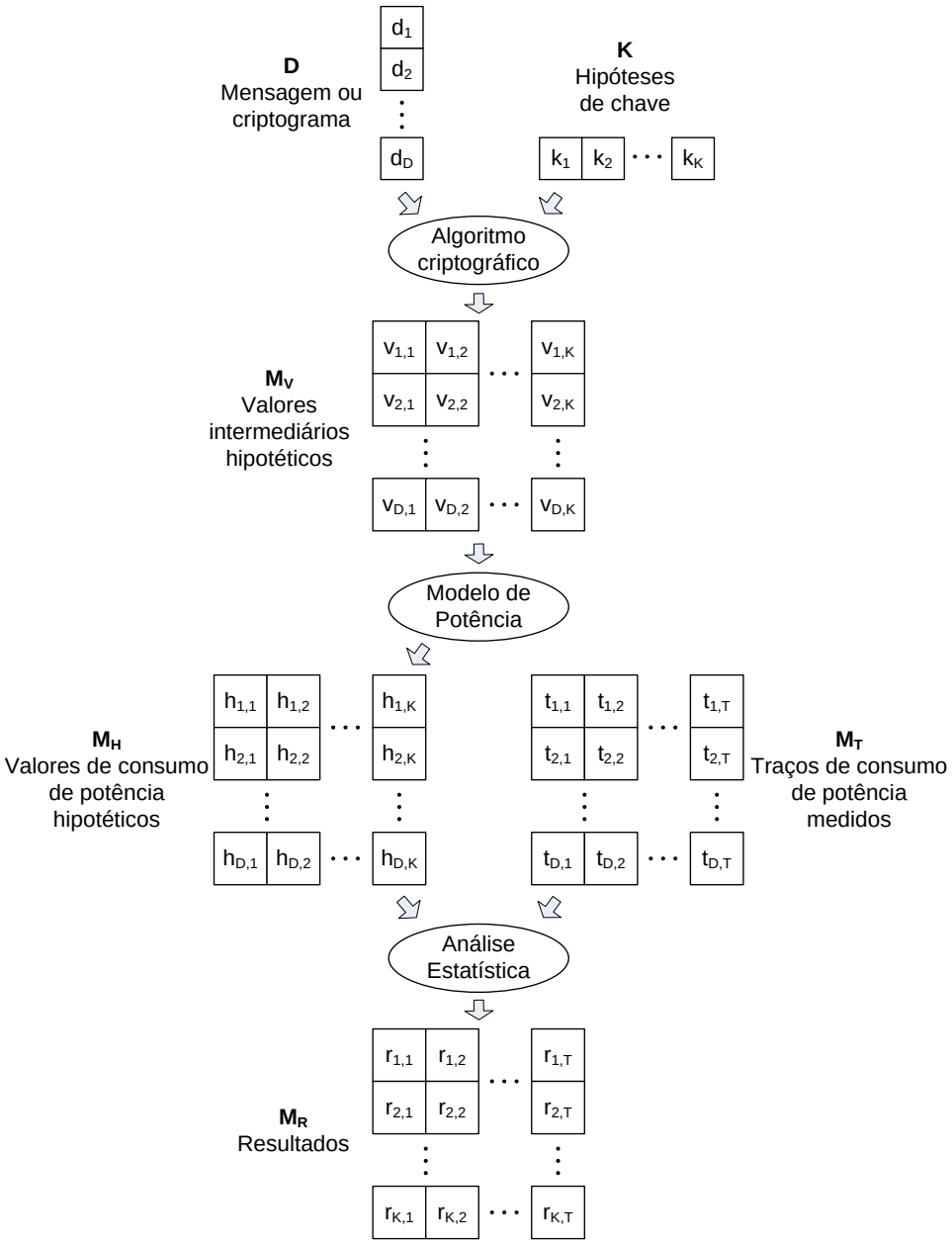

Figura 8 – Fluxo de execução do ataque DPA.
Fonte: (SOARES, R. I., 2010)

Da mesma forma que o ataque DPA, o ataque DEMA (GEBOTYS, C.; TIU, C.; CHEN, C., 2005) avalia e monitora a emissão de ondas eletromagnéticas do dispositivo atacado. Os ataques DEMA capturam os traços gerados pelos campos eletromagnéticos emitidos pelos circuitos durante a execução de encriptação ou decriptação, através de sondas especiais utilizadas em conjunto com estágios amplificadores devido à baixa intensidade do sinal produzido. Para esse tipo de ataque, o atacante deve levar em conta os problemas causados por ruídos e interferências eletromagnéticas do ambiente onde é realizado o ataque, ocasionando

em erros nas leituras dos traços.

2.4 Contramedidas

Na literatura são encontradas muitas estratégias de projeto tanto em software quanto em hardware para reduzir a ação dos ataques DPA e DEMA, chamadas de contramedidas. Estas contramedidas podem ser divididas em três categorias. A primeira consiste em impedir análises de correlação mascarando os dados processados, tornando valores intermediários aleatórios. A segunda, tenta obter um consumo uniforme para qualquer sequência de dados de entrada. Já a terceira consiste em introduzir ruído nas medições de potência, fazendo com que a cada ciclo de relógio a quantidade de potência consumida seja pseudoaleatória. Este trabalho concentra-se em promover uma proposta para atacar dispositivos contendo esta última estratégia de proteção.

Esta terceira categoria de contramedidas baseia-se na injeção de ruídos elétricos a fim de gerar distúrbios na potência do dispositivo. O propósito desta categoria de contramedidas é reduzir a relação sinal-ruído (em inglês, *Signal to Noise Ratio* – SNR) dos traços de potência, dificultando desta forma o sucesso dos ataques. Este tipo de contramedida é interessante, pois visa elevar a contribuição do ruído na SNR sem elevar excessivamente a potência dos circuitos envolvidos.

Neste sentido, o trabalho proposto por (BENINI, L.; MACII, A.; MACII, E.; OMERBEGOVIC, E.; PRO, F.; PONCINO, M., 2003) combina técnicas de redução de potência e controle da atividade de chaveamento através do controle do sinal de relógio para introduzir aleatoriedade, ou ruído, à potência dos sistemas criptográficos. Devido à redução na potência, esta contramedida torna-se atrativa, visto que muitas vezes, os sistemas criptográficos implementados em *hardware* estão submetidos a limitações de potência.

Em (BUCCI, M.; LUZZI, R.; GUGIELMO, M.; TRIFILETTI, A., 2005) os autores propuseram a inserção de elementos de atraso no caminho dos dados do sistema criptográfico. Isso é feito com flip-flops tipo D e um multiplexador, além de um circuito pseudo-aleatório para determinar se os dados passarão pela(s) cadeia(s) de atraso ou não, podendo ser implementadas diversas cadeias de atraso. Isso causa um desalinhamento dos traços de potência no domínio do tempo, dificultando o ataque DPA. Claro que a implementação dessa contramedida implica em aumentar o

hardware. Obviamente, esta contramedida tem como desvantagens o aumento na área, consequência do aumento de hardware, e o aumento da latência do sistema criptográfico, consequência da cadeia de atrasos.

Circuitos que apresentem variações aleatórias na tensão de alimentação e na frequência de relógio (*Dynamic Voltage and Frequency Switching* – DVFS) foram utilizados como contramedida para os ataques DPA em (YANG, S.; WOLF, W.; VIJAYKRISHNAN, N.; SERP, 2005). Esta técnica tem como características a redução no consumo médio do sistema e o aumento no tempo de execução do algoritmo criptográfico. Essa técnica foi analisada posteriormente por (BADDAM, K.; ZWOLINSKI, M., 2007). Neste trabalho, os autores apresentam um método que se baseia na variação aleatória somente da tensão de alimentação dos circuitos criptográficos, mantendo sua frequência de operação constante. Evitando assim, a necessidade de alterações no projeto dos sistemas. Foi observado também, que esta técnica não acarreta em aumento significativo de área, potência ou desempenho. Porém, o método proposto deve impedir o acesso do atacante à entrada do controlador de tensão, pois o mesmo pode substituir o gerador de números pseudo-aleatório por um controle conhecido, removendo assim, a característica de aleatoriedade na variação da tensão. Além disso, a taxa de variação da tensão deve ser menor do que o processamento para impedir que o ataque seja bem sucedido.

Estudos com relação à técnica de inserção de atrasos aleatórios (em inglês, *Random Delay Insertion* – RDI) aplicada a sistemas prototipados em FPGA foram realizados por (LU, Y.; O'NEILL, M.; MCCANNY, J., 2008). Através desses estudos, foi confirmada a efetividade desta técnica para contramedir ataques DPA, com a possibilidade, dependendo do projeto, de otimização de área, consumo e/ou desempenho. (LU, Y.; O'NEILL, M.; MCCANNY, J., 2008) comprovaram em seu trabalho que a implementação da técnica de RDI em FPGA é mais eficiente do que sua versão em *software*, proposta anteriormente por (CLAVIER, C.; CORON, J.; DABBOUS, N., 2000). Tanto (LU, Y.; O'NEILL, M.; MCCANNY, J., 2008) quanto (CLAVIER, C.; CORON, J.; DABBOUS, N., 2000) permitem ajustar o atraso inserido no caminho dos dados. Uma desvantagem da proposta de (LU, Y.; O'NEILL, M.; MCCANNY, J., 2008) é o aumento de área em função do *hardware* responsável pelos atrasos.

Uma outra forma de causar o desalinhamento dos traços, é fazer com que as operações realizadas pelos circuitos criptográficos trabalhem sob frequências de

relógio diferentes. (ZAFAR, Y.; HAR, D., 2008) propõem uma implementação do algoritmo criptográfico AES (em inglês, *Advanced Encryption Standard*) em que a cada entrada de dados, uma nova frequência de relógio é gerada. Esta contramedida também é explorada em (RÉAL, D.; CANOVAS, C., 2008) e (AVIRNENI, N. D. P.; SOMANI, A. K., 2013).

Ainda com relação à classe de injeção de ruídos nas medições da potência, alguns artigos encontrados na literatura colocam como necessária uma etapa de pré-processamento antes de realizar o ataque, a fim de realinhar os traços desalinhados pelas contramedidas. Como em (TIAN, Q.; HUSS, S. A., 2012), (RÉAL, D.; CANOVAS, C., 2008), (TIAN, Q.; HUSS, S. A., 2012) e (LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J., 2007).

2.5 Arquiteturas GALS *pipeline*

Conforme apresentado em (SOARES, R. I., 2010), a proposta combina o estilo de projeto de *hardware* Globalmente Assíncrono e Localmente Síncrono (do inglês, *Globally Asynchronous Locally Synchronous* - GALS) com a implementação de arquiteturas *pipeline* para contramedir os ataques DPA ou DEMA em sistemas criptográficos baseados no algoritmo DES.

Como já apresentado, o algoritmo DES que pode ser implementado tanto em *software* quanto em *hardware*, cria um bloco de encriptação para executar as operações em 16 rodadas de iteração sobre os dados de entrada. Por outro lado, levando em conta o conceito implementações com *pipeline*, que permite a simultaneidade na execução de operações é possível perceber que o bloco das rodadas do algoritmo DES pode ser replicado na construção do *hardware* que o executará. Isso faz com que tenhamos arquiteturas *pipeline* de 2 até 16 estágios para executar todo o algoritmo, possibilitando diferentes configurações para as arquiteturas GALS *pipeline*. Esta forma de implementação permite um aumento na vazão de dados encriptados devido as operações serem realizadas em paralelo. Além disso, o processamento em pipeline causa a sobreposição da potência dos diversos estágios em processamento simultâneo, o que levada a ocultar a informação vazada em cada estágio, sendo este um dos principais objetivos buscado por (SOARES, R. I., 2010).

O projeto do *hardware* segue o estilo GALS, significando que os blocos de encriptação se comunicam de maneira assíncrona, gerenciada por um módulo de

controle que implementa o protocolo de comunicação de 2 fases conhecido como *handshake* (ou aperto de mãos, em português). Desse modo, o estágio transmissor sinaliza o envio de um dado através de um sinal de requisição (em inglês, *Requisition* - Req) para o estágio receptor, que por sua vez, ao perceber a sinalização envia um sinal de reconhecimento (do inglês, *Acknowledge* – Ack) informando que os dados foram recebidos com sucesso.

Cada módulo de encriptação, que executa as operações de uma das 16 rodadas do algoritmo DES, trabalha de maneira síncrona; com seu sinal de relógio gerado com frequência pseudoaleatória por um subsistema externo ao *pipeline*. A frequência de trabalho dos estágios do *pipeline* (os blocos de encriptação), é alterada a cada nova requisição de processamento de dados, adicionando a contramedida de variação da frequência de relógio. Como os blocos de encriptação desta estrutura trabalham de forma síncrona e comunicam-se externamente por meio de protocolos assíncronos, estes são também aqui referenciados como ilhas síncronas. A Figura 9. mostra uma estrutura de 2 ilhas síncronas da implementação do algoritmo DES em arquiteturas GALS *pipeline*.

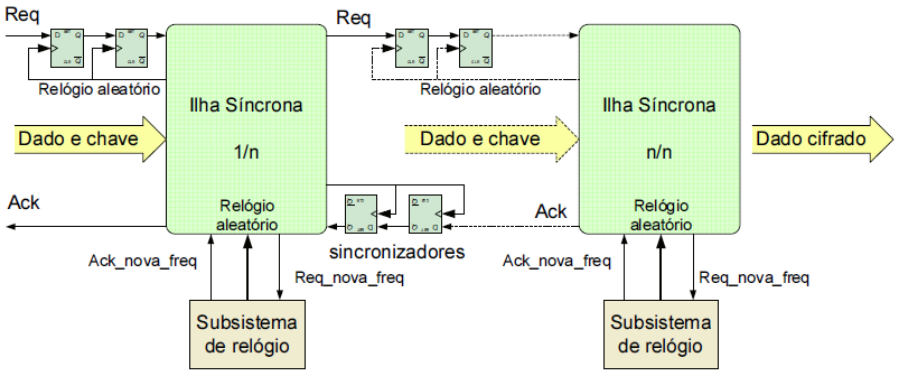

Figura 9 – Arquitetura GALS *pipeline*.
Fonte: (SOARES, R. I., 2010)

Como podemos ver na Figura 9, tanto os sinais de *Req* quanto os de *Ack* do protocolo de comunicação assíncrona precisam passar por dois *flip-flops tipo D* antes de chegar ao seu destino. Isto faz com que sejam necessários dois ciclos de relógio do estágio (ou ilha síncrona) receptor(a) para que estes sinais sejam efetivamente recebidos pelo mesmo(a). Isso é feito para evitar o problema da *metaestabilidade* durante a transmissão de dados, devido às diferentes frequências de trabalho das ilhas síncronas. Os atrasos inseridos pela cadeia de *flip-flops* no início do

processamento de cada estágio, ou ilha síncrona, são pseudoaleatórios, pois dependem da frequência de relógio desta ilha. Como pode ser observado, esse mecanismo provê a contramedida de inserção de atrasos aleatórios para as arquiteturas GALS *pipeline*.

Podemos citar como desvantagens das arquiteturas propostas por (SOARES, R. I., 2010), o acréscimo de área, causado pela replicação do bloco de encriptação do algoritmo criptográfico e o aumento da latência do sistema em função da inserção dos atrasos aleatórios.

Conforme já mencionado, o *pipeline* possibilita a criação de diferentes configurações de implementação do algoritmo DES, em função do maior ou menor número de replicações das rodadas do algoritmo. Com isto, (SOARES, R. I., 2010) prototiparam 4 versões de *hardware* com 2, 4, 8 e 16 estágios. Sendo que para cada uma destas versões existe a possibilidade de trabalhar com um relógio global em modo síncrono ou então com relógios locais independentes para cada estágio, com frequências geradas pseudoaleatoriamente. Deste modo é possível resumir que é possível avaliar 8 configurações diferentes para as arquiteturas GALS *pipeline*.

3 PROCESSAMENTO DE SINAIS

Neste Capítulo são revisados conceitos e técnicas utilizados em processamento de sinais, com o intuito de obter-se um melhor entendimento sobre as etapas de pré-processamento propostas de um modo geral na literatura e que são implementadas e adaptadas ao fluxo de ataque proposto neste trabalho.

3.1 Amostragem e Ruído nos Traços de Potência

A amostragem de um sinal consiste em coletar amostras do sinal original a intervalos de tempo constantes, transformando um sinal contínuo no tempo em um sinal discreto no tempo. O intervalo de tempo com que as amostras são coletadas é denominado de período de amostragem, sendo o seu inverso, a frequência de amostragem tal como representado na Equação (9).

$$f_A = \frac{1}{T_A} \tag{9}$$

Assim, a partir de um sinal contínuo no tempo $x_1(t)$, obtém-se o sinal de tempo discreto $x_2[n]$, através do processo de amostragem, conforme Equação (10):

$$x_2[n] = x_1(n.T_A) \tag{10}$$

A Equação (10) representa um sinal de tempo discreto em função de suas (*n*) amostras. Este sinal pode ser representado no tempo através da Equação (11):

$$x_2(t) = x_2[n].\delta(t - n.T_A) \tag{11}$$

De acordo com o teorema de *Nyquist*, visto em (WILSKY, A. S.; NAWAB, S. H.; OPPENHEIM, A. V., 2010), para que um sinal amostrado contenha ainda a informação do sinal original, a frequência de amostragem deve ser no mínimo maior do que duas vezes a frequência máxima do sinal amostrado, assim:

$$f_A > 2f_S \tag{12}$$

Onde f_A é a frequência de amostragem e f_S é a frequência do sinal amostrado.

Como exemplo do processo de amostragem, temos na Figura 10 uma forma de onda senoidal, e na Figura 11, a mesma senóide, porém amostrada.

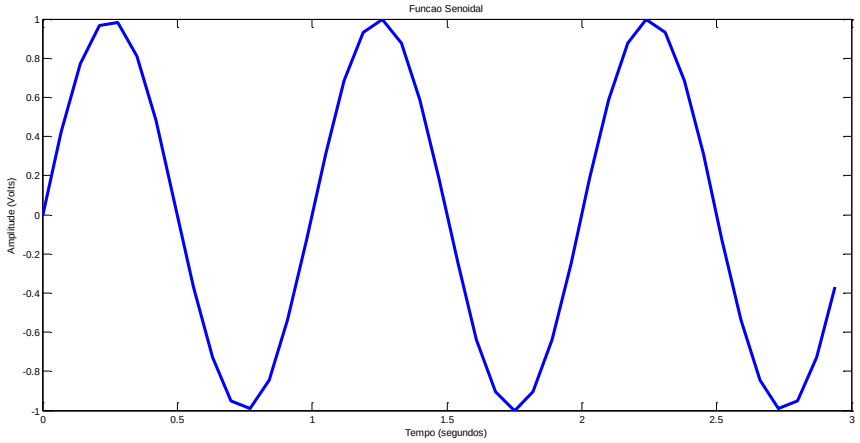

Figura 10 – Sinal Senoidal Contínuo no tempo.
Fonte: Própria.

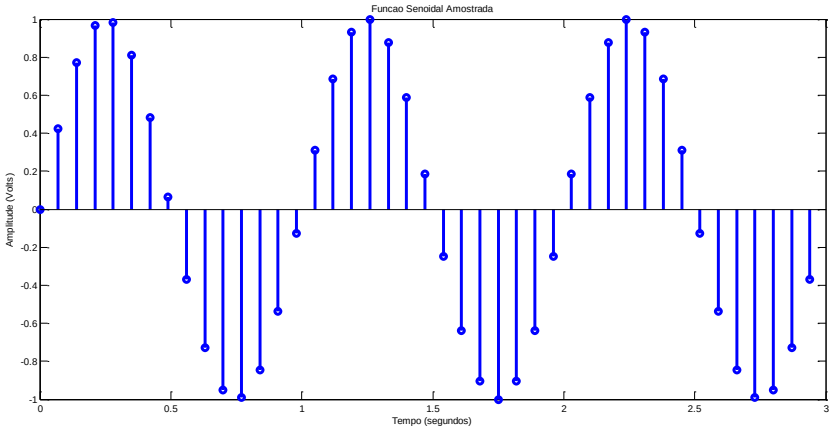

Figura 11 – Sinal Senoidal Amostrado.
Fonte: Própria.

Como os sinais provenientes dos traços de potência são sinais amostrados, seu espectro é o espectro de um sinal discreto no tempo. De (HAYKIN, S.; MOHER, M., 2011), temos que o espectro de um sinal amostrado é igual a multiplicação do espectro do sinal original, antes da amostragem; por um trem de impulsos. O que resulta em múltiplos espectros do sinal original. Isso justifica o teorema de *Nyquist*, pois se a frequência de amostragem for inferior ao dobro da máxima frequência do

sinal amostrado, haverá a sobreposição entre os espectros das amostras, fenômeno conhecido como *aliasing*, como observado na Figura 12.

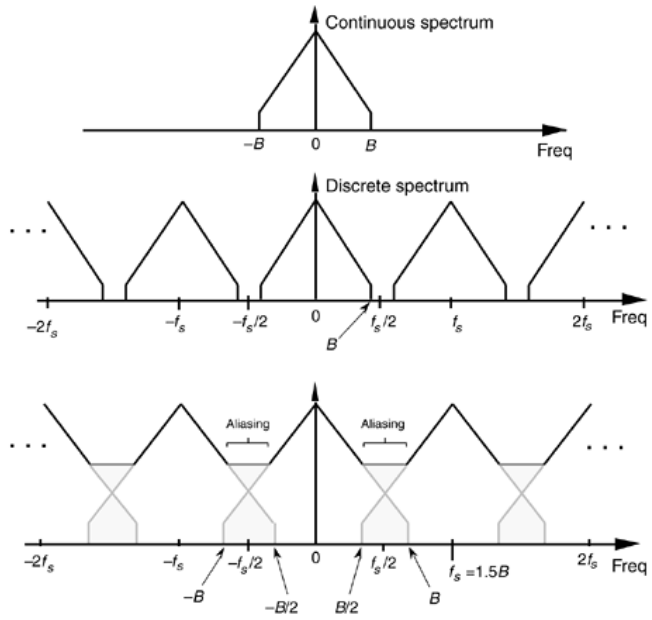

Figura 12 – Aliasing.
Fonte: (HAYKIN, S.; MOHER, M., 2011)

No tempo, o efeito do *aliasing* consiste em coletar amostras em tempos incoerentes em cada período do sinal de tempo contínuo. Para ilustrar, consideremos uma máquina girante em uma indústria sob a iluminação artificial, em que a fonte luminosa oscila entre o estado ligado e desligado em uma determinada frequência. Suponhamos que a frequência de chaveamento da luz, que é a frequência de amostragem, não obedece ao teorema de *Nyquist*, sofrendo, portanto, o fenômeno de *aliasing*. Consideremos ainda, que a frequência de chaveamento da luz seja igual a frequência de rotação da máquina girante. Nesta situação, um observador ao olhar para a máquina girante terá a impressão que a mesma está parada, quando na verdade está em movimento na mesma frequência de chaveamento que a iluminação ambiente, causando uma impressão errada do processo.

Como mencionado, os traços de potência no momento de sua aquisição foram amostrados com uma frequência de amostragem f_A, dada em função do osciloscópio utilizado. Esses sinais estão suscetíveis a ruídos de diversas naturezas, inclusive no próprio processo de amostragem: o ruído de quantização, ruído intrínseco aos circuitos que compõem o sistema criptográfico, o ruído advindo de fontes externas, etc.

O processo de aquisição dos traços de potência também está suscetível a fontes externas de ruídos, como por exemplo, flutuações na tensão de alimentação. Outra fonte de ruído que pode estar presente são interferências eletromagnéticas presentes no ambiente em que está sendo feita a medição, como ondas de comunicação, ruídos gerados por motores elétricos, descargas elétricas, etc. Existem também ruídos gerados pelo funcionamento de aparelhos eletrônicos nas proximidades, chamado de ruído intrínseco, que é um tipo de ruído existente em todos os circuitos eletrônicos, e que se dá devido ao funcionamento dos componentes do circuito. Como exemplo, temos o ruído térmico, que está presente em circuitos compostos por semicondutores como os transistores, presentes nos circuitos digitais. O chaveamento dos transistores é outra fonte de ruído que está presente nos circuitos digitais.

Além disso, é possível considerar os ruídos inseridos propositalmente como estratégia de contramedida aos ataques DEMA/DPA, como por exemplo, a introdução de operações redundantes ou sem propósito durante a encriptação dos dados, entre outros. Isso faz com que o sistema de aquisição dos traços de potência ou radiação eletromagnética necessite de etapas para reduzir ou eliminar estes ruídos.

As medições realizadas em (SOARES, R. I., 2010), por exemplo, não fizeram uso de filtros para reduzir a quantidade de ruído durante a aquisição dos dados. Esta etapa pode ser acrescentada no fluxo dos ataques DEMA/DPA a fim de obter melhor efetividade dos ataques. Como será visto na Seção 3.2, a é realizada uma filtragem na etapa de subamostragem dos traços, eliminando uma parte do ruído presente nos mesmos.

3.2 Subamostragem ou Reamostragem

O conjunto de traços utilizados neste trabalho, disponibilizado por (SOARES, R. I., 2010), possuem frequências de relógio que vão desde 38MHz a 60MHz. Portanto, considerando-se a máxima frequência do conjunto, sua taxa de amostragem deve ser no mínimo de 120MHz. Entretanto, estes traços foram capturados por um osciloscópio com uma taxa de 20GSamples/s. Isso significa que os traços estão superamostrados, ou seja, sua taxa de amostragem é muitas vezes superior ao dobro da máxima frequência dos sinais (166,67 vezes). Portanto, a princípio não há

problemas em reamostrá-los, desde que sua taxa não caia abaixo de 120MSamples/S. Contudo, é aconselhável calcular um fator de subamostragem (Δ) para chegar-se à taxa de amostragem adequada. Este fator de amostragem é calculado com base nas funções de autocorrelação dos traços e de suas primeiras séries de potências, dadas pelas Equações (13) e (14) conforme visto em (LODER, L. L., 2014).

$$r_y[k] = Ey[n].y[n-k] \tag{13}$$

$$r_y[k] = Ey[n]^2.y[n-k]^2 \tag{14}$$

onde *k* é o índice de atraso da função de autocorrelação e *n* é o índice que identifica a amostra dentro da série. Como pode ser visto em (LODER, L. L., 2014), o fator de subamostragem é dado como o menor valor de *k* que minimiza a autocorrelação em cada uma das funções, sendo seu valor para o presente conjunto de traços, igual a 50 (LODER, L. L., 2014). Com isto, temos que a taxa e amostragem dos traços do consumo de pode ser reduzida em até 50 vezes, sem que se perca as informações contidas nestes traços. Em outras palavras, a mínima taxa de amostragem para os traços disponíveis é de 800MSamples/s.

Para implementar a subamostragem ou reamostragem neste trabalho, foi utilizada uma função do MATLAB sobre os traços do consumo de potência processados em etapa anterior do fluxo, que será explicada em seções futuras. Esta função chama-se *resample* (que em português significa reamostragem), e tem o seguinte protótipo: *resample(X,P,Q)*. Em que X é um vetor que armazena a sequência a ser amostrada, neste caso, os valores do consumo de potência. A sequência X é remostrada em P/Q vezes a taxa original, arredondando P/Q para o próximo inteiro maior do que o resultado, caso a divisão resultar em um número fracionário. A reamostragem é feita usando implementação de filtros FIR polifásicos. Esse tipo de filtro divide a resposta ao impulso do filtro passa-baixas digital FIR em *M* diferentes subfiltros, onde *M* é o fator de subamostragem ou reamostragem (P/Q). Como exemplo, considerando *M* = 2, dividimos a equação (15), reescrita abaixo (16) por conveniência, que mostra a resposta ao impulso do filtro FIR no plano *z*, em dois subfiltros: um contendo os coeficientes de índice par e o outro os impares.

$$H[z] = \sum_{l=0}^{M} h[l] \,.\, z^{-l} \tag{15}$$

$$H[z] = \sum_{l=0}^{M} h[2l] \,.\, z^{-2l} + \sum_{l=0}^{M} h[2l+1] \,.\, z^{-(2l+1)}$$

$$H[z] = \sum_{l=0}^{M} h[2l] \,.\, z^{-2l} + z^{-1} \sum_{l=0}^{M} h[2l+1] \,.\, z^{-2l} \tag{16}$$

Reescrevendo os termos da equação (16), temos os dois subfiltros a seguir:

$$E_0[z] = \sum_{l=0}^{M} h[2l] \,.\, z^{-l} \tag{17}$$

$$E_1[z] = \sum_{l=0}^{M} h[2l+1] \,.\, z^{-l} \tag{18}$$

Assim, a equação (16) pode ser subdividida em *M* subfiltros (no nosso caso *M* = 2) para sua implementação, como vemos na equação (19):

$$H[z] = E_0[z^2] + z^{-1} E_1[z^2] \tag{19}$$

A equação (20) pode ser generalizada para um *M* qualquer:

$$H[z] = \sum_{l=0}^{M-1} z^{-l} \,.\, E_l[z^M] \tag{20}$$

Segundo (MATHWORKS, 2014), ao dividir os coeficientes do filtro em dois ou mais subfiltros polifásicos, não são realizados cálculos desnecessários na convolução. Obviamente, as saídas das convoluções com os subfiltros polifásicos são intercaladas e somadas para produzir a saída do filtro.

A função *resample* também aplica um filtro passa-baixas digital FIR anti-

aliasing em *X* durante o processo de reamostragem, e compensa o atraso gerado pelo filtro. Este filtro é implementado com a função *firls*, que serve para projetar filtros de fase linear FIR, usando minimização de erro através de mínimos quadrados. Esta função possui a seguinte assinatura: *firls(N, F, A)*, sendo *N* a ordem do filtro. *F* é um vetor de pares de frequência, especificado na faixa entre 0 e 1, em que 1 corresponde a frequência de Nyquist. Deve ser observado que as frequências devem estar em ordem crescente. E *A* é um vetor de mesmo tamanho de *F*, que contém a amplitude desejada nos pontos especificados em *F*. A função de amplitude desejada nas frequências entre pares de pontos (F[k], F[k+1]) é o segmento de linha que conecta os pontos (F[k], A[k]) e (F[k+1], A[k+1]) para valores de k impares. Para k com valores par, a resposta não é especificada, sendo chamadas de regiões de transição, ou regiões que não importam (do inglês, *don't care regions*). Assim, a resposta de amplitudes é linear por partes, com bandas de transição. O erro quadrático neste tipo de filtro é minimizado. A Figura 13, ilustra a relação entre os vetores F e A na definição de uma resposta de amplitude desejada.

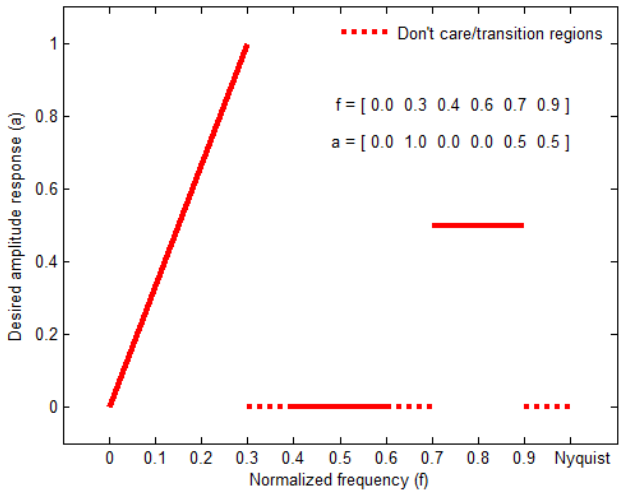

Figura 13 – Relação entre F e A.
Fonte: (MATHWORKS, 2014)

4 TRABALHOS RELACIONADOS

Neste Capítulo é apresentada uma revisão dos principais trabalhos que visam encontrar vulnerabilidades em sistemas criptográficos protegidos por contramedidas baseadas no desalinhamento horizontal, ou seja, que de algum modo inserem distúrbios temporais durante a execução de algoritmos criptográficos alvo de ataques. Estes trabalhos em sua essência procuram estratégias para realinhar traços de potência que contém a assinatura da execução do algoritmo criptográfico distorcida pelas contramedidas já mencionadas.

4.1 Técnicas de Alinhamento baseadas em Sliding Window e Phase-Only Correlation

Para neutralizar a contramedida de inserção de atrasos aleatórios (CLAVIER, C.; CORON, J.; DABBOUS, N., 2000) apresentam uma técnica que baseia-se na reconstrução dos picos de correlação gerados como resultado dos ataques.

Conforme revisado no Capítulo 2, o ataque DPA é realizado sobre uma parte, ou função, do algoritmo criptográfico atacado. Conforme (CLAVIER, C.; CORON, J.; DABBOUS, N., 2000), um exemplo pode ser uma das SBOX do algoritmo DES. Quando executado o ataque, o resultado são traços diferenciais, em que o pico desses traços indica a subchave correta.

Segundo os autores, para que o pico do traço diferencial correspondente à chave correta seja identificado, a inequação (21) deve ser satisfeita:

$$\Delta_D(j) > {}^{\sigma}/_{\sqrt{N}} \tag{21}$$

Onde $\Delta_D(j)$ é o j-ésimo elemento do traço diferencial, σ representa o ruído e *N* o número de traços necessários. Mas, devido a contramedida de inserção de atrasos aleatórios, o ruído da equação (21) tem seu valor aumentado. O resultado disto, é um aumento no número de traços necessários, *N*, para que o ataque tenha sucesso.

Portanto, o trabalho de (CLAVIER, C.; CORON, J.; DABBOUS, N., 2000) propõe a reconstrução da amplitude do pico Δ_D, reduzindo assim, o número de traços de potência necessário. Para isto, o novo valor do sinal deve ser $\frac{\Delta_D(j)}{k}.k$, e o novo

ruído $\frac{\sigma}{\sqrt{N''}}.\sqrt{k}$.

Então o pico será visível se:

$$\Delta_D(j) > \frac{\sigma}{\sqrt{N''}}.\sqrt{k} \tag{22}$$

Sendo que *N''* deve ser:

$$N'' = kN \tag{23}$$

Devido a este método realizar a integração dos valores dos traços de potência em *k* ciclos consecutivos, os autores o denominaram de DPA com Janela Deslizante (do inglês, *Sliding Window* DPA – SW-DPA). Este método consiste em dois passos: Primeiramente, as curvas diferenciais devem ser obtidas. No segundo passo, as curvas diferenciais obtidas no passo anterior devem ser integradas. Ou seja, deve-se somar os valores ponto a ponto nos *k* ciclos consecutivos.

Ainda, uma técnica baseada no peso *Hamming* é utilizada para encontrar-se os bits que vazam mais informação.

Este trabalho possui como limitação a implementação apenas em *software* não sendo possível sua implementação em *hardware*.

Nagashima em (NAGASHIMA, S.; HOMMA, N.; IMAI, Y.; AOKI, T.; SATOH, A., 2007) demonstram a definição da função de correlação de fase, *Phase-Only Correlation – POC*, utilizando-a para realinhar os traços de potência, desalinhados tanto por problemas durante a medição quanto por contramedidas presentes no sistema criptográfico atacado, antes de realizar os ataques DPA. Como prova de conceito, os autores realizam ataques ao algoritmo DES implementado em *software* e executado no microprocessador Z80, utilizando como contramedida atrasos aleatórios através de funções NOPs (do inglês, *No Operations*).

Conforme *Nagashima*, o espectro de fase cruzado normalizado é dado pela Equação (24):

$$R_{FG}[k] = \frac{F[k]\overline{G[k]}}{|F[k]G[k]|} = e^{j\theta_{FG}[k]} \tag{24}$$

Onde $F[k]$ e $G[k]$ são as *Transformadas Discretas de Fourier* (revisada em

capítulo prévio) de duas funções denotadas por $f[n]$ e $g[n]$, $\overline{G[k]}$ é o complexo conjugado de $G[k]$ e $\theta_{FG}[k] = \theta_F[k] - \theta_G[k]$.

Dada $R_{FG}[k]$, a função *POC* é encontrada através da *Transformada Inversa de Fourier Discreta* de acordo com a Equação (25).

$$r_{fg}[n] = \frac{1}{L} \sum_{k=-Z}^{Z} R_{FG}\,[k] W_L^{-kn} \tag{25}$$

Esta função resulta em um pico abrupto, se houver similaridade entre as formas de onda, sendo que quando $f[n] = g[n]$, a função resulta em um impulso, dado pela função Delta de *Kronecker*, sendo possível determinar o grau de similaridade entre duas formas de onda, observando a altura do pico da $r_{fg}[n]$. Já a localização do pico indica o deslocamento de fase entre as duas ondas.

Ainda, (NAGASHIMA, S.; HOMMA, N.; IMAI, Y.; AOKI, T.; SATOH, A., 2007) comentam que foram empregadas técnicas avançadas de alta precisão para estimar o deslocamento, como: janelamento para reduzir os efeitos de borda e ponderação espectral para reduzir os efeitos do ruído. Para isso, os autores tomam qualquer traço dentro do conjunto por eles obtido como referência e então avaliam e ajustam os erros de deslocamento entre o traço referência e todos os outros do conjunto, antes de realizar o ataque DPA, resultando em um incremento de tempo de apenas 5% do tempo do ataque no fluxo total. Porém, este método possui uma limitação para a faixa de frequências de relógio dos traços. Assim, este método não é aplicável a traços com uma faixa de frequências com mais de 10% de variação.

Em (GUILLEY, S.; KHALFALLAH, K.; LOMNÉ, V.; DANGER, J., 2011) os autores comparam POC com uma técnica chamada de correlação de amplitude (do inglês, *Amplitude Only Correlation* – AOC) e também com uma técnica por eles proposta, denominada de T-POC (em inglês, *Threshold – Phase Only Correlation*). O artigo destaca que no POC as *Transformadas Discretas de Fourier* das formas de onda de referência e a deslocada são normalizadas antes de serem multiplicadas. De acordo com (GUILLEY, S.; KHALFALLAH, K.; LOMNÉ, V.; DANGER, J., 2011) esta normalização pode fazer com que POC apresente resultados falsos de alinhamento das formas de onda. Em sua comparação do POC com AOC, é destacado que por não realizar esta normalização em seu processo AOC não possui o mesmo problema. Porém este método depende das amplitudes das formas de onda em questão, o que

caracteriza uma limitação. Em sua proposta, o T-POC, os autores somam um fator de correção, designado por ε, ao denominador do POC para reduzir os efeitos negativos causados pela normalização anteriormente mencionada. Com isto, o cálculo do T-POC entre duas formas de onda denotadas por *X* e *Y*, é dado pela Equação (26).

$$T - POC(X;Y) = IDFT\left(\frac{\overline{DFT(X)}.DFT(Y)}{\left|\overline{DFT(X)}\right|.|DFT(Y)| + \varepsilon}\right)/n \quad (26)$$

Onde *n* é o fator de normalização.

A definição do fator de correção ε do T-POC é feita de forma empírica segundo os autores, o que implica em uma limitação para este método.

(LODER, L. L., 2014) também faz uso de POC como uma técnica de pré-processamento em seu fluxo proposto juntamente com a aplicação de filtragem através de filtro digital passa-baixas FIR de médias móveis. (LODER, L. L., 2014) aplicou POC em sistemas criptográficos dotados de contramedidas de variação de frequências de relógio e inserção de atrasos aleatórios, utilizando como estudo de caso as arquiteturas GALS *pipeline* de (SOARES, R. I., 2010). Em seu trabalho (LODER, L. L., 2014) mostra que é possível encontrar-se os dados ocultos, mesmo em dispositivos dotados das contramedidas citadas. Porém, como mencionado anteriormente, este método possui o inconveniente de não poder ser aplicado em traços com frequências muito distintas, o que pode acarretar em uma quantidade insuficiente de traços com frequências de relógio próximas.

A Tabela 1 apresenta um resumo das técnicas de alinhamento revisadas nesta seção.

4.2 Técnicas de Alinhamento baseadas em Dynamic Time Warping

Outra ferramenta na área de processamento de sinais que pode ser utilizada para o realinhamento de traços de potência nos ataques DPA é a *Dynamic Time Warping – DTW,* que é um algoritmo de comparação utilizado em sistemas de reconhecimento de voz, conforme (SAKOE, H.; CHIBA, S., 1978).

(WOUDENENBERG, J; WITTEMAN, M; BAKKER, B., 2009) utilizaram o método DTW para realinhar os traços de potência horizontalmente. Contudo, este método calcula o caminho da distorção, necessitando gerar matrizes para armazenar

Tabela 1 - Comparativo entre propostas de alinhamento baseadas em SW e POC

Trabalho	Ferramenta	Contribuição	Limitação
[CLAVIER et al., 2000]	SW-DPA	Neutraliza a contramedida de inserção de atrasos aleatórios	Aplicável somente em implementações em *software*
[Nagashima et al., 2007]	POC	Realinha os traços com pouco esforço computacional	Não se aplica em traços com frequências muito distintas
[GUILLEY et al., 2011]	T-POC	Reduz o erro causado pela normalização realizada pelo POC	Exige a definição empírica de um fator de correção, ε.
[LODER et al., 2014]	POC + Filtro	Fluxo de ataque para neutralizar contramedidas por variação de frequência	Quantidade de traços para obter sucesso e necessidade de dividir os traços por frequência.

os valores intermediários para cada padrão combinado, o que demanda esforço computacional e necessita também de uma grande quantidade de espaço de memória para armazenar uma grande quantidade de dados do modelo e dos traços desalinhados; o que claramente torna-se uma desvantagem desse método.

No DTW, o alinhamento entre duas sequências atua distorcendo a forma de onda dessas sequências a fim de que as duas possuam a maior semelhança possível, ou seja, a menor distância euclidiana. Essa distorção pode ocorrer apenas nas séries que são comparadas com a referência (forma assimétrica) ou em ambas as séries (forma simétrica).

A forma simétrica do DTW dificultaria sua aplicação nos ataques DPA em função da grande quantidade de traços a ser realinhada, uma vez que a cada realinhamento de um par de traços, teríamos a deformação do traço escolhido como referência. Por este motivo, (WOUDENENBERG, J; WITTEMAN, M; BAKKER, B., 2009) utilizam DTW na forma assimétrica em seu trabalho para realinhar os traços de potência antes de realizar os ataques DPA.

(WOUDENENBERG, J; WITTEMAN, M; BAKKER, B., 2009) comentam que o tempo de processamento e a complexidade (que é quadrática, $O(n^2)$ do DTW podem ser restritivos na prática. Assim, os autores propõem o uso de um algoritmo que apresenta os mesmos resultados do DTW, porém com complexidade linear $O(n)$, para realizar o alinhamento dos traços. Este algoritmo é chamado de FastDTW. Mesmo o FastDTW apresenta grandes tempos de computação e exige grandes quantidades de

memória.

(LODER, L. L., 2014) realiza em seu trabalho experimentos utilizando DTW, que obtiveram melhores resultados em relação ao alinhamento utilizando POC, descrito anteriormente. Destacando o fato de que o DTW consegue alinhar traços de diferentes frequências com diferentes números de amostras. Porém, conforme mencionado, o tempo de execução do alinhamento por DTW é muito superior ao método que utiliza a correlação de fase, POC. Em seus experimentos (LODER, L. L., 2014) constataram que DTW teve um tempo de execução de aproximadamente 200 vezes o tempo necessário para execução do algoritmo POC, o que nas condições dos experimentos realizados se aproxima ao tempo de computação do ataque DPA, que é a etapa mais lenta do fluxo inteiro. Através de uma estimativa, os autores verificaram que seria inviável realizar o alinhamento dos 100000 traços do conjunto utilizado como estudo de caso, levando meses para terminar o processamento. Para que fosse possível a realização do ataque nos traços realinhados por DTW, os mesmos foram subamostrados, a fim de reduzir o esforço computacional para realizar o alinhamento.

Uma comparação entre as técnicas de alinhamento presentes nesta seção é mostrada na Tabela 2.

Tabela 2 - Comparativo entre propostas de alinhamento baseadas em DTW

Trabalho	Ferramenta	Contribuição	Limitação
[Woudenberg, Witteman, Bakker, 2009]	FastDTW	Método muito eficaz para alinhar os traços, não tendo a frequência como limitação	Grande esforço computacional, o que eleva muito o tempo de execução e necessidade de grande quantidade de memória, mesmo com FastDTW
[LODER et al. 2014]	Wavelets + DTW	Eficácia no alinhamento dos traços	Necessidade de subamostrar os traços para que fosse possível a execução do método. Porém, ainda possui excessivo esforço computacional e necessidade de memória

4.3 Técnicas de Alinhamento baseadas em Transformada Wavelet

(SOUISSI, Y.; ELAABID, M. A.; DEBANDE, N.; GUILLEY, S.; DANGER, J., 2011) utilizam transformada *wavelet* no processo de alinhamento dos traços para realizar a extração da assinatura alvo do algoritmo criptográfico nos traços do consumo. Em (SOUISSI, Y.; ELAABID, M. A.; DEBANDE, N.; GUILLEY, S.; DANGER, J., 2011) também são realizadas filtragens dos traços com base nas transformadas *wavelets*. Porém, o uso das transformadas *wavelets*, por realizar sucessivas filtragens ou subamostragens tem como limitação a possibilidade de perder-se informação dos traços importante aos ataques.

Já (PATEL, H.; BALDWIN, R., 2012) realizam subamostragem através das transformadas *wavelets* e em seguida realizam o ataque diretamente sobre os traços subamostrados. Com isto, os traços são filtrados, destacando-se a informação da assinatura alvo em relação ao ruído presente nos mesmos. Esse processo tem a vantagem de exigir menos esforço computacional, resultando em menor tempo de execução. Porém, como observado em (SOUISSI, Y.; ELAABID, M. A.; DEBANDE, N.; GUILLEY, S.; DANGER, J., 2011), a subamostragem realizada pode acarretar em perda de informação importante ao ataque.

Como mencionado, (LODER, L. L., 2014) realizam uma subamostragem nos traços para poder executar DTW. Para isso, fazem uso das transformadas *wavelets* justamente por estas realizarem subamostragem dos traços, tornando viáveis os ataques sobre os traços realinhados com DTW. Além disso, os autores realizaram experimentos com alinhamento através da correlação de fase, POC, depois de subamostrar os traços com a transformada *wavelet*, porém o método mostrou-se caro computacionalmente, não sendo considerado satisfatório pelo autor. (CHARVET, X.; PELLETIER, H., 2005) também fazem uso das transformadas *wavelets* para subamostrar os traços com o intuito de filtrá-los.

A Tabela 3 resume a comparação das técnicas desta seção.

Tabela 3 - Comparativo entre propostas de alinhamento baseadas na Transformada Wavelet.

Trabalho	Ferramenta	Contribuição	Limitação
[SOUISSI et al., 2011]	wavelets	Filtrar os traços e extrair a assinatura alvo dos ataques	Possibilidade de perder informação útil ao ataque durante o

			processo
[PATEL; BALDWIN, 2012]	wavelets	Menor esforço computacional por aplicar os ataques aos traços subamostrados pelas wavelets	Como em [SOUISSI et al., 2011], existe a possibilidade de perder informação útil ao ataque durante o processo
[LODER et al., 2014]	Wavelet + POC	Subamostrar os traços, reduzindo o esforço computacional dos ataques.	Método caro computacionalmente.

4.4 Técnicas de Alinhamento variadas

(LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J., 2007) propusera um método para alinhamento temporal dos traços de potência baseado no cálculo da energia dos mesmos. Este método consiste em dividir os traços de potência em segmentos e calcular a energia dos mesmos. Segundo observaram os autores, independentemente dos deslocamentos temporais entre diferentes traços, a quantidade de energia contida na parcela compreendida dentro do segmento, se mantém praticamente a mesma. A Figura 14 ilustra um exemplo da aplicação deste método. Considerando-se que a informação relevante destes sinais encontra-se no pico, podemos perceber que embora o pico esteja deslocado entre nos traços s_1, s_2 e s_3, suas energias são praticamente idênticas.

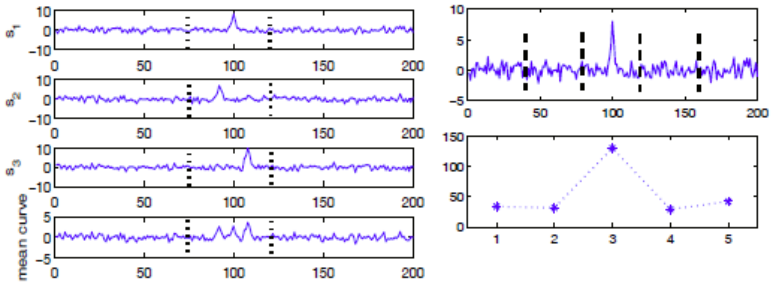

Figura 14 – Traços desalinhados (esquerda) Traço de Energia (direita).
Fonte: **(LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J., 2007)**

Dessa forma, cada ponto dos traços de energia contém a informação de todas as amostras presentes no segmento correspondente do traço original, de maneira que estes sejam realinhados. Observa-se que os segmentos devem ter tamanho suficiente para englobar os possíveis deslocamentos entre os traços, e ao mesmo tempo, o tamanho não pode ser excessivamente grande. Segmentos muito grandes devem

armazenar muita informação, ocasionando em ruído que prejudica os ataques. Como esta estratégia consiste em realizar uma subamostragem nos traços do consumo, se os segmentos forem muito grandes, a taxa de amostragem desse processo pode cair abaixo da frequência de amostragem de *Nyquist*, resultando em *aliasing*, conforme revisado previamente. Isso limita o método a aplicação sobre traços com pequenos deslocamentos temporais entre si, conforme observado no histograma da Figura 15, onde o deslocamento máximo entre picos dista de 15 amostras entre os traços.

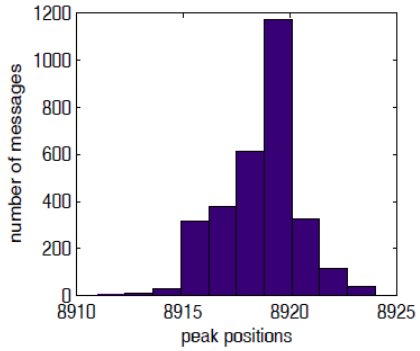

Figura 15 – Histograma de posição dos picos.
Fonte: **(LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J., 2007)**

Neste trabalho os autores não analisam o impacto do cálculo da energia com diferentes tamanhos de segmentos no desempenho do ataque DPA, deixando uma lacuna em aberto que é explorado nesta dissertação.

(TIAN, Q.; HUSS, S. A., 2012) propõem um método de alinhamento temporal dos traços de potência baseado na detecção dos picos das rodadas presentes nestes traços. Porém, como já observado anteriormente, a variação da frequência de relógio dos traços causa não somente um desalinhamento temporal ou horizontal, mas também um desalinhamento em amplitude, o que prejudica a detecção a etapa de detecção dos picos.

Portanto, os autores desenvolveram um método de detecção dos picos das rodadas do algoritmo criptográfico que contorna esse problema. O método foi chamado de detecção de pico em altas frequências, pois o deslocamento é função da frequência. O algoritmo de detecção de picos é baseado na inclinação dos mesmos, não usando mais um limiar de amplitude. Isto torna possível a detectar a posição dos picos em traços com qualquer frequência de relógio. Esse método é dividido em três algoritmos: o primeiro armazena uma forma de onda que informa as subidas e descidas dos traços. O segundo detecta as regiões de pico e não-pico. E o terceiro

detecta o ponto inicial dos picos.

Quando traços desalinhados com altas frequências de relógio são submetidos aos ataques DPA, não se consegue encontrar a chave criptográfica. O que, segundo os autores, evidencia a obrigatoriedade do uso de etapas de pré-processamento.

Este método tem como desvantagem a necessidade de buscar-se empiricamente conjuntos de parâmetros, adequados para conseguir-se a detecção dos picos com precisão.

Em (HODGERS, P.; HANLEY, N.; O'NEILL, M., 2013) evidencia a dificuldade de se identificar os pontos de interesse aos ataques DEMA/DPA nos traços adquiridos, pois os circuitos que realizam as tarefas de encriptação/decriptação são apenas uma parte dos dispositivos em que estão inseridos, sendo seu consumo sobreposto ao de outras partes do circuito que realizam outras atividades. Além disso, outra dificuldade encontrada se dá em circuitos operando com uma frequência de relógio elevada. Nestes casos, em função das capacitâncias inerentes a concepção dos circuitos, como vimos anteriormente, o decaimento da corrente é parcial, tornando a identificação das rodadas mais difícil.

Para contornar esses problemas, (HODGERS, P.; HANLEY, N.; O'NEILL, M., 2013) utilizaram um detector de fase, ou detector sensível a fase (em inglês, *Phase-Sensitive Detector* - PSD), com o intuito de gerar uma saída filtrada e ajudar no realinhamento dos traços.

O detector sensível a fase proposto por Hodgers et al., baseia-se na Série de Fourier e suas propriedades. Mais especificamente no fato de que se multiplicarmos duas senóides com frequência e fase iguais, estará presente no resultado um valor constante, enquanto que quando as frequências e/ou fases são diferentes, a média do resultado é igual a zero.

Assim, pode-se filtrar um sinal imerso em ruído, através da multiplicação do sinal original, contendo ruído, por um sinal de referência, com a mesma frequência e fase do sinal que se deseja recuperar, acentuando os picos das rodadas do algoritmo criptográfico. Com isso, é possível definir-se um limiar para que os picos da assinatura alvo seja detectada.

Podemos perceber que a frequência de referência deve ser igual à frequência de relógio nas aplicações de ataques DPA/DEMA. Ou seja, a frequência de relógio do sistema criptográfico em questão deve ser conhecida. Com isto, chegamos à conclusão que este método não se aplica a sistemas com contramedida de variação

da frequência de relógio. Além de necessitar da definição empírica de um limiar.

Denis (RÉAL, D.; CANOVAS, C., 2008) observaram que variações na amplitude dos traços são causadas não somente pela inserção de falhas, mas também pela a variação da frequência do sinal de relógio. Assim, uma variação na frequência altera os traços de potência, tanto no domínio do tempo quanto na amplitude. Neste sentido, os autores propuseram uma solução para este problema baseada em dois algoritmos, um para correção estática e outro para correção dinâmica dos traços para potencializar ataques em arquiteturas de *hardware* protegidas por este tipo de contramedida.

Num primeiro instante é feita uma correção estática do consumo induzido pelas alterações no relógio do circuito integrado, utilizando para isto, um método de extração dos valores máximos de cada pico de potência. É sabido que devido à frequência aleatória do relógio interno do circuito, o consumo de energia estático é afetado pelo atraso no tempo dos ciclos de relógio. Assim sendo, essas variações mudam completamente o valor máximo dos picos dos traços, enfraquecendo os ataques DPA. É feita uma análise prévia do comportamento do consumo do *hardware* atacado executando o algoritmo criptográfico *DES,* alvo do ataque neste caso. Com isto, concluiu-se que a potência estática é uma função linear da frequência de operação do circuito.

Os autores observam que as variações dos valores dos picos máximos devido as alterações na frequência é mais do que duas vezes a variação resultante da modificação dos operandos no algoritmo do hardware *DES* para uma frequência fixa. Com isso, chegaram à conclusão de que para obter sucesso em um ataque DPA nessa implementação de *DES* com deslocamento no relógio e extração do pico é necessário um aumento no número de curvas comparado ao mesmo processamento com uma frequência fixa. Além disso, foi verificado que com um número reduzido de 64 textos claros, a análise DPA de um bit realizada com frequência aleatória e extração dos picos é falha. Já com o algoritmo por proposto os ataques são executados com sucesso.

Foi comprovado que uma correção estática a fim de levar em conta as variações da frequência do relógio interno permite montar o ataque DPA com sucesso. Porém, é necessário que se conheça os pontos inicial e final da assinatura dos traços, além da frequência de relógio dos mesmos. O que pontua como uma limitação do método proposto.

Em (TIAN, Q.; HUSS, S. A., 2012) desenvolveram uma técnica de alinhamento vertical para corrigir o deslocamento de amplitude entre os traços, conforme visto em (RÉAL, D.; CANOVAS, C., 2008). Para tal alinhamento, primeiramente calcularam a diferença de altura dos picos antes do pré-processamento vertical, a qual indica a diferença entre o máximo e o mínimo valor de potência em todos os traços. Depois, é realizado um deslizamento dentro do intervalo de amplitudes, com uma janela de duas vezes a diferença calculada no passo anterior. Enquanto o deslizamento é realizado, a distância Euclidiana em relação a um modelo é calculada. O valor mínimo da distância indica que o traço horizontalmente alinhado com um certo *offset* no domínio da amplitude está alinhado com o modelo. O procedimento é repetido para todos os traços.

Como foi possível observar, este método tem o inconveniente de necessitar de um modelo a ser utilizado como referência. Neste trabalho, os autores não discorrem sobre tal modelo.

Já Youssef et al. em (SOUISSI, Y.; GUILLEY, S.; BHASIN, S.; DANGER, J., 2011) propõem um fluxo de avaliação da robustez da segurança de sistemas embarcados contra-ataques a canais laterais - SCA, composto de cinco fases distintas: caracterização, simulação, aquisição, pré-processamento e análise.

A fase de caracterização tem como base a documentação do sistema, provida pelo fornecedor. Nesta fase, são coletadas informações sobre como acessar o dispositivo, o tipo de contramedidas implementadas e a escolha da estratégia na análise. Na fase de aquisição, são medidos os traços de potência ou de emissão eletromagnética. Já na etapa de pré-processamento o avaliador deve utilizar técnicas para contornar dois problemas, que são o desalinhamento dos traços e o ruído excessivo presente nos mesmos. A simulação é outra fase que depende da documentação do dispositivo, pois é necessário predizer o comportamento do sistema criptográfico através de *software*, representando componentes reais através de seus modelos elétricos. Finalmente, a avaliação pode ser vista como uma ferramenta que visa dar o máximo de detalhes possível sobre o real nível de segurança presente no dispositivo criptográfico. Para isso, hoje em dia os avaliadores dispõem de muitas métricas para testar a performance dos ataques aos canais laterais. Essas métricas podem ser divididas em duas classes: a primeira classe engloba as métricas que visam medir a eficiência de um ataque em termos do número de traços necessários para descobrir a chave secreta. E a segunda, visa quantificar a informação vazada.

Este trabalho se propõe a apresentar uma estrutura genérica de um fluxo de ataques a canais laterais, sem aprofundar sua análise. Assim, sua análise torna-se superficial, não abordando nem ao menos a maneira como os autores chegaram ao fluxo final e quais tipos de alterações o mesmo pode sofrer.

Na Tabela 4 é apresentado um comparativo entre as contramedidas, anteriormente revisadas.

Tabela 4 - Comparativo entre propostas de alinhamento

Trabalho	Ferramenta	Contribuição	Limitação
[Le et al. 2007]	Cálculo da Energia	Os traços de energia contém muito menos pontos do que os originais, reduzindo muito o tempo de execução dos ataques	Os autores não discorrem sobre o tamanho dos segmentos no cálculo da energia. Não se aplica para traços com grandes deslocamentos
[Tian e Huss, 2012]	Slope Analysis	Alinha os traços tanto horizontalmente, quanto verticalmente	Tem a necessidade da definição de parâmetros empíricos para efetuar a detecção dos picos com precisão
[Hodgers, P. et al. 2013]	*Phase-Sensitive Detector*	Detecta os picos das rodadas mesmo em altas frequências.	A frequência de relógio deve ser conhecida, não sendo possível aplicar o método para sistemas com variação de frequência. Definição empírica de um limiar
[Réal et al. 2008]	Correção estática e dinâmica	Realiza o alinhamento horizontal e vertical dos traços	Necessita do conhecimento da frequência de relógio e dos pontos inicial e final da assinatura alvo
[Tian e Huss, 2012]	*Vertical Matching*	Realiza o alinhamento horizontal e vertical dos traços	Necessita de um modelo para referência. Os autores não discorrem sobre tal modelo.
[Youssef et al. 2011]	Fluxo comum para ataques	Desenvolvem um fluxo para avaliar a robustez de dispositivos criptográficos frente aos ataques por canais laterais	Análise superficial, não informando como foi obtido o fluxo e se o mesmo pode sofrer alterações
Trabalho Proposto	Fluxo Proposto Baseado no Cálculo da Energia	Redução na quantidade de traços para realizar o ataque DPA/DEMA	Necessidade da definição de parâmetros para a etapa de extração

Esta dissertação propõe um fluxo de ataques DPA/DEMA, cujas contribuições são relativas justamente às etapas de pré-processamento para realinhar os traços do consumo. A proposta apresentada tem por base o cálculo da energia dos traços (LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J., 2007), além de uma etapa para a extração da assinatura alvo dos ataques e uma etapa de subamostragem. Este fluxo tem como limitação a necessidade da definição de alguns parâmetros, através de observação, para a extração da assinatura alvo dos traços.

5 FLUXO PROPOSTO

Este Capítulo apresenta o fluxo de pré-processamento proposto para realizar ataques DPA/DEMA visando remover a ação de contramedidas que inserem distúrbios temporais por meio de variação da frequência de relógio e inserção de atrasos aleatórios durante a execução de algoritmos de criptografia. O fluxo proposto surge como uma alternativa aos trabalhos relacionados permitindo o uso completo dos traços de potência ou de radiação eletromagnética e realizando o alinhamento com base na geração da energia dos traços.

Conforme a revisão no Capítulo 2, um ataque DPA/DEMA pode ser dividido em 5 etapas, como segue:

I. Medição dos traços de potência ou radiação eletromagnética;
II. Etapa de pré-processamento de sinais (Etapa onde concentra-se este trabalho);
III. Separação de traços em função de um valor intermediário, um modelo de potência e todas as possíveis hipóteses de chaves;
IV. Média dos traços e aplicação da diferença das médias;
V. Determinação da hipótese de chave correta de acordo com o pico de potência.

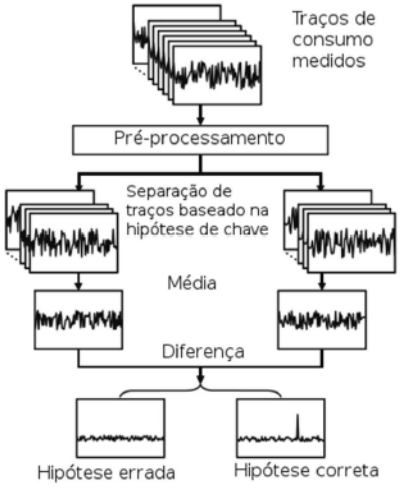

Figura 16 – Fluxo dos ataques DPA/DEMA.
Fonte: (SOARES, R. I., 2010)

Conforme visto na Figura 16, este trabalho concentra-se justamente em propor uma série de etapas de pré-processamento ao fluxo de ataque para realinhar os traços de potência ou radiação eletromagnética com o intuito de neutralizar

contramedidas baseadas na inserção de atrasos aleatórios e/ou variação da frequência de relógio, e avaliar a vulnerabilidade do dispositivo atacado.

5.1 Descrição das etapas do Fluxo

O fluxo de ataque proposto concentra esforços na etapa de pré-processamento dos traços. As demais etapas já se encontram disponíveis e validadas tal como realizado por (SOARES, R. I., 2010). Uma vez que os traços de potência, ou de radiação eletromagnética tenham sido adquiridos e encontrem-se disponíveis, as seguintes etapas de pré-processamento podem ser aplicadas:

I. Definição de parâmetros para extração da assinatura alvo dos traços;
II. Execução da extração da assinatura de todos os traços do conjunto;
III. Subamostragem das assinaturas para normalização, filtragem e pré-alinhamento;
IV. Cálculo da energia das assinaturas subamostradas, como etapa final de alinhamento;
V. Ataque DPA/DEMA.

A Figura 17 ilustra as etapas de pré-processamento do fluxo proposto neste trabalho:

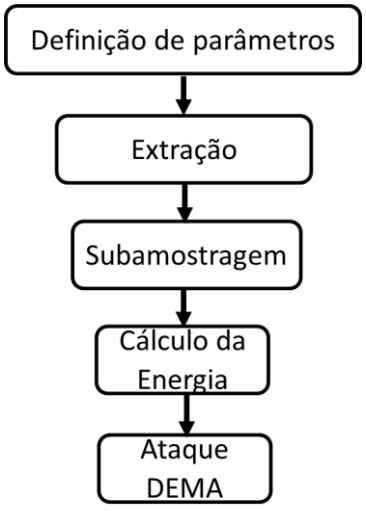

Figura 17 – Etapas de pré-processamento propostas.
Fonte: Própria.

Definição de parâmetros: são definidos dois parâmetros, um limiar que define um valor de amplitude que diferencie assinatura e ruído para ser aplicado durante a varredura do traço a fim de extrair a assinatura alvo (Etapa Extração). Nesta etapa,

também é definido o ponto inicial no traço para começar o processo de varredura.

Neste passo, uma transformação simples é aplicada a cada amostra dos traços a fim de aumentar a amplitude da assinatura de execução do algoritmo em relação ao ruído presente nas demais partes do traço. Isto favorece a definição de um valor de amplitude que seja o limiar entre ruído e a assinatura do algoritmo. Esta transformação tem o objetivo de aumentar a relação sinal-ruído (SNR) dos traços, e consiste em elevar cada ponto do traço a terceira potência. Esta potência foi escolhida, pois ressalta suficientemente a potência do traço em relação ao ruído, mantendo valores positivos e negativos das amostras originais. Em seguida, deve-se observar uma pequena quantidade de traços a fim de definir-se os parâmetros para a etapa de extração realizada a seguir. Aqui, é necessário que o atacante tenha um conhecimento básico sobre o comportamento do algoritmo de criptografia, para identificar quantas rodadas o mesmo possui; e do *hardware* a ser atacado, pois é possível que, como proposto em (SOARES, R. I., 2010), haja paralelismo na execução do algoritmo. Tomando-se como exemplo uma arquitetura executando o algoritmo de criptografia DES, espera-se observar uma assinatura no traço revelando 16 rodadas de processamento, conforme revisado na Seção 2.1. Além disso, a exemplo das arquiteturas GALS *pipeline*, onde a execução do algoritmo ocorre em diferentes ilhas de processamento, é preciso saber se existem e quantas ilhas estão sendo utilizadas para que se identifique o número de rodadas de processamento em cada ilha. Por exemplo, considerando-se a configuração com duas ilhas de processamento, GALS2, sabe-se que cada ilha deve executar 8 rodadas de processamento do algoritmo.

A Figura 18 ilustra um exemplo de traço de radiação eletromagnética do algoritmo DES, referente a sua execução em uma arquitetura GALS2, ou seja, 8 rodadas de processamento por ilha. Na parte inferior da Figura 18, vemos o traço após a transformação descrita anteriormente.

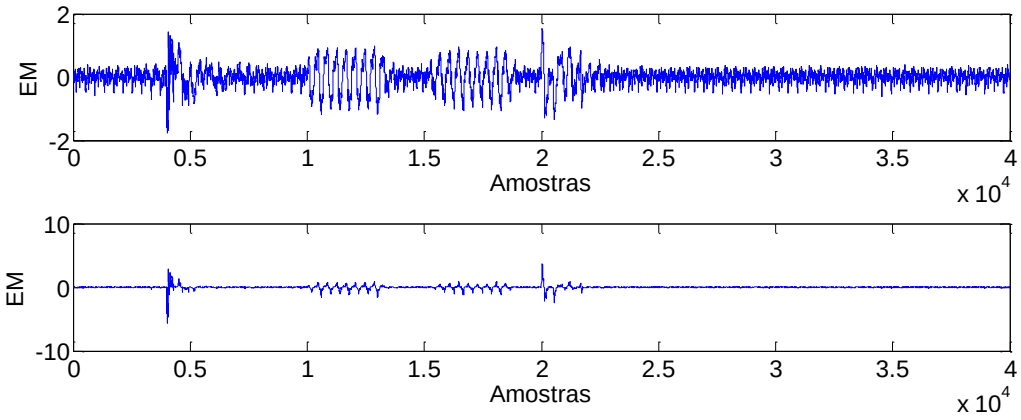

Figura 18 – Traço de EM (superior) Traço de EM após transformação (inferior).
Fonte: Própria.

Como é possível ver na Figura 19, no início do traço existe um ruído de amplitude elevada. Este ruído é causado pelo sinal de *trigger* gerado pela arquitetura para sincroniza-la com o osciloscópio, marcando o início da aquisição dos dados. Ao se analisar os traços, observa-se que este ruído inicial ocorre no mesmo instante de tempo em todos os traços. Assim, esta observação permite determinar um parâmetro importante para a etapa de extração, a definição do ponto inicial de varredura do traço a fim de evitar o ruído do *trigger* e que permita encontrar a assinatura alvo para sua extração. A Figura 19 apresenta detalhes do traço transformado e suas características.

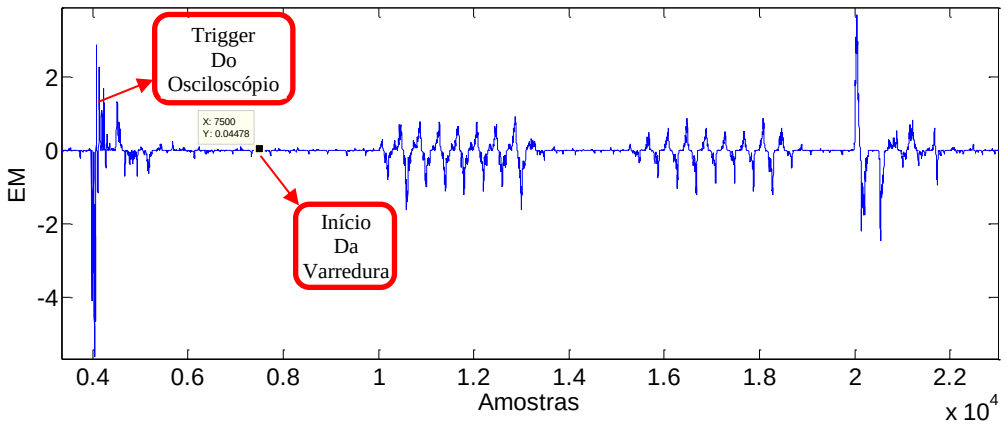

Figura 19 – Características dos Traços.
Fonte: Própria.

Extração: Com base nos parâmetros encontrados na etapa anterior, é possível realizar a extração automática da assinatura alvo em todos os traços de um determinado conjunto. Para isto, em cada traço, é feita uma varredura em todo seu comprimento, começando pelo ponto inicial estabelecido na etapa anterior. Fazendo-

se uma comparação do valor de cada amostra do traço corrente com o limiar previamente definido, conforme Figura 20, é possível encontrar o início da assinatura alvo. A partir do início da assinatura, o algoritmo busca a amostra de amplitude zero imediatamente à esquerda do início da assinatura. Ao encontrá-la esta amostra será definida como o ponto inicial de extração (*Ponto*$_{Inicial}$).

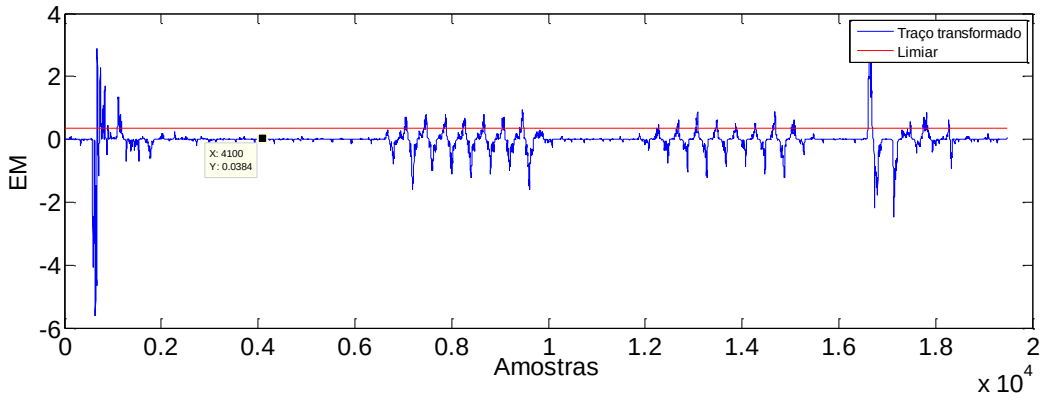

Figura 20 – Comparação dos pontos do Traço com o limiar.
Fonte: Própria.

Com esta informação, conhecendo-se a faixa de frequências de relógio do conjunto de traços disponível e o período de amostragem do osciloscópio usado na aquisição dos mesmos, é estimado o ponto final da assinatura (*Ponto*$_{Final}$). Para isso é calculado o tamanho aproximado da assinatura com base no número de amostras contidas em *n* ciclos de relógio da menor frequência do conjunto, tal como mostrado na Equação (27).

$$Tam_{janela} = \frac{n.T_{relógio}}{T_{amostragem}} \quad (27)$$

Onde *n* é o número de ciclos, ou rodadas do algoritmo em questão. Por exemplo, para o DES, *n* = 16. Já para o DES implementado em GALS2, *n* = 8. $T_{relógio}$ é o período da menor frequência de relógio do conjunto, ou seja, o maior período dentro do conjunto de traços, e $T_{amostragem}$ é o período de amostragem do osciloscópio utilizado na etapa de aquisição dos traços.

Este pré-recorte define uma janela retangular sobre a qual é aplicada a FFT a fim de descobrir-se a frequência de relógio do traço em questão. O uso de janela retangular não traz prejuízos a qualidade do resultado da FFT, pois o pico do lóbulo principal mostrou-se suficientemente maior, aproximadamente 4 vezes, em relação ao

pico do primeiro lóbulo secundário para os traços observados, conforme observa-se na Figura 21. Além disso, experimentos com traços de frequências conhecidas foram realizados, confirmando a eficiência do método.

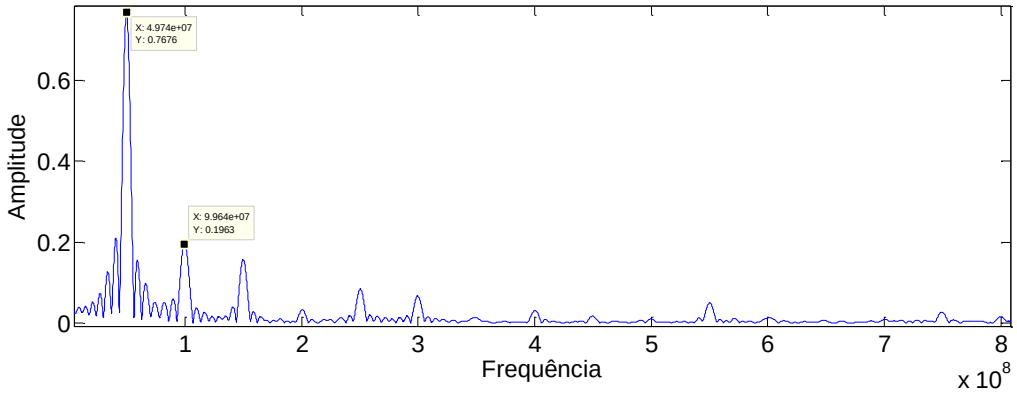

Figura 21 – FFT do Traço pré-recortado.
Fonte: Própria.

De posse da frequência de relógio do traço, pode-se encontrar o ponto final da assinatura alvo para sua extração através da Equação (28).

$$Ponto_{final} = Ponto_{inicial} + \frac{n.T_{relógio}}{T_{amostragem}} \quad (28)$$

onde *n* é o número de ciclos, ou rodadas do algoritmo alvo, sendo *n* dependente da sua implementação. O ponto inicial da extração $(Ponto_{inicial})$ já encontrado anteriormente, o período de amostragem do osciloscópio $(T_{amostragem})$ e $T_{relógio}$ calculado através da FFT. Este processo produz como resultado a assinatura de execução do algoritmo alvo do ataque, tal como mostrado na Figura 22. Na curva inferior da Figura 22 observa-se a assinatura das oito rodadas de execução da primeira ilha da arquitetura GALS2, mostrada como exemplo.

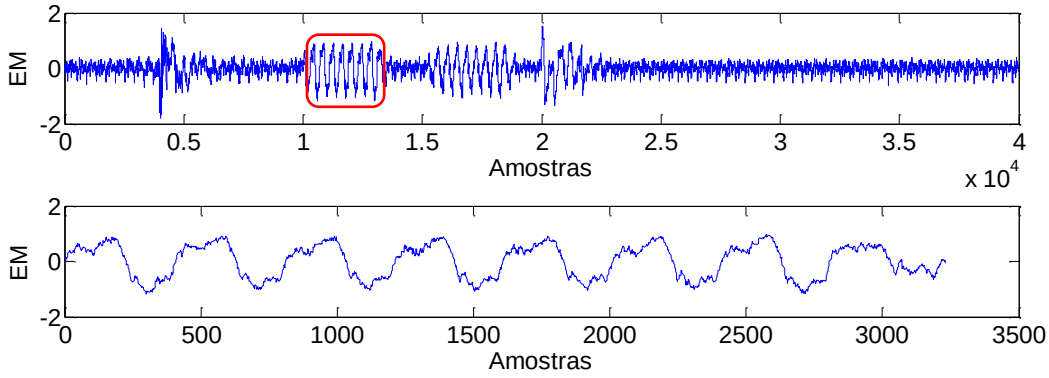

Figura 22 – Traço de EM (superior) e Assinatura alvo extraída (inferior).
Fonte: Própria.

Ao final desta etapa, obtém-se como resultado um conjunto com a assinatura alvo de todos os traços originais.

Subamostragem: Devido as diferentes frequências de relógio dos traços impostas pela ação da contramedida, as assinaturas extraídas na etapa anterior possuem tamanhos diferentes e por consequência quantidades de amostras diferentes. Para que o ataque DPA/DEMA possa ser realizado, os traços devem ter o mesmo tamanho.

Tanto os traços de potência, quanto os de radiação eletromagnética, normalmente são superamostrados pelos equipamentos usados durante a etapa de aquisição. Nas revisões e experimentos realizados por (LODER, L. L., 2014) mostram que traços superamostrados podem ser subamostrados até um dado limite sem perda significativa de informações. Assim, sem prejuízo, as assinaturas resultantes da etapa anterior são subamostradas de maneira que todas tenham ao final o mesmo tamanho em quantidade de amostras. O tamanho das assinaturas é obtido durante a etapa de extração.

O processo de subamostragem é realizado por meio da função *resample* do MATLAB a qual implementa um filtro anti-aliasing, tal como revisado anteriormente. Conforme a Seção 3.2, reduz o esforço computacional para etapas posteriores devido a redução da quantidade de amostras em cada assinatura. Além disso, esta etapa produz um pré-alinhamento dos traços, conforme pode ser visto na Figura 23. Na parte superior da Figura 23 são apresentadas duas assinaturas com diferentes tamanhos devido as suas frequências de operação. Na parte inferior são mostradas as assinaturas com o mesmo número de amostras e praticamente alinhadas após a subamostragem.

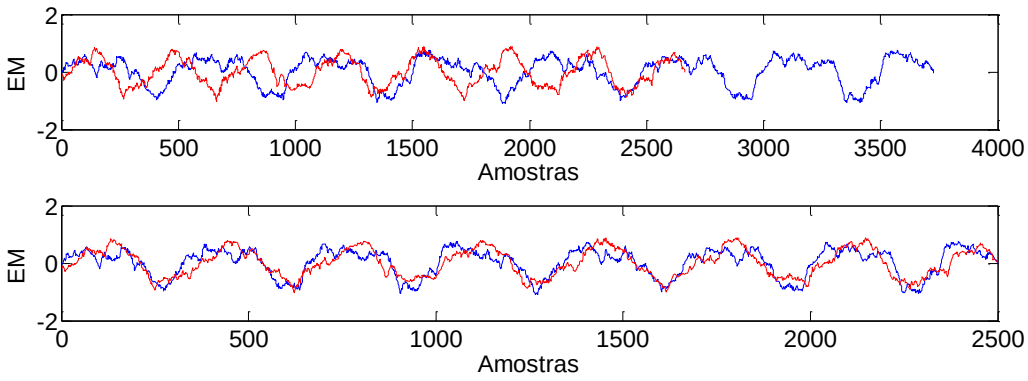

Figura 23 – Assinaturas com diferentes frequências (superior) e assinaturas subamostradas pré-alinhadas (inferior).
Fonte: Própria.

Cálculo da Energia dos Traços: Conforme revisado no Capítulo 4, (LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J., 2007) dividem os traços de potência em segmentos, e calculam a energia de cada um dos segmentos, condensando toda a informação presente no segmento em apenas um ponto. De acordo com (HAYKIN, S.; MOHER, M., 2011), a energia de um sinal de tempo discreto, descrito por $x[n]$, é calculado através da Equação (29):

$$Energy = \sum_{n=-\infty}^{+\infty} x^2[n] \tag{29}$$

A Equação (29) é aplicada às assinaturas dos traços já realinhadas por subamostragem, com base no algoritmo proposto por (LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J., 2007) e mostrado na Figura 24:

```
para i = 1 : m/L
    inicioSegmento = ((i-1)*L+1);
    fimSegmento = i*L;
    segmento = t(inicioSegmento: fimSegmento);
    soma = 0;
    para j = 1 : L
        soma = segmento(j)^2 + soma;
    fimpara
    EBS(i) = soma;
fimpara
```

Figura 24 – Algoritmo para o cálculo da energia dos traços
Fonte: **(LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J., 2007)**

onde *L* representa o tamanho do segmento, *m* o tamanho do sinal que se quer calcular a energia, *t* um vetor que armazena o sinal propriamente, e EBS (do inglês, *Energy*

Based Signal) consiste em um vetor que armazena as amostras de energia, sendo cada amostra o resultado do cálculo da energia de um segmento. A Figura 25 mostra a assinatura alvo de um traço (superior) e seu traço de energia correspondente (inferior).

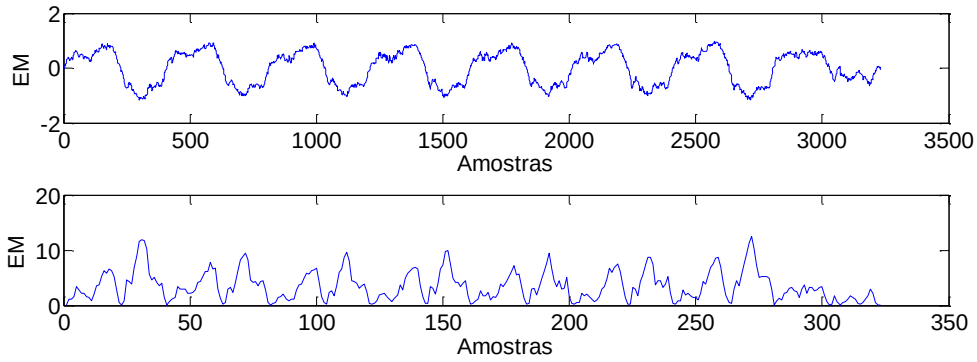

Figura 25 – Assinatura alvo (superior) e Traço de energia (inferior).
Fonte: Própria.

Nesta etapa são avaliados diferentes tamanhos de segmento para o cálculo da energia das assinaturas e seu impacto no desempenho dos ataques. Porém, intuitivamente os segmentos não podem ser muito grandes, pois muita informação precisaria ser condensada em um só ponto, o que ocasionaria perda de informações durante o processo, inviabilizando os ataques.

O método proposto por (LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J., 2007) limita-se a pequenos deslocamentos, conforme visto na Figura 15. Dentro do fluxo proposto nesta dissertação, as assinaturas provenientes da etapa de subamostragem já estão previamente alinhadas, fazendo com que esta limitação não represente um problema à proposta. Na Figura 26 um histograma mostra os deslocamentos existentes entre os traços utilizados como estudo de caso deste trabalho, através da qual podemos observar deslocamentos horizontais muito maiores do que os apresentados em (LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J., 2007).

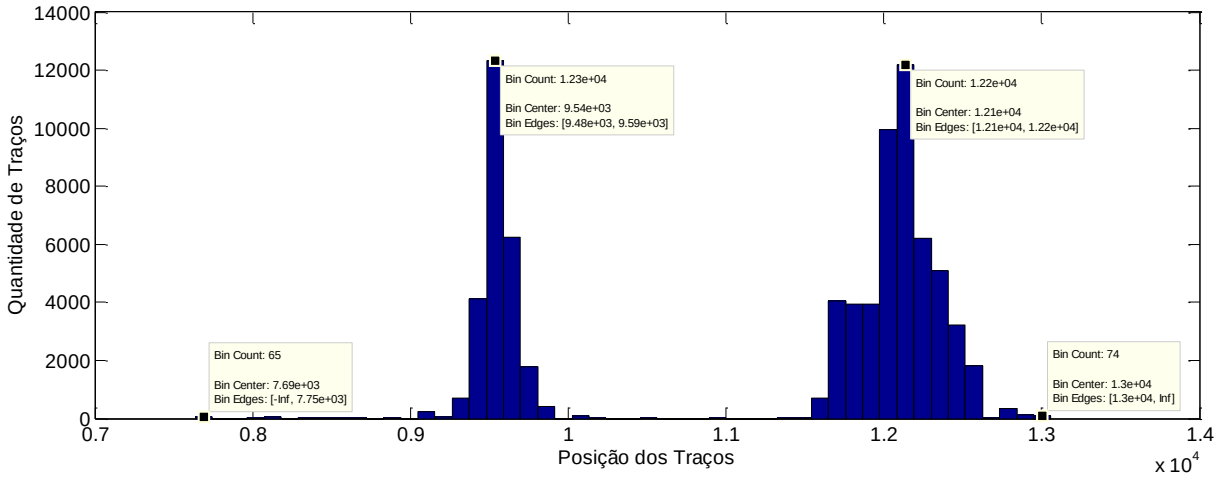

Figura 26 – Histograma da posição do pico da primeira rodada dos traços analisado.
Fonte: Própria.

Com o intuito de avaliar o tamanho dos segmentos no cálculo da energia são gerados conjuntos de traços de energia correspondentes a diferentes tamanhos de segmento em seus cálculos, a fim de avaliar relações que pudessem indicar qual o melhor tamanho de segmento para realizar o alinhamento dos traços.

Ataques DPA/DEMA: A última etapa do fluxo proposto é finalmente realizar os ataques DPA/DEMA nos diferentes conjuntos de traços de energia relativos aos diferentes tamanhos de segmento, resultantes do cálculo da energia sobre as assinaturas alvo, extraídas dos traços de radiação eletromagnéticas originais disponibilizados por (SOARES, R. I., 2010). Os ataques realizados nesta etapa, consistem nos *scripts* em MATLAB que implementam o ataque DPA/DEMA convencional (LOMNÉ, V.; MAURINE, P.; TORRES, L.; ROBERT,M. S.; CALAZANS, N., 2009), seguindo o fluxo de operações apresentado na Seção 2.3 do presente trabalho.

6 EXPERIMENTOS REALIZADOS

Neste Capítulo são apresentados experimentos realizados para validação e prova de conceito do fluxo de ataque proposto. O fluxo de ataque é aplicado sobre traços de radiação eletromagnética emitidos durante a execução das arquiteturas GALS *pipeline* contendo dois estágios conforme revisado previamente. Estas arquiteturas oferecem as seguintes opções de operação: (i) utilização de frequência de relógio única e *pipeline* vazio, ou seja, o texto claro é processado por todos os estágios da arquitetura para que um novo texto claro possa ser processado; (ii) utilização de frequências de relógio aleatórias gerados por osciladores locais escolhidas a cada novo processamento, e *pipeline* vazio; (iii) utilização de frequência de relógio global, e *pipeline* cheio, ou seja, os textos claros são processados sempre que houver demanda e que o estágio estiver ocioso; e (iv) com frequências de relógio locais e *pipeline* cheio.

Para a realização dos experimentos com o fluxo proposto usou-se dois conjuntos contendo 100.000 traços de radiação eletromagnética cada, obtidos por Soares et al. em (SOARES, R. I., 2010). Estes conjuntos referem-se à prototipação da arquitetura GALS *pipeline* no dispositivo FPGA Spartan3 XC3S200 da Xilinx (XILINX, INC., 2005). No primeiro conjunto, os traços foram medidos com o *hardware* configurado na opção (i) executando com frequência do relógio global de 50MHz. Já para o segundo conjunto de traços foi obtido a partir do *hardware* configurado na opção (ii) com frequências de relógio locais variando pseudo-aleatoriamente dentro de uma faixa de 38 a 60MHz.

O fluxo proposto neste trabalho foi aplicado a cada um destes conjuntos de traços. Os resultados obtidos são comparados com o fluxo proposto em (LODER, L. L., 2014). Assim é possível ver o desempenho do fluxo proposto em relação a técnicas de realinhamento conhecidas tais como POC, filtro de médias móveis, subamostragem e DTW.

6.1 Resultados Obtidos

Nesta Seção são apresentados aos resultados obtidos com a aplicação do fluxo proposto sobre as arquiteturas GALS *pipeline* operando em duas opções de operação.

6.2 Etapa de Extração

A etapa de extração do fluxo proposto neste trabalho foi aplicada no conjunto de traços de radiação eletromagnética disponibilizados por (SOARES, R. I., 2010), oriundos da execução do algoritmo DES na arquitetura GALS2 (SOARES, R. I., 2010). Lembrando que GALS2 é dotada de contramedidas de variação da frequência de relógio numa faixa de 38 a 60MHz e inserção de atrasos aleatórios.

O conjunto acima mencionado possui 100 mil traços. Porém, 19873 traços apresentaram anomalias e foram excluídos dos experimentos restando 80127 traços, sendo que nos primeiros experimentos o conjunto de traços foi dividido em 53358 traços com frequências de 38 a 42MHz e 26769 traços com frequências de relógio entre 55 e 60MHz. Em um segundo momento, o conjunto completo, contemplando os 80127 foi submetido à etapa de subamostragem.

Assim, para o grupo com frequências de 38 a 42MHz, conseguiu-se extrair corretamente a assinatura de 51585 traços, o que representa 96,68% dos traços. Já para o conjunto constituído de traços com frequências entre 55 e 60MHz, obteve-se sucesso na extração da assinatura alvo em 26005 traços, ou seja, 97,15% dos traços deste conjunto.

Realizando-se a extração nos 80127 traços, cujas frequências de relógio estão entre 38 e 60MHz, conforme mencionado, a etapa de extração foi bem sucedida em 77820 traços, totalizando 97,12% dos traços submetidos a esta etapa.

6.3 Arquitetura GALS2 com frequência de relógio global de 50MHz

Os experimentos iniciais foram realizados no conjunto de traços de radiação eletromagnética extraídos da execução das arquiteturas GALS *pipeline*, operando com duas ilhas de processamento sob um mesmo sinal de relógio, relógio global, com a frequência de operação de 50MHz e *pipeline* vazio. Neste conjunto de traços nenhuma contramedida é aplicada, podendo ser considerada como uma arquitetura vulnerável.

Para esta configuração, inicialmente é avaliado o impacto de diferentes tamanhos de segmentos para o cálculo da energia, na quantidade média de traços necessários para revelar as subchaves do algoritmo criptográfico. Por se tratar de uma arquitetura

sem contramedidas, a extração da assinatura pode ser realizada simplesmente pela definição do ponto inicial e final que são iguais em todos os traços.

Nesta etapa, cada traço é dividido em segmentos de mesmo tamanho para o cálculo da energia como método de alinhamento. Assim, um novo conjunto de traços é gerado para cada tamanho de segmento. Este procedimento reduz a quantidade de amostras no traço, concentrando a informação das amostras contidas no segmento, resultando em uma filtragem, além é claro de alinhar os traços de acordo com (LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J., 2007).

Na Tabela 5 são apresentados os resultados obtidos nestes experimentos, onde os valores apresentados correspondem a quantidade mínima de traços necessários para revelar a subchave de cada SBOX. Os valores N/C indicam que o ataque não convergiu para a chave correta. A última coluna da Tabela 5, apresenta a média de traços necessária para encontrar cada subchave do algoritmo criptográfico DES.

Tabela 5 – Resultados dos ataques DEMA com etapa de pré-processamento baseada no cálculo da energia dos traços

Tamanho Segmento	SBOX1	SBOX2	SBOX3	SBOX4	SBOX5	SBOX6	SBOX7	SBOX8	Média
Traços Originais	298	4250	2520	2163	50002	3996	3051	42154	2713,00
10	124	1406	2560	980	5283	870	1835	97260	1295,83
20	162	1489	2373	916	23625	931	1644	98529	1252,50
20 End Round 16 sum	923	1776	N/C	1237	19407	341	535	3795	962,40
20 End Round 16 cpa	825	1974	N/C	317	8596	183	100	3425	679,80
30	120	1134	1611	884	5115	943	1316	N/C	1001,33
40	120	341	1494	290	6677	622	539	N/C	567,67
50	120	553	1489	285	23658	588	540	73091	595,83
100	120	1193	932	295	54545	630	581	22719	625,17
150	104	409	1617	326	2447	622	729	N/C	634,50
200	120	1377	1000	301	2428	850	1057	65856	784,17
300	119	361	1130	716	2292	572	505	N/C	567,17
400	120	1175	993	327	2390	581	698	N/C	649,00

A Tabela 5 traz os resultados dos ataques DEMA realizados em conjuntos de traços de energia obtidos a partir de cálculos com segmentos de 10 tamanhos diferentes, segmentos com tamanhos variando desde 10 até 400 amostras. A justificativa para isso está no fato da arquitetura estar processando com um sinal de relógio de frequência de 50MHz. Como os traços foram amostrados a uma taxa de 20GSamples/s, conclui-se que um ciclo do relógio corresponde a um total de 400 amostras dos traços. Portanto, os segmentos podem conter informações de até um ciclo de relógio de execução.

É importante ressaltar que segundo (SOARES, R. I., 2010), as SBOXs 5 e 8 apresentaram uma disparidade em relação as demais, provavelmente por ruído e baixa captação de radiação proveniente destas SBOXs. Assim, neste trabalho será considerado como métrica a média de traços necessários para revelar as 6 subchaves criptográficas restantes, justamente em função dos problemas apresentados nas SBOXs 5 e 8.

Os resultados mostram que a quantidade média de traços é reduzida em relação aos traços sem nenhum pré-processamento (traços originais), até o segmento de energia de tamanho 40, que representa 10% do ciclo de relógio, onde em média são necessários aproximadamente 567 traços para se revelar as subchaves. Depois disso, há um aumento gradativo na média até atingir-se outro mínimo, novamente de 567 traços aproximadamente, para um segmento de tamanho 300, que representa 75% do ciclo de relógio de operação da arquitetura. Um valor médio superior é encontrado para o segmento de 400 amostras.

É interessante observar que segmentos grandes armazenam muita informação em uma única amostra, o que pode ser prejudicial ao ataque. Com base nestes resultados é possível verificar que no melhor caso, média de 567,17 traços, existe uma redução de 79,09% na quantidade média de traços necessária para se obter um ataque bem-sucedido, em relação aos traços originais.

A Figura 27 ilustra o comportamento de cada uma das SBOX em função do tamanho de segmento escolhido para o cálculo da energia dos traços. Novamente são excluídas aquelas SBOX que tiveram problemas de aquisição, conforme mencionado anteriormente. Observando a Figura 27, é interessante perceber que algumas SBOX são mais sensíveis à variação do tamanho do segmento, como por exemplo as SBOX2 e SBOX3, apresentando um comportamento oscilatório dentro de uma faixa de diferentes tamanhos de segmento. Já a SBOX1, possui uma característica bastante constante em relação ao número de traços, independentemente do tamanho de segmento escolhido para o cálculo da energia. Assim, um determinado tamanho de segmento pode apresentar melhores resultados para uma SBOX e para outra não, sendo portanto importante conhecermos a quantidade média de traços necessários para obter-se sucesso nos ataques para cada um dos segmentos utilizados.

Podemos ver também, analisando a Figura 27, que algumas SBOX necessitam de uma quantidade maior de traços para que seja revelada sua subchave, como por exemplo a SBOX3. Isso pode estar relacionado a aquisição dos traços, como ocorrido

com as SBOX5 e SBOX8, ou algum outro fator desconhecido. Outras SBOX tem seu segredo revelado mais facilmente, como a SBOX1 que exige uma quantidade menor de traços e possui uma dependência menor do tamanho do segmento.

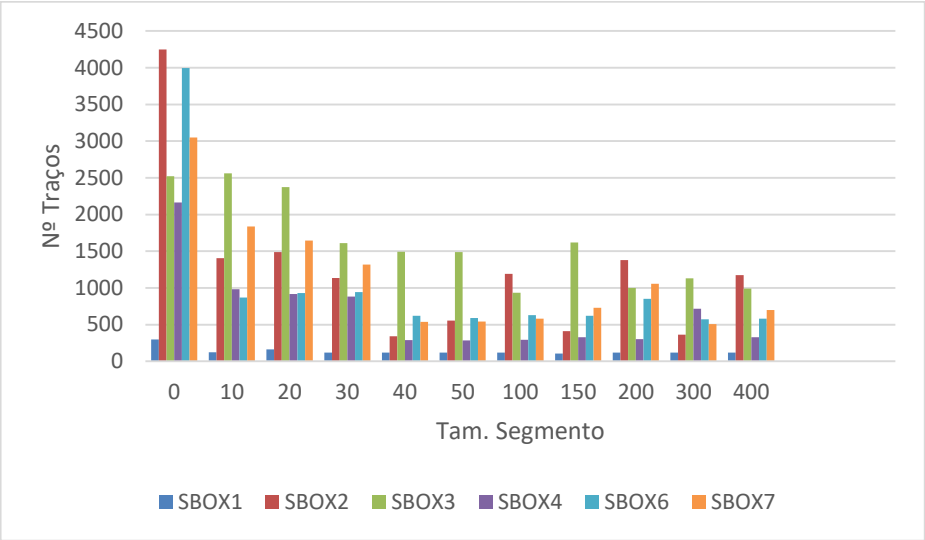

Figura 27 – Gráfico Nº de Traços vs. Tamanho dos segmentos – 50MHz.
Fonte: Própria.

Na Tabela 6, encontram-se os resultados obtidos por (LODER, L. L., 2014), relativos ao ataque DEMA, sem nenhuma etapa de pré-processamento sobre os traços. Como se pode observar nesta Tabela, com pouco mais de 50 mil traços é possível revelar a chave criptográfica completa usada no processamento, comprovando a vulnerabilidade desta configuração da arquitetura.

Tabela 6 – Resultados dos ataques DEMA sem pré-processamento (LODER, L. L., 2014)

sbox	obtained key	rank	key margin	first break	stabilization
01	24	01	57%	00221	00355
02	63	01	46%	00024	05723
03	33	01	24%	02547	05042
04	51	01	56%	00377	02366
05	09	01	09%	10873	28328
06	54	01	43%	01840	05074
07	53	01	50%	02424	03572
08	61	01	10%	49627	50752

Através da Tabela 6, podemos concluir que em média 3688,67 traços são necessários para descobrir cada uma das 6 subchaves do algoritmo criptográfico.

Como esta configuração não apresenta contramedidas, os únicos fatores que

podem dificultar os ataques, são os pequenos deslocamentos temporais entre os traços provocados pelo sistema de medição e os ruídos inerentes a plataforma de prototipação e aquisição dos traços. Com base nisto, (LODER, L. L., 2014) realizam novos testes sobre este conjunto de traços, aplicando uma etapa de pré-processamento, que consiste em uma filtragem nos traços para reduzir a quantidade de ruído presente, utilizando para isso, um filtro de médias móveis. A frequência de corte do filtro corresponde a 91MHz. Com isto, (LODER, L. L., 2014) obtiveram os seguintes resultados mostrados na Tabela 7.

Tabela 7 – Resultados dos ataques DEMA com etapa de filtragem dos traços (LODER, L. L., 2014)

sbox	obtained key	rank	key margin	first break	stabilization
01	24	01	60%	00032	00105
02	63	01	48%	00027	01279
03	33	01	22%	01613	01613
04	51	01	60%	00321	00844
05	09	01	27%	02896	03441
06	54	01	46%	00102	01127
07	53	01	50%	00628	00767
08	61	01	10%	00099	31163

Com a aplicação do filtro, (LODER, L. L., 2014) mostra que com pouco mais de 30 mil traços a chave criptográfica é revelada e em média são necessários 955,83 traços para encontrar cada subchave.

Comparando os resultados da Tabela 5, obtidos através do fluxo de ataques aqui proposto, com os resultados obtidos por (LODER, L. L., 2014) através da utilização de filtro de médias móveis (Tabela 7), percebemos uma redução de até 40,66% na quantidade de traços. Além disso, como os traços de energia possuem menos amostras que os originais, o esforço computacional para realizar o ataque é menor para a execução do ataque DEMA, conforme será apresentado posteriormente. Estes experimentos confirmam a efetividade do uso do cálculo da energia dos traços como uma técnica de alinhamento e servindo também para redução da quantidade média dos traços necessários para obter-se um ataque com sucesso.

A Figura 28 compara o desempenho em número de traços necessários para o sucesso do ataque dos segmentos de tamanho 40 e 300 com os resultados obtidos por (LODER, L. L., 2014), apresentados na Tabela 7. Os resultados são mostrados

por SBOX. É interessante notar que as curvas apresentam uma forma semelhante, que caracteriza as variações na quantidade de traços necessários de SBOX para SBOX. Neste caso é possível perceber que a quantidade de traços é menor para ambos os segmentos do que para (LODER, L. L., 2014) em todas as SBOX, exceto para a SBOX1. Isto confirma a eficiência do método aplicado.

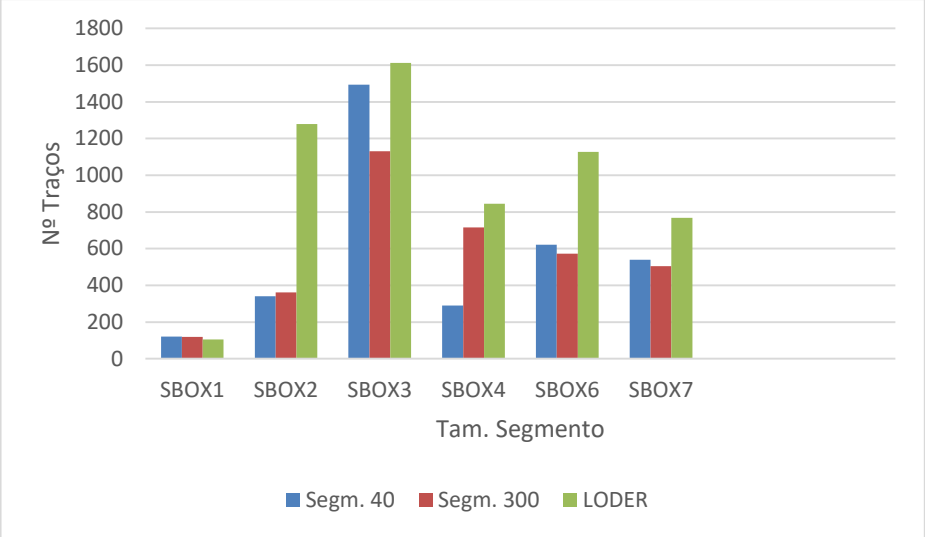

Figura 28 – Comparativo entre o fluxo proposto e (LODER, L. L., 2014) – 50MHz.
Fonte: Própria.

6.4 Arquitetura GALS2 com frequências de relógio locais de 38 a 60MHz

Nestes experimentos o fluxo proposto é usado para atacar as arquiteturas GALS *pipeline* operando com duas ilhas de processamento, com frequências de relógio pseudoaleatórias geradas localmente em cada ilha. As frequências dos sinais de relógio locais variam entre 38 a 60 MHz tal como proposto em (SOARES, R. I., 2010). Um conjunto de 100 mil traços de radiação eletromagnética também foi obtido a partir de medições durante a execução do algoritmo criptográfico DES.

O fluxo proposto é usado para atacar o conjunto inteiro de traços, cobrindo toda a faixa de frequências possíveis. Entretanto, para fins de comparação com o fluxo de (LODER, L. L., 2014), foram realizados experimentos dividindo-se os traços em dois grupos distintos, tal como proposto por em (LODER, L. L., 2014). Nesta divisão, os traços devem ter frequências próximas devidos as limitações de POC, utilizado para realinhar os traços. Deste modo, o fluxo de ataque proposto é aplicado em ambos os grupos, tanto no grupo frequências de relógio variando entre 38 até 42MHz, quanto

no grupo contendo frequências de 55 a 60MHz. O processo de divisão dos traços em grupos por frequências foi realizado por (LODER, L. L., 2014). É interessante observar que a limitação de frequências imposta ao trabalho desenvolvido por (LODER, L. L., 2014) pode acarretar em uma quantidade de traços insuficiente para garantir o sucesso do ataque.

Desse modo, os primeiros experimentos com esta configuração das arquiteturas GALS *pipeline* são realizados sobre o Grupo 1 contendo traços com frequências de relógio entre 38 a 42MHz, alinhados por POC em (LODER, L. L., 2014). Neste caso, também são realizados experimentos baseados no cálculo da energia para 10 diferentes tamanhos de segmentos, além dos traços originais, ou seja, apenas com o processamento de alinhamento por POC. Os resultados são mostrados na Tabela 8.

Tabela 8 – Resultados dos ataques DEMA com etapa de pré-processamento baseada no cálculo da energia dos traços de 38 a 42MHz.

Tamanho Segmento	SBOX1	SBOX2	SBOX3	SBOX4	SBOX5	SBOX6	SBOX7	SBOX8	Média
Traços Originais	5816	38515	28936	22791	N/C	53302	33840	N/C	30533,33
10	3200	13647	2509	13036	N/C	48336	17475	N/C	16367,17
20	784	13029	2294	12694	53302	53302	17313	N/C	16569,33
30	767	12896	2107	12969	53302	53302	11394	N/C	15572,50
40	764	13246	2085	12969	N/C	53213	18220	N/C	16749,50
50	765	14082	2085	12933	53302	48545	18409	N/C	16136,50
100	1035	34069	24749	14348	53302	26524	18550	N/C	19879,17
150	1035	13200	24734	15910	53302	19289	11382	N/C	14258,33
200	2167	14150	28550	18664	N/C	42600	12135	N/C	19711,00
300	3140	16791	32177	22252	N/C	45233	14571	N/C	22360,67
400	3214	16783	32197	22575	N/C	52942	15468	N/C	23863,17

O objetivo deste experimento é avaliar o impacto do cálculo da energia com diferentes tamanhos de segmentos sobre traços previamente alinhados. Os resultados demonstram no melhor caso, segmento de 150 amostras, uma redução na quantidade de traços de até 53% em relação aos traços originais.

Através do gráfico da Figura 29 pode-se claramente perceber que para o segmento com 150 amostras de tamanho a quantidade de traços é reduzida em relação aos segmentos com outros tamanhos. Também observa-se que a SBOX6 destoa das outras SBOX, sendo necessária uma quantidade muito superior de traços para que os dados ocultos sejam revelados. Para este conjunto de traços os diferentes segmentos apresentaram uma menor oscilação na quantidade de traços.

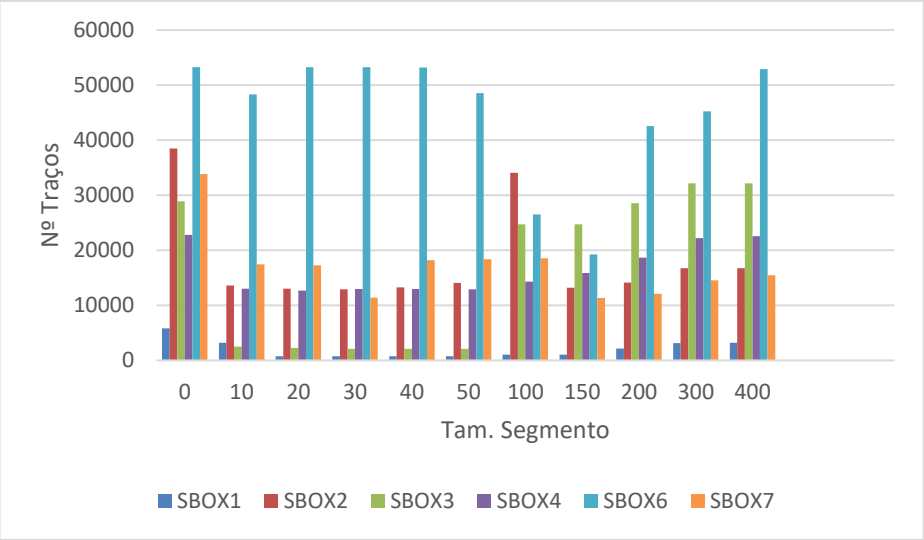

Figura 29 – Gráfico Nº de Traços vs. Tamanho dos segmentos – 38 a 42MHz.
Fonte: Própria.

Um novo experimento realiza a aplicação do fluxo completo proposto neste trabalho ao conjunto com traços de frequências de relógio de 38 a 42MHz. Primeiramente é realizada a extração da assinatura alvo de todos os traços do conjunto, gerando um novo conjunto com as assinaturas de interesse. Em seguida, as assinaturas são subamostradas, de modo que todas tenham o mesmo tamanho, ou seja, o mesmo número de amostras. Esse tamanho utilizado como referência deve ser menor do que o tamanho do menor traço, para que todos traços sejam subamostraços em relação a esta referência. Na última etapa de pré-processamento são gerados os traços de energia com diferentes tamanhos de segmentos. Em seguida os ataques foram realizados sobre todos os conjuntos de traços de energia gerados. A Tabela 9 apresenta os resultados obtidos com a execução do fluxo completo.

Tabela 9 – Resultados dos ataques DEMA com etapa de pré-processamento composta do fluxo completo proposto sobre os traços de 38 a 42MHz.

Tamanho Segmento	SBOX1	SBOX2	SBOX3	SBOX4	SBOX5	SBOX6	SBOX7	SBOX8	Média
Traços Originais	4479	35705	51815	13013	N/C	51626	15460	N/C	28683,00
10	2839	11608	32124	6625	45115	29259	8207	N/C	15110,33
20	2803	11611	28445	5571	40366	26548	6179	N/C	13526,17
30	2827	10266	26221	5533	36762	17338	10264	N/C	12074,83
40	2660	11449	32906	5387	40752	26807	6254	N/C	14243,83
50	2307	8746	37070	5530	35633	17336	10370	N/C	13559,83
100	2507	10402	23228	7058	36038	16163	14193	N/C	12258,50
150	354	2082	2408	780	36047	8234	1474	N/C	2555,33
200	1239	5849	28566	5533	34189	8574	6400	N/C	9360,17
300	193	1596	1795	1129	42228	4887	1259	N/C	1809,83
400	624	3735	25873	2812	33969	3473	5383	N/C	6983,33

Obtendo a frequência de relógio média deste conjunto de traços, ou seja, frequência de 40MHz, e considerando a frequência de amostragem na aquisição dos traços de 20GSamples/s é possível afirmar que 500 amostras estão contidas em um ciclo de relógio na frequência média.

A partir dos dados da Tabela 9, percebe-se que em média a menor quantidade de traços para recuperar as subchaves criptográficas, é de aproximadamente meio ciclo da frequência média deste grupo. Isto parece razoável, pois cada semiciclo da frequência média tem suas informações armazenadas em uma amostra de energia correspondente. Ao aumentar o tamanho do segmento, informações do próximo ciclo de relógio são adicionadas as informações do ciclo atual, prejudicando os ataques. Os dados da Tabela 9 demonstram uma redução na quantidade de traços de até 93,69% em relação aos traços sem o cálculo da energia, resultados contidos na primeira linha da Tabela, referenciados como Traços Originais.

Podemos ver, pela Figura 30, que os segmentos de tamanho igual a 150 e 300 tem os melhores resultados quanto ao número de traços necessários. Destaca-se a SBOX3 como a mais difícil de aplicar o ataque, devido a sua característica oscilatória para os segmentos de diferentes tamanhos e maior quantidade de traços necessária.

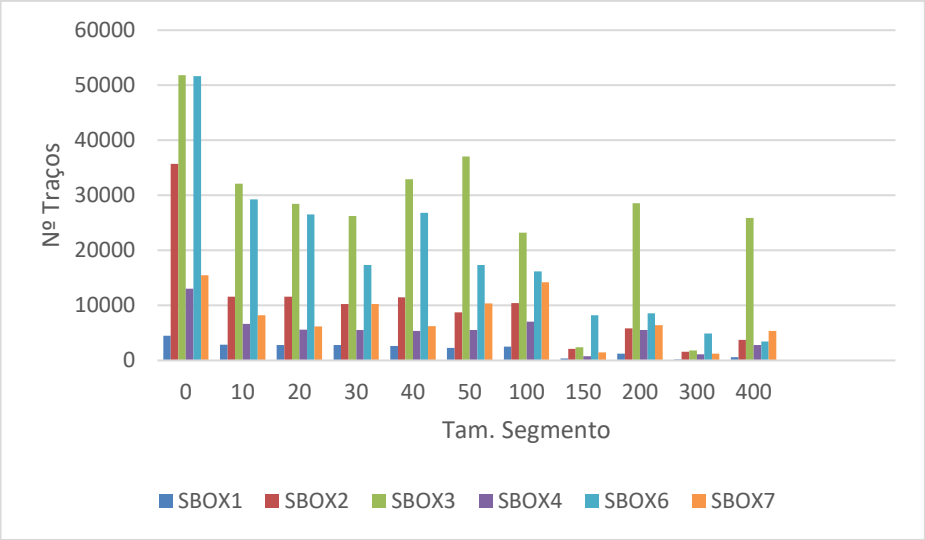

Figura 30 – Gráfico Nº de Traços vs. Tamanho dos segmentos – 38 a 42MHz – Fluxo Completo.
Fonte: Própria.

Em outro estudo de caso, o fluxo proposto foi aplicado aos traços do Grupo 2, contendo traços com frequências de relógio na faixa de 55 a 60MHz. Estes experimentos têm seus resultados apresentados na Tabela 10.

Tabela 10 – Resultados dos ataques DEMA com etapa de pré-processamento composta do fluxo completo proposto sobre os traços de 55 a 60MHz.

Tamanho Segmento	SBOX1	SBOX2	SBOX3	SBOX4	SBOX5	SBOX6	SBOX7	SBOX8	Média
Traços Originais	7528	21010	26006	16753	25430	17112	19600	N/C	18001,50
10	2781	12139	24808	5245	25812	6741	9832	N/C	10257,67
20	2824	12601	21481	5171	25803	6668	9890	N/C	9772,50
30	2200	10835	22783	5397	25815	6590	8985	N/C	9465,00
40	3002	10707	10484	5436	25471	6071	9737	N/C	7572,83
50	1713	8726	6539	6388	N/C	7749	5844	N/C	6159,83
100	1693	8869	4694	2460	25641	2926	4624	N/C	4211,00
150	898	1291	1772	3001	26007	1464	1296	18491	1620,33
200	1071	2691	2650	1207	18910	1460	2063	20622	1857,00
300	1128	6336	2506	3093	16730	1608	1025	N/C	2616,00
400	3541	13213	4668	4213	22655	1305	2128	N/C	4844,67

Os resultados mostram novamente que o melhor caso para os ataques DEMA ocorre no segmento de tamanho aproximado a meio ciclo da frequência de relógio média do grupo, neste caso a frequência média é 57,5MHz. Com a taxa de amostragem de 20G Samples/s, um ciclo de relógio na frequência média possui 347,83 amostras e por consequência 173,91 amostras em um semiciclo. Para este caso, a redução atinge até 91% na quantidade de traços, em relação aos traços originais, ou seja, os traços em que a etapa de cálculo da energia não foi realizada.

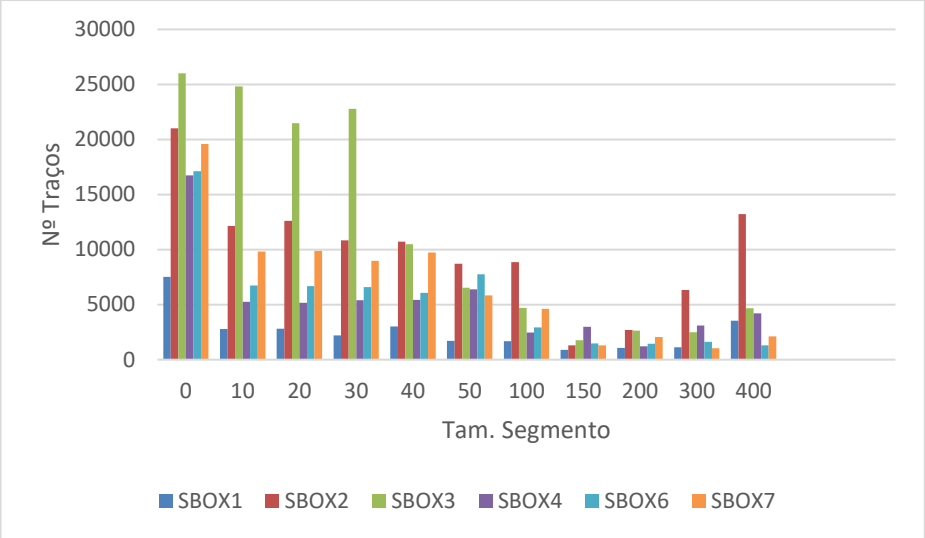

Figura 31 – Gráfico Nº de Traços vs. Tamanho dos segmentos – 55 a 60MHz.
Fonte: Própria.

A Figura 31 mostra que para o conjunto de traços com frequências de relógio variando de 55 a 60MHz tem-se uma convergência das SBOX para o segmento de tamanho 150, sendo este o segmento com a menor quantidade de traços necessários em média para revelar a subchave. Outra vez, tem se a SBOX3 necessitando de uma quantidade maior de traços para os segmentos de menor tamanho, convergindo juntamente com as outras SBOX conforme aumenta o tamanho dos segmentos.

Loder et al. em (LODER, L. L., 2014) realizaram experimentos sobre os dois conjuntos de traços supracitados, dentre os quais destacam-se os experimentos realizados com POC + filtragem e os que utilizam DTW. Experimentos realizados com subamostragem também foram levados em conta neste trabalho para questões de comparação.

A Tabela 11 mostra os resultados de experimentos realizados sem nenhuma etapa de pré-processamento adicionado ao fluxo de ataque (SSP), revelando a robustez desta arquitetura frente aos ataques DEMA, conforme constatado anteriormente por (SOARES, R. I., 2010). Também são mostrados os resultados aplicando-se somente o (Filtro) nos traços antes do ataque, aplicação somente do realinhamento dos traços por correlação de fase (POC), e finalmente os resultados do ataque, adicionado da etapa de filtragem e posterior alinhamento dos traços baseado no método POC (POC+Filtro). Na Tabela 11, #T significa a quantidade mínima de traços necessária

para revelar a subchave, e os Grupos 1 e 2, são respectivamente o conjunto de traços com frequências de 38 a 42MHz e o conjunto de traços de 55 a 60MHz, citados anteriormente.

Tabela 11 – Resultados dos ataques DEMA com etapa de filtragem e alinhamento por POC (LODER, L. L., 2014).

		SPP		Filtro		POC		POC + Filtro	
SBOX	**Grupo**	**#T**	**Rank**	**#T**	**Rank**	**#T**	**Rank**	**#T**	**Rank**
sbox1	1	–	40	—–	28	06943	01	01290	01
	2	–	29	—–	–	02513	01	01382	01
sbox2	1	–	21	—–	02	22130	01	12932	01
	2	–	28	—–	–	10468	01	02032	01
sbox3	1	–	31	—–	16	29752	01	18836	01
	2	–	15	—–	–	03798	01	01077	01
sbox4	1	–	08	—–	07	12211	01	04790	01
	2	–	31	—–	–	03510	01	02562	01
sbox5	1	–	61	—–	26	27238	01	21516	01
	2	–	12	—–	–	—–	02	16023	01
sbox6	1	–	25	—–	43	15987	01	15987	01
	2	–	37	—–	–	09202	01	02294	01
sbox7	1	–	32	—–	34	33511	01	13886	01
	2	–	38	—–	–	02187	01	02097	01
sbox8	1	–	05	—–	38	—–	56	—–	22
	2	–	22	—–	–	—–	02	13777	01

Como podemos ver na Tabela 11, os experimentos com aplicação apenas de filtragem não obtiveram sucesso na tentativa de revelar as subchaves criptográficas em nenhum dos dois grupos. Já os experimentos utilizando apenas POC como etapa de pré-processamento revelaram quase todas as subchaves. Neste caso, a quantidade de traços é reduzida quando são utilizadas as etapas de filtragem e alinhamento com POC.

Para os experimentos com POC apenas, a média de traços necessários para obter-se sucesso na tarefa de encontrar as subchaves do algoritmo criptográfico é igual a 20089 para o conjunto de 38 a 42MHz (Grupo 1) contra 14258,33 do fluxo proposto neste trabalho (Tabela 8) para o mesmo conjunto de traços. Isto resulta em uma redução de 29% na quantidade de traços em relação a (LODER, L. L., 2014).

Considerando-se agora, os melhores resultados obtidos por (LODER, L. L., 2014) para estes experimentos, ou seja, os resultados para POC+Filtro. A quantidade média mínima necessária de traços para encontrar-se as subchaves, é de 11286,83 para o Grupo 1. O fluxo de ataques desta dissertação apresenta para este grupo, uma quantidade de 1809,33 traços, conforme visto na Tabela 9. Assim, a redução na

quantidade de traços é de 83,96%, comparado a (LODER, L. L., 2014).

Já para o Grupo 2, (LODER, L. L., 2014) utilizando POC+Filtro necessita em média de 1907,33 para obter sucesso nos ataques. O trabalho aqui proposto revela a chave criptográfica com uma média de 1620,33 traços, resultando em uma redução de 15,05% na quantidade de traços.

No experimento em que o fluxo proposto completo é aplicado ao Grupo 1 dos traços, cujos resultados são apresentados na Tabela 9, é observada uma redução de 87,31% na quantidade de traços em relação aos experimentos que utilizaram alinhamento por POC antes do cálculo da energia conforme a Tabela 8, o que confirma a efetividade da extração das assinaturas alvo, proposta neste trabalho.

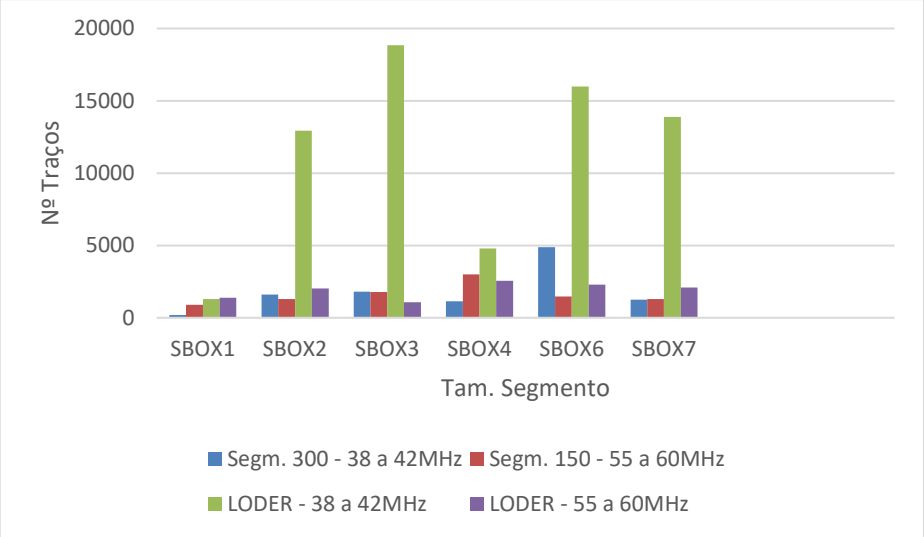

Figura 32 – Comparativo entre o fluxo proposto e (LODER, L. L., 2014) – 55 a 60MHz.
Fonte: Própria.

Como pudemos observar, existe uma disparidade maior na quantidade de traços entre o fluxo proposto e (LODER, L. L., 2014) no grupo de traços com frequências de 38 a 42MHz do que o de 55 a 60MHz. Isso está ilustrado na Figura 32, onde percebemos que as diferenças na quantidade de traços para o grupo de 55 a 60MHz são bastante difíceis de analisar, sendo importante utilizar a média como métrica, como feito anteriormente.

O fluxo de ataque proposto não possui limitações quanto a faixa de frequências de relógio dos traços de potência ou de radiação eletromagnética. Deste modo é possível realizar experimentos com o conjunto inteiro de traços gerados com a arquitetura

GALS2, e as contramedidas de atrasos aleatórios e variação de frequência de relógio, como descrito anteriormente. Já os métodos de alinhamento usados no fluxo proposto por (LODER, L. L., 2014) possuem restrições em relação frequência dos traços. Por este motivo, neste estudo de caso não são realizadas comparações entre o fluxo proposto e o fluxo proposto em (LODER, L. L., 2014). A Tabela 12 resume os resultados obtidos no estudo de caso com a aplicação do fluxo proposto no conjunto completo de traços, contendo traços com frequências entre 38 a 60 MHz.

Tabela 12 – Resultados dos ataques DEMA com etapa de pré-processamento composta do fluxo completo proposto sobre os traços de 38 a 60MHz

Tamanho Segmento	SBOX1	SBOX2	SBOX3	SBOX4	SBOX5	SBOX6	SBOX7	SBOX8	Média
Traços Originais	15026	41236	77711	18337	N/C	64164	33944	N/C	41736,33
10	1614	21835	20476	7867	67233	51147	9001	N/C	18656,67
20	1749	13944	17739	7903	68944	25545	9281	N/C	12693,50
30	1955	13189	9692	7155	63358	17607	8734	N/C	9722,00
40	1507	10412	35313	6385	67632	23764	9223	N/C	14434,00
50	4405	10893	11552	4955	68117	17241	6508	N/C	9259,00
100	3069	3791	8205	4198	72961	10477	5653	N/C	5898,83
150	849	4062	6565	3705	69226	3456	4858	N/C	3915,83
200	539	2220	5229	2860	N/C	3716	3073	64482	2939,50
300	1341	6694	2657	2960	N/C	7874	2475	77103	4000,17
400	1521	6715	1392	3243	21081	5756	6619	N/C	4207,67

Observando os dados da Tabela 12 pode-se notar uma redução de até 92,96% em média no número de traços necessários para ataques bem-sucedidos em relação aos Traços Originais mostrados na primeira linha da Tabela 12 (observando novamente que os traços originais são os traços que não foram submetidos apenas a etapa de cálculo da energia do fluxo proposto). Este resultado ocorre quando utilizamos traços de energia de 200 amostras de comprimento. Ao analisar a quantidade de amostras por ciclo da frequência de relógio média do conjunto, chega-se a um total de 408,16 amostras por ciclo. Novamente o melhor caso ocorre para o cálculo da energia com comprimento aproximado a meio ciclo de relógio da frequência média do conjunto. Percebe-se ainda, que para um total de traços disponíveis consegue-se encontrar a chave criptográfica, com uma média de 2939,5 traços para cada subchaves, desconsideradas as SBOXs 5 e 8, por apresentarem problemas na aquisição conforme mencionado anteriormente. Isto confirma a eficácia do fluxo proposto neste trabalho, para viabilizar ataques DPA/DEMA em arquiteturas dotadas de contramedidas de variação randômica da frequência de relógio e inserção de atrasos aleatórios.

A Figura 33 mostra os resultados apresentados na Tabela 12.

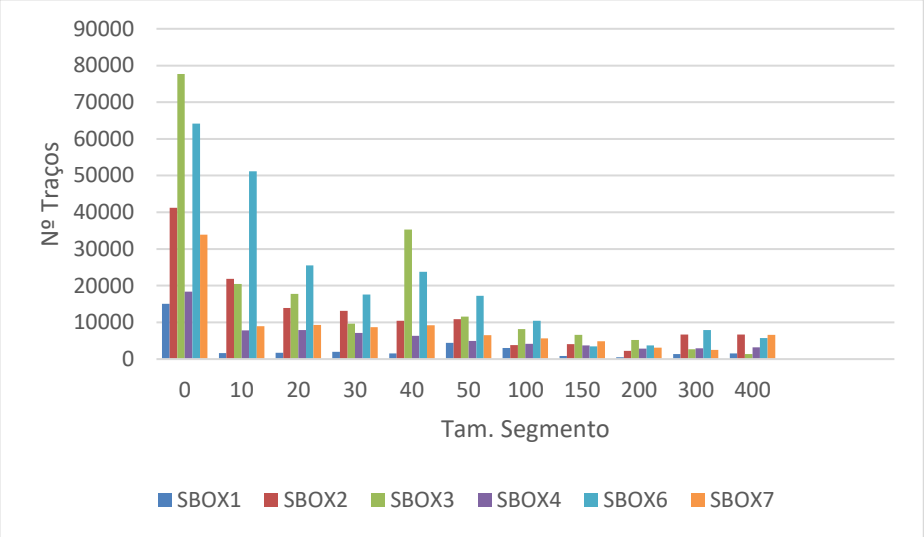

Figura 33 – Gráfico Nº de Traços vs. Tamanho dos segmentos – 38 a 60MHz.
Fonte: Própria.

6.5 Segmentos de Tamanho Variável

Em um novo estudo de caso são realizados experimentos aplicando o fluxo proposto sem a etapa de subamostragem, ou seja, realizando o cálculo de energia sobre as assinaturas extraídas dos traços originais, para o grupo com frequências de relógio entre 38 e 60MHz, tal como mostrado na Figura 34. Isso significa que as assinaturas extraídas permanecem com tamanhos diferentes, proporcionais as frequências de operação. Logo, o alinhamento deve ser realizado na integra pela etapa de cálculo de energia. Neste caso, para cada tamanho de segmento utilizado no estudo de caso anterior, observou-se a quantidade de amostras dos traços de energia resultantes. Assim, ao invés de utilizarmos um segmento de tamanho fixo para calcular a energia dos traços, definiu-se como parâmetro a quantidade de amostras que os traços de energia resultantes devem ter, de modo que o tamanho dos segmentos deve variar para cada traço.

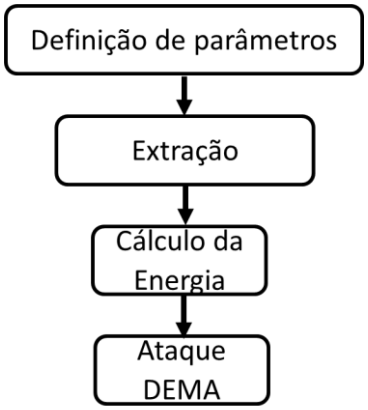

Figura 34 – Fluxo de ataque DEMA proposto sem a etapa de subamostragem.
Fonte: Própria.

Os resultados obtidos com estes experimentos são resumidos na Tabela 13:

Tabela 13 – Resultados dos ataques DEMA com etapa de pré-processamento composta do fluxo proposto sem a etapa de subamostragem.

Amostras Traço de Energia	SBOX1	SBOX2	SBOX3	SBOX4	SBOX5	SBOX6	SBOX7	SBOX8	Média
Traços Originais	15026	41236	77711	18337	N/C	64164	33944	N/C	41736,33
250	1968	17470	14595	9386	N/C	40594	12440	N/C	16075,50
125	2938	18914	26476	9893	67172	42886	17803	N/C	19818,33
83	4564	13184	13860	7598	69942	33400	10678	N/C	13880,67
62	1596	11279	55173	6387	77699	23268	15724	N/C	18904,50
50	4297	17573	24791	5086	68672	31944	8505	N/C	15366,00
25	3705	3766	11649	4594	58468	13192	5651	N/C	7092,83
16	400	2911	6271	3026	58082	3525	3844	N/C	3329,50
12	583	2255	4511	2973	63415	4699	4006	N/C	3171,17
8	1029	3580	1966	2347	62713	3359	3686	N/C	2661,17
6	1703	6150	2953	5083	43125	3424	9195	N/C	4751,33

Neste estudo de caso obtém-se uma redução na quantidade média de traços igual a 93,62% em relação aos traços originais, em que o cálculo da energia não foi realizado, e no melhor caso um aumento de 9,47% em relação ao melhor caso dos experimentos contendo a etapa de subamostragem no fluxo. Observa-se que os melhores casos com ou sem a etapa de subamostragem são praticamente os mesmos e em muitos casos sem a subamostragem foram iguais ou piores, o que nos leva a concluir, que o cálculo da energia com segmentos de tamanho fixo, pode ser considerado melhor que o uso de segmentos com tamanho variável. Também pode-se concluir que a perda de informações decorrente do processo de subamostragem é irrelevante, sendo a quantidade de ruído removida muito superior. Isso reafirma ser vantajoso o uso desta etapa. Além, do ganho em esforço computacional resultante da subamostragem, como já verificado.

6.6 Alinhamento Vertical

Conforme revisado, (RÉAL, D.; CANOVAS, C., 2008) verificaram que a variação da frequência de relógio causa não só um desalinhamento temporal entre os traços, mas também um desalinhamento em amplitude, ou seja, um desalinhamento no eixo vertical. Portanto, experimentos foram realizados com o objetivo de tentar-se realizar um alinhamento vertical dos traços para melhorar o desempenho dos ataques DPA/DEMA. Adicionando-se para isto uma nova etapa ao fluxo proposto, tal como mostrado na Figura 35.

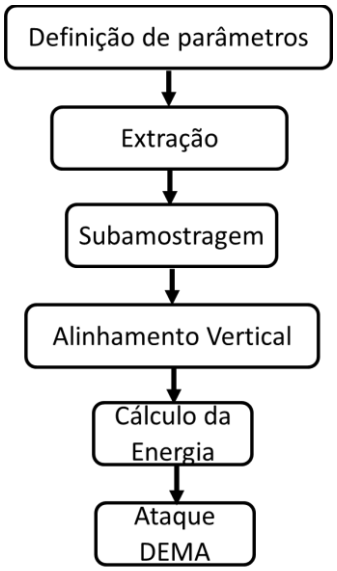

Figura 35 – Fluxo completo de ataque DEMA proposto.
Fonte: Própria.

Neste estudo de caso foram usados todos os traços de energia do conjunto com frequências entre 38 a 60MHz. Especificamente, utilizou-se o conjunto de assinaturas de energia calculadas com o segmento de 200 amostras. A escolha deste tamanho justifica-se pelo fato de produzir o melhor resultado em termos de quantidade média de traços para obter ataques bem-sucedidos.

A primeira abordagem para o alinhamento vertical foi remover o deslocamento provocado por uma tensão contínua associada ao sinal, conhecido em inglês como *Direct Current Offset.* Este *offset* pode ser calculado por meio do valor médio de todas as amostras contidas no sinal, neste caso da assinatura alvo extraída. Dessa forma, calculou-se o valor médio de cada assinatura e posteriormente a média dos valores

encontrados em todos os traços. Em seguida, é calculado um fator de correção através da Equação (30). Para cada traço, aplica-se o fator de correção a fim de que todos os traços tenham o mesmo valor médio, eliminando a diferença do offset entre eles, causada pelas diferentes frequências de relógio utilizadas e ruídos do sistema.

$$fc = offset_{médio} - offset \tag{30}$$

Na Equação (30) fc é o fator de correção, $offset_{médio}$ é a média dos deslocamentos verticais dos traços do conjunto e $offset$ é o valor médio, ou o deslocamento vertical do traço em questão. Este experimento trouxe os seguintes resultados, resumidos na Tabela 14.

Tabela 14 – Resultados dos ataques DEMA com etapa alinhamento vertical baseado na correção do valor médio dos traços.

	SBOX1	SBOX2	SBOX3	SBOX4	SBOX5	SBOX6	SBOX7	SBOX8	Média
Segm. Tamanho 200	653	2723	5829	2894	N/C	4224	3887	48414	3368,33
Dif. Sem o alinhamento	114	503	600	34	N/C	503	814	-16068	428,33

Na primeira linha da Tabela 14, temos a mínima quantidade de traços necessária para que a subchave em questão seja encontrada. Já na segunda linha, a diferença da quantidade de traços em relação aos experimentos anteriores utilizando o fluxo completo proposto. Os valores positivos significam que os resultados obtidos com a inclusão da etapa de alinhamento vertical ao fluxo exigem mais traços para encontrar a subchave em relação aos experimentos sem esta etapa. Já os números negativos informam uma redução do número de traços, o que seria desejado com tal inclusão da etapa.

Com base nos resultados obtidos na Tabela 14, nota-se que praticamente em todas as subchaves, o uso do alinhamento vertical piorou o desempenho do ataque, exceto com relação a SBOX 8, o que não pode ser considerado relevante pois esta SBOX normalmente apresenta disparidade em relação as demais como já observado previamente. Desta forma conclui-se que esta abordagem de alinhamento não é satisfatória.

Para este mesmo conjunto de traços, também é usada a abordagem para alinhamento vertical por meio do cálculo do valor RMS (em inglês, *root mean square*).

Portanto, da mesma maneira que na abordagem anterior, calculou-se um fator de correção com base no valor RMS, de acordo com a Equação (31):

$$fc = \frac{rms_{médio}}{rms} \quad (31)$$

Nesta Equação, fc é o fator de correção, $rms_{médio}$ é a média dos valores *RMS* dos traços e RMS é o valor RMS correspondente ao traço atual. Este fator *fc* é aplicado a todos os traços do conjunto a fim de garantir que todos tenham o mesmo valor RMS médio do conjunto. A aplicação do fluxo com esta etapa produziu os resultados resumidos na Tabela 15:

Tabela 15 – Resultados dos ataques DEMA com etapa alinhamento vertical baseado na correção do valor rms dos traços.

	SBOX1	SBOX2	SBOX3	SBOX4	SBOX5	SBOX6	SBOX7	SBOX8	Média
Segm. Tamanho 200	1773	2685	6523	3474	N/C	5050	4049	50349	3925,67
Dif. Sem o alinhamento	1234	465	1294	614	N/C	1334	976	-14133	986,17

Do mesmo modo que o experimento anterior, é possível perceber que novamente, a adição da etapa de alinhamento vertical, não melhorou a qualidade dos ataques. Como este conjunto contém a maioria dos traços com frequências de relógio variando entre 38 a 42MHz, por estratégia foi realizado o alinhamento vertical através da correção do valor RMS médio gerado por este grupo. Para este experimento obteve-se os resultados apresentados na Tabela 16:

Tabela 16 – Resultados dos ataques DEMA com etapa alinhamento vertical baseado na correção do valor rms dos traços para o rms médio do grupo de 38 a 42MHz.

	SBOX1	SBOX2	SBOX3	SBOX4	SBOX5	SBOX6	SBOX7	SBOX8	Média
Segm. Tamanho 200	1773	2685	6523	3474	N/C	5050	4049	50349	3925,67
Dif. Sem o alinhamento	1234	465	1294	614	N/C	1334	976	-14133	986,17

Estes resultados são similares aos obtidos no experimento anterior, o que confirma a influência dos traços de 38 a 42MHz no conjunto devido a sua maioria. Da mesma forma, estes resultados não são satisfatórios, não justificando a inclusão desta etapa no fluxo de ataque.

O experimento que consiste em ajustar todos os traços para o RMS médio do conjunto, foi realizado também com o conjunto de traços de 38 a 42MHz. O experimento foi realizado aplicando-se o ataque sobre os traços de energia com melhor tamanho de segmento para os experimentos com este conjunto, que é de 300 amostras. Este experimento teve como resultado, o mostrado na Tabela 17, abaixo:

Tabela 17 – Resultados dos ataques DEMA com etapa alinhamento vertical baseado na correção do valor rms médio no grupo de 38 a 42MHz.

	SBOX1	SBOX2	SBOX3	SBOX4	SBOX5	SBOX6	SBOX7	SBOX8	Média
Segm. Tamanho 300	526	1866	7221	2141	41050	7917	1406	51817	3512,83
Dif. Sem o alinhamento	333	270	5426	1012	-1178	3030	147	-	1703

Este experimento mostra redução na quantidade de traços somente na SBOX 5, que teve problemas durante sua aquisição e deve ser desconsiderada. Nas demais SBOXs houve um aumento na quantidade de traços, o que confirma mais uma vez que o alinhamento vertical dos traços através do ajuste dos valores RMS produziu resultados insatisfatórios.

Em (HAJRA, SUVADEEP; MUKHOPADHYAY, DEBDEEP, 2013) o alinhamento vertical foi realizado ajustando o valor médio em função da média dos *offsets* de uma porção dos traços onde não ocorre computações. Logo, um estudo de caso visando usar esta estratégia é realizado sobre o conjunto de traços disponíveis. Assim, ajusta-se o *offset* dos traços para que todos tenham o valor médio dos *offsets* das últimas 5000 amostras dos traços originais, ou seja, média da região em que não é executado processamento. Os resultados obtidos com este estudo de caso são resumidos na Tabela 18:

Tabela 18 – Resultados dos ataques DEMA com etapa alinhamento vertical baseado na correção do valor rms médio da região sem computação dos traços.

	SBOX1	SBOX2	SBOX3	SBOX4	SBOX5	SBOX6	SBOX7	SBOX8	Média
Segm. Tamanho 300	193	1596	1794	1130	42225	4932	1259	N/C	1817,33
Dif. Sem o alinhamento	0	0	-1	1	-3	45	0	-	7,5

Observando as informações da Tabela 18 nota-se claramente que não há alteração da quantidade de traços com esta estratégia.

Com isto, conclui-se que pelo fato destas estratégias de alinhamento vertical

não alterarem a forma dos traços, parte em que encontra-se o vazamento das informações, realizando apenas deslocamentos ou então alterando sua escala, a mesma não proporcionou uma redução no número de traços com a realização dos ataques, ao contrário, na maioria dos casos mantém-se ou aumenta a quantidade média de traços necessários. Além é claro de adicionar mais uma etapa de processamento no fluxo de ataques, mostrando-se insatisfatória sua aplicação.

6.7 Tempo de Processamento

Ainda em se tratando do fluxo de ataques da Figura 17 proposto nesta dissertação, foram realizadas observações em relação ao tempo de processamento das etapas constituintes do mesmo. Neste ponto, são utilizados como métrica de comparação os tempos de processamento da execução do POC e do DTW no trabalho realizado por Loder et al, em (LODER, L. L., 2014), mostrado na Tabela 21.

De acordo com o fluxo de ataques visto na Figura 17, a primeira etapa de processamento consiste na extração da assinatura alvo dos traços. Essa etapa é seguida de um processo de subamostragem, conforme visto no Capítulo 5. Os tempos dessas duas etapas do alinhamento são mostradas na Tabela 19.

Tabela 19 – Tempo de processamento das etapas de extração e subamostragem.

Tempos		
	Extração	**Subamostragem**
1 Traço	0.340797s	2.882738s
77820 Traços	2h18min	1h29min

Esses tempos não podem ser confrontados com (LODER, L. L., 2014), pois em seu trabalho, o mesmo menciona etapas de filtragem e subamostragem por transformada *wavelet*, mas não informa o tempo de execução destas etapas.

Os tempos de execução do cálculo da energia, etapa seguinte do fluxo da Figura 17, para cada um dos tamanhos de segmento calculado, são mostrados na Tabela 20.

Tabela 20 – Tempo de processamento da etapa de cálculo da energia dos traços.

Energia		
Tamanho Seg.	**1 Traço**	**77820 Traços**
10	1.978132s	40.2 min
20	2.423546s	24.6 min
30	2.510886s	25.8 min
40	2.633730s	26.4 min
50	2.730861s	19.8 min
100	2.564215s	24 min
150	1.981449s	23.4 min
200	2.143772s	16.2 min
300	4.175328s	33.6 min
400	1.994034s	16.2 min

Como podemos ver na Tabela 20, os tempos de processamento desta etapa variam entre 16 e 40 min aproximadamente. Essa etapa de processamento pode ser comparada com as etapas de POC e DTW realizadas em (LODER, L. L., 2014). Para isto, temos na Tabela 21 os resultados obtidos pelo mesmo.

Tabela 21 – Tempo de processamento das etapas de POC e DTW por (LODER, L. L., 2014) – média de 100 alinhamentos.

	Freq. próximas	**Freq. distintas**	**Traços subamostrados**
POC - 1 traço	0.047s	0,0705s	0.0063s
DTW - 1 traço	134.8803s	135.3049s	0.2292s
POC - 77820 traços	1h01min	1h31min	7min10s
DTW - 77820 traços	17.36 semanas	17.41 semanas	4h57min

Como pode ser visto na Tabela 21, os tempos de execução do DTW são impraticáveis, sendo portanto desconsiderados para termos comparativos. Já o POC resulta em tempos maiores que os do cálculo da energia, vistos na Tabela 20, para traços não subamostrados. Para os traços subamostrados o tempo é menor que os tempos da Tabela 20. Porém, (LODER, L. L., 2014) utilizam uma etapa adicional de subamostragem baseada na transformada wavelet, a qual não menciona o tempo dispendido. Além disso, deve ser observado, que tanto na Tabela 19 quanto na Tabela 20, os tempos computados são relativos à execução real das etapas do fluxo, e não estimativas tal como apresentado em (LODER, L. L., 2014), o que pode comprometer os resultados apresentados. Destaca-se também, que em (LODER, L. L., 2014) algumas das funções utilizadas foram otimizadas pelo autor, o que não foi feito nesta

dissertação. O que contribuiu para o tempo reduzido no processamento dos traços subamostrados realizado por (LODER, L. L., 2014).

A seguir, na Tabela 22 são mostrados os tempos relativos ao processamento dos ataques DEMA executados sobre os 77820 traços de energia para cada um dos segmentos calculado.

Tabela 22 – Tempo de processamento dos ataques DEMA sobre os traços de energia.

Ataques DEMA	
Tamanho Seg.	**Tempo**
0	16h29min
10	8h05min
20	7h49min
30	7h46min
40	7h35min
50	7h31min
100	7h43min
150	8h50min
200	9h11min
300	8h43min
400	8h28min

Através da Tabela 22, podemos ver que o ataque varia de 7 h31min a 16h29min aproximadamente. Assim, se pegarmos o pior caso do fluxo proposto, contabilizando 2h18min da etapa de extração, 1h29min da subamostragem e 40min do pior caso dos cálculos de energia, temos como resultado um tempo total de aproximadamente 4h27min. Esse tempo é pelo menos 1,69 vezes menor que o tempo de processamento do ataque no seu melhor caso (7h31min). Com isto, podemos perceber que o fluxo proposto não apresenta tempo de processamento excessivo, o que o inviabilizaria.

7 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou um fluxo de ataques DEMA/DPA baseado no cálculo da energia dos traços de radiação eletromagnética ou de potência, capaz de revelar a chave criptográfica em dispositivos dotados de contramedidas baseadas na inserção de ruído nos traços de radiação eletromagnética ou de potência dos dispositivos. Destaca-se como contribuição deste trabalho, a etapa de extração da assinatura alvo dos traços e a avaliação do impacto do tamanho dos segmentos para o cálculo da energia nos ataques realizados.

O presente trabalho teve como motivação o crescente uso de dispositivos criptográficos implementados em *hardware*, que são potenciais alvos dos ataques DEMA/DPA. Por sua vez, este tipo de ataque foi escolhido como estudo de caso em função de sua popularidade, consequência de características como a necessidade de *hardware* reduzido para sua implementação, ser um tipo de ataque não-invasivo, revelar os dados ocultos mesmo na presença de ruído.

Como destacado neste estudo, o principal desafio desta dissertação consiste em realinhar no domínio do tempo os traços previamente desalinhados por contramedidas como a inserção de atrasos aleatórios e/ou variação da frequência de relógio do dispositivo atacado. Além disso, o número de traços para se obter sucesso nos ataques deve ser o menor possível, e o tempo de processamento dos algoritmos envolvidos não deve ser excessivo. Neste caso, considerou-se como métrica de comparação os resultados obtidos por (LODER, L. L., 2014).

Através dos resultados obtidos, pode-se verificar a eficácia do fluxo de ataques proposto no objetivo de revelar a chave criptográfica, mesmo em dispositivos dotados das contramedidas de variação randômica das frequências de relógio e inserção de atrasos aleatórios. Os experimentos mostraram ainda, que existe uma redução na quantidade média de traços necessários para revelar as subchaves criptográficas em relação aos experimentos realizados por (LODER, L. L., 2014), com POC e filtragem. O fluxo de ataques proposto realiza duas etapas de subamostragem, pois o cálculo da energia reduz também a quantidade de amostras em relação aos traços originais. Isto resulta em uma redução no esforço computacional do sistema que implementa o ataque, reduzindo assim o tempo necessário para a descoberta da chave criptográfica. Como o fluxo proposto por (LODER, L. L., 2014) possui restrições de frequência, podendo esta ter uma variação de no máximo 10% do seu valor. Foram

realizados experimentos com grupos de traços separados por frequência de relógio, formando um grupo com frequências de 38 a 42MHz e outro com frequências de 55 a 60MHz para comparações com os resultados obtidos por (LODER, L. L., 2014). Sobre estes grupos alcançou-se uma redução de até respectivamente 83,96% e 15,05% na quantidade média de traços necessários para obter-se sucesso nos ataques, em relação aos experimentos com POC e filtro realizados por Loder (LODER, L. L., 2014). Estes resultados foram obtidos graças a efetividade da etapa de extração, conforme os resultados mostrados na Seção 6.1, nos quais é observada uma taxa de sucesso de acima de 90%. Também destaca-se a subamostragem, como método de pré-alinhamento, reduzindo a defasagem inicial entre os traços. Conforme vimos, a etapa de cálculo da energia, utilizada como ajuste fino do alinhamento, obteve bons resultados, por estar associada a subamostragem.

Experimentos com traços cobrindo toda a faixa de frequências de relógio disponibilizadas, ou seja, desde 38 até 60MHz foram realizados, através dos quais a chave criptográfica foi revelada com menos de 3 mil traços. Vale observar, que estes resultados foram obtidos a partir de experimentos com o cálculo da energia para diferentes tamanhos de segmento, dentre os quais, os segmentos com tamanho de meio ciclo da frequência de relógio intermediária dos grupos obtiveram os melhores resultados.

O fluxo proposto apresenta como limitação, a necessidade da definição de parâmetros através da observação dos traços, como o limiar e o ponto inicial da varredura, fazendo com que o processo de extração das assinaturas dos traços não seja totalmente automático. Porém, através da análise de uma pequena amostra dos traços é possível encontrar estes parâmetros com facilidade, o que não torna o processo inviável.

Pode-se também citar como limitação do fluxo apresentado, o fato de não se conseguir atacar dispositivos que possuam variação de frequência de relógio durante a execução das rodadas do algoritmo. Pois as contramedidas utilizadas como estudo de caso, apresentam frequência de relógio diferentes para cada nova execução de decriptação/encriptação, mantendo-se constante durante a execução. Como a etapa de extração é realizada com base na frequência de relógio de operação do traço, a mesma deve ser constante para que a extração seja realizada satisfatoriamente. Para atacar dispositivos com esta contramedida, deve ser investigada nova forma de realizar a extração das assinaturas alvo.

Experimentos com relação ao alinhamento vertical dos traços, ou seja, alinhamento em amplitude também foram realizados, porém não apresentaram resultados satisfatórios, mantendo ou até mesmo aumentando a quantidade de traços necessária para realizar os ataques. Acredita-se que as técnicas de alinhamento vertical aplicadas não obtiveram sucesso pois realizam apenas mudança na amplitude dos traços, sendo que a informação vazada está relacionada a forma de onda dos mesmos.

Para trabalhos futuros indica-se a busca por técnicas de pré-processamento capazes de revelar a chave criptográfica mesmo nas arquiteturas operando com o *pipeline* cheio, pois não foram realizados testes com esta configuração no presente trabalho. Porém, novas estratégias deverão ser desenvolvidas para este fim, pois com o *pipeline* cheio, há a sobreposição dos traços do consumo, sendo as técnicas aqui apresentadas inadequadas para atacar dispositivos dotados desta contramedida.

Embora acreditamos que o fluxo de ataques proposto tenha sucesso nas configurações GALS4 e GALS8, experimentos não foram realizados nesta dissertação. Assim, em trabalhos futuros pode-se explorar estes sistemas, utilizando-se para isto, o fluxo aqui proposto além de outras estratégias de alinhamento dos traços. Como o fluxo apresentado neste trabalho realiza a extração da assinatura alvo independentemente do número de rodadas de execução do algoritmo criptográfico em questão, há uma probabilidade grande de sucesso nos ataques.

Futuramente, seria interessante também, obter-se novas aquisições dos traços de potência, para possibilitar novos testes. Pretende-se desenvolver um kit para aquisição dos traços de radiação eletromagnética, construindo uma sonda e utilizando circuitos integrados amplificadores de baixo-ruído para amplificar os traços antes de realizar a aquisição através de um osciloscópio. Com isto, pretende-se construir um equipamento de aquisição dos traços que seja de baixo custo, pois tais equipamentos são bastante caros.

Ainda, citamos como trabalhos futuros a proposta de novas arquiteturas contendo diferentes contramedidas para inviabilizar os ataques DPA/DEMA, contribuindo com a área de segurança de dados.

REFERÊNCIAS

AVIRNENI, N. D. P.; SOMANI, A. K. Countering Power Analysis Attacks using Reliable and Aggressive Designs. **IEEE Transactions on Computers**, Los Alamitos, USA, vol. 63, nº. 6, p. 1408-1420, 2013.

BADDAM, K.; ZWOLINSKI, M. **Evaluation of Dynamic Voltage and Frequency Scaling as a Differential Power Analysis Countermeasure**. In: 20th International Conference on VLSI Design. Bangalore, India: (VLSID'07). 2007. p. 854-862.

BENINI, L.; MACII, A.; MACII, E.; OMERBEGOVIC, E.; PRO, F.; PONCINO, M. **Energy-Aware Design Techinques for Differential Power Analysis Protection**. In: 40th Design Automation Conference. Anaheim, USA: (DAC'03). 2003. p. 36-41.

BUCCI, M.; LUZZI, R.; GUGIELMO, M.; TRIFILETTI, A. **A Countermeasure against Differential Power Analysis based on Random Delay Insertion.** In: IEEE International Symposium on Circuits and Systems. Kobe, Japan: (ISCAS'05). 2005. p. 3547-3550.

CHARVET, X.; PELLETIER, H. **Improving DPA Attack using Wavelet Transform**. In: NIST Physical Security Testing Workshop. Honolulu, USA: (NIST PSTW'05). 2005.

CLAVIER, C.; CORON, J.; DABBOUS, N. **Differential Power Analysis in the Presence of Hardware Countermeasures**. In: Cryptographic Hardware and Embedded Systems. Worcester, USA: (CHES'00). 2000. p. 252-263.

DINIZ, P. S. R.; SILVA, E. A. B.; NETTO, S. L. **Processamento Digital de Sinais - Projeto e Análise de Sistemas**. 2. ed. Porto Alegre, Brasil: Bookman, 2014.

FILHO, S. N. **Filtros Seletores de Sinais**. 2. ed. Florianópolis, Brasil: Editora da UFSC, 2003.

GEBOTYS, C.; TIU, C.; CHEN, C. **A Countermeasure for EM Attacks of a Wireless PDA**. In: International Conference on Information Technology: Coding and Computing. Las Vegas, USA: (ITCC'05). 2005. p. 544-549.

GUILLEY, S.; KHALFALLAH, K.; LOMNÉ, V.; DANGER, J. **Formal Framework for the Evaluation of Waveform Resynchronization Alorithms**. In: Workshop in Information Secutity and Practice. Heraklion, Greece: (WISTP'11). 2011. p. 100-115.

HAJRA, SUVADEEP; MUKHOPADHYAY, DEBDEEP. Pushing the Limit of Non-Profiling DPA using Multivariate Leakage Model. **Cryptology ePrint Archive (IACR)**, 2013. 9.

HALLIDAY, D.; RESNICK, R.; WALKER, J. **Fundamentos de Física 3 - Eletromagnetismo**. 9. ed. São Paulo: LTC, v. 3, 2012.

HAYKIN, S.; MOHER, M. **Sistemas de Comunicação**. 5. ed. Porto Alegre, Brasil: Bookman, 2011.

HODGERS, P.; HANLEY, N.; O'NEILL, M. **Pre-Processing Power Traces with a Phase-Sensitive Detector**. In: IEEE International Symposium on Hardware-Oriented Security and Trust. Austin, USA: (HOST'13). 2013. p. 131-136.

KIM, N. S.; AUSTIN, T.; BLAAUW, D.; MUDGE, T.; FLAUTNER, K.; HU, J. S.; IRWIN, M. J.; KANDEMIR, M.; NARAYANAN, V. Leakage Current: Moore's Law Meets Static Power. **IEEE Computer**, vol. 36, n°. 12, p. 68-75, Dec 2003.

KOCHER, P. C. **Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems.** In: 16th International Cryptology Conference on Advances in Cryptology. London, UK: (CRYPTO'96) Springer-Verlag. 1996. p. 104-113.

KOCHER, P. C.; JAFFE, J.; JUN, B. **Differential Power Analysis**. In: 19th International Cryptology Conference on Advances in Cryptology. Santa Barbara, USA: (CRYPTO'99) Springer-Verlag. 1999. p. 388-397.

LE, T. H.; CLÉDIÈRE, J.; SERVIÈRE, C.; LACOUME, J. **Efficient Solution for Misalignment of Signal in Side Channel Analysis**. In: IEEE International Conference on Acoustics, Speech and Signal Processing. Honolulu, USA: (ICASSP'07). 2007. p. 257-260.

LODER, L. L. **Proposta de um fluxo de ataque DPA para avaliar a vulnerabilidade de arquiteturas criptográficas protegidas por aleatorização de processamento**. UFPel. Pelotas, p. 91. 2014. (Dissertação de Mestrado).

LOMNÉ, V.; MAURINE, P.; TORRES, L.; ROBERT,M. S.; CALAZANS, N. **Evaluation on FPGA of Triple Track Logic Robustness against DPA and DEMA**. In: Design, Automation and Test in Europe Conference and Exhibition. Nice, France: (DATE'09). 2009. p. 634-639.

LU, Y.; O'NEILL, M.; MCCANNY, J. **Implementation and Analysis of Random Delay Insertion Countermeasure against DPA**. In: International Conference on Field-Programmable Technology. Taipei, Taiwan: (FTP'08). 2008. p. 201-208.

MATHWORKS. Mathworks. **Mathworks**, 2014. Disponivel em: <https://www.mathworks.com/help/dsp/ref/dsp.firdecimator-class.html>. Acesso em: 20 nov. 2016.

MATLAB. Mathworks. Disponivel em: <https://www.mathworks.com/products/matlab.html>. Acesso em: 20 Dec 2016.

NAGASHIMA, S.; HOMMA, N.; IMAI, Y.; AOKI, T.; SATOH, A. **DPA Using Phase-Based Waveform Matching against Random-Delay Countermeasure**. In: International Symposium on Circuits and Systems. New Orleans, USA: (ISCAS'07). 2007. p. 1807-1810.

NIST. http://csrc.nist.gov. **DATA ENCRYPTION STANDARD (DES)**, 1999. Disponivel em: <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>. Acesso em: 20 dezembro 2016.

PATEL, H.; BALDWIN, R. **Differential Power Analysis using Wavelets Decomposition**. In: Military Communications Conference. Orlando, USA: (MILCOM'12). 2012. p. 1-5.

POPP, T.; MANGARD, S. **Implementation Aspects of the DPA-Resistant Logic Style MDPL**. In: IEEE International Symposium on Circuits and Systems. Kos, Greece: (ISCAS'06). 2006. p. 2913-2916.

RÉAL, D.; CANOVAS, C. **Defeating classical Hardware Countermeasures:** a new processing for Side Channel Analysis. In: Design, Automation & Test in Europe Conference & Exhibition. Munich, Germany: (DATE'08). 2008. p. 1274-1279.

SAKOE, H.; CHIBA, S. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. **IEEE Transactions on Acoustics, Speech and Signal Processing**, vol 26, nº 1. 1978. p. 43-49.

SOARES, R. I. **Arquiteturas GALS Pipeline para Criptografia Robusta a Ataques DPA e DEMA**. PUCRS. Porto Alegre, Brasil, p. 147. 2010. (Tese de Doutorado).

SOUISSI, Y.; ELAABID, M. A.; DEBANDE, N.; GUILLEY, S.; DANGER, J. **Novel Applications of Wavelet Transform Based Side-Channel Analysis**. In: Non Invasive Attack Testing Workshop. Nara, Japan: (NIAT'11). 2011. p. 30-35.

SOUISSI, Y.; GUILLEY, S.; BHASIN, S.; DANGER, J. **Common Framework to Evaluate Modern Embedded Systems against Side-Channel Attacks**. In: Technologies for Homeland Security IEEE International Conference. Waltham, USA: (HST'11). 2011. p. 89-91.

TIAN, Q.; HUSS, S. A. **A General Approach to Power Trace Alignment for the Assessment of Side-Channel Resistance of Hardened Cryptosystems**. In: Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. Piraeus-Athens, Greece: (IIH-MSP). 2012. p. 465-470.

TIAN, Q.; HUSS, S. A. **On Clock Frequency Effects in Side Channel Attacks of Symmetric Block Ciphers**. In: New Technologies, Mobility and Security. Istanbul, Turkey: (NTMS). 2012. p. 1-5.

TIAN, Q.; SHOUFAN, A.; STOETTINGER, M.; HUSS, S. A. **Power Traces for Cryptosystems featuring Random Frequency Countermeasures**. In: IEEE Digital Infomation Processing and Comunications. Klaipeda City, Lithuania: (ICDIPC). 2012. p. 51-55.

VAHEDI, H.; MURESAN, R.; GREGORI, S. **On-chip Current Flattering Circuit with Dynamic Voltage Scaling**. In: IEEE International Symposium on Circuits and Systems. Kos, Greece: (ISCAS'06). 2006. p. 4277-4280.

WILSKY, A. S.; NAWAB, S. H.; OPPENHEIM, A. V. **Sinais e Sistemas**. 2. ed. Rio de Janeiro, Brasil: Pearson Education, 2010.

WOUDENENBERG, J; WITTEMAN, M; BAKKER, B. **Improving Differential Power Analysis by Elastic Alignment**. In: Cryptographer's Track at the RSA Conference. San Francisco, USA: Springer: (CT-RSA). 2009. p. 104-119.

XILINX, INC. **Spartan-3 Starter Kit Board User Guide**. User Guide. (UG130), p. 64. 2005. (v1.1).

YANG, S.; WOLF, W.; VIJAYKRISHNAN, N.; SERP. **Power Attack Resistant Cryptosystem Design:** A Dynamic Voltage and Frequency Switching Approach. In: Design, Automation and Test in Europe Conference and Exhibition. Munich, Germany: ('DATE'05). 2005. p. 64-69.

ZAFAR, Y.; HAR, D. **A Novel Countermeasure Enhancing Side Channel Immunity in FPGAs**. In: International Conference on Advanced in Electronics and Microelectronics. Valencia, Spain: (ENICS'08). 2008. p. 132-137.