UMA ABORDAGEM PARA ESTIMAR A SEGURANÇA DE CIRCUITOS CRIPTOGRÁFICOS À ANALISES POR CONSUMO DE POTÊNCIA

LODER, Luciano Ludwig[1]; SOARES, Rafael Iankowski[2]

[1] *Universidade Federal de Pelotas – lucianoloder@gmail.com*
[2] *Universidade Federal de Pelotas – rafael.soares@inf.ufpel.edu.br*

1. INTRODUÇÃO

A confidencialidade de informações sempre foi um fator chave na história da humanidade. Segundo [STA08], cifras obtidas por meio da criptografia de mensagens vêm sendo desenvolvidas desde o Império Romano, para manter secreta a comunicação entre generais. Em paralelo com a arte de ocultar mensagens com o uso da criptografia, a arte de desvendar métodos de cifragem conhecida como criptoanálise foi se desenvolvendo, sendo responsável pelo fracasso e sucesso de inúmeras batalhas, como exemplifica [SIN01]. Deste modo o processo de cifragem de informação passou a exigir o uso de recursos computacionais para suportar tal complexidade. Desde então, algoritmos criptográficos se baseiam cada vez mais em funções matemáticas complexas executadas por máquinas dedicadas, ou softwares dedicados à criptografia. Nas últimas décadas, o aumento de serviços oferecidos através da internet exige protocolos e algoritmos criptográficos que garantam o sigilo de informações trafegadas em redes públicas.

Embora a criptografia tenha sido desenvolvida para garantir que algoritmos sejam robustos à tentativas de violação de dados confidenciais, novas técnicas demostram que através do monitoramento de propriedades físicas dos circuitos é possível revelar dados secretos processados. Esta classe de ataque é conhecida como ataque a canais laterais – Side Channel Attacks – SCA. Estes ataques procuram estabelecer uma relação de dependência entre os dados processados e grandezas físicas tais como consumo de potência, radiação eletromagnética e tempo de processamento. As vulnerabilidades têm origem principalmente nas características de implementação da tecnologia CMOS (Complementary Metal Oxide Semiconductor) de fabricação de circuitos e no paradigma síncrono adotado para a concepção de sistemas digitais. A sincronização das operações com o uso do relógio global facilita a correlação entre dados e efeitos físicos mensuráveis, como apresentado por Kocher [KOC99].

A potência consumida por um circuito CMOS é fundamentalmente oriunda de duas diferentes classes de consumo: componente estática e componente dinâmica. A componente estática corresponde à potência dissipada quando o circuito se encontra em estado de equilíbrio, ou seja, não existem transições em suas entradas ou saídas. Já a componente dinâmica ocorre durante as transições ocorridas no circuito, em decorrência de variações de sinais de entrada e do estado interno do circuito. Para as análises por consumo de potência é possível desprezar a potência estática em relação à potência dinâmica.

Variações nos dados de entrada provocam o chaveamento de sinais internos do circuito, que são propagados até sua saída. Os ataques por análise de consumo de potência exploram estas variações e através de métodos estatísticos estabelecem uma relação de dependência entre essas variações e os dados de entrada do circuito [KOC99].

Uma das principais estratégias para evitar a relação de dependência entre dados processados e o consumo de potência é construir circuitos que tenham o mesmo consumo de potência independente do dado a ser processado, ou seja, tenham a mesma atividade de chaveamento para qualquer dado de entrada. Desta forma o presente trabalho tem como objetivo propor uma abordagem para analisar a atividade de chaveamento de circuitos criptográficos de modo a estimar sua vulnerabilidade a ataques por análises de consumo de potência.

2. MATERIAL E MÉTODOS

As análises por consumo de potência exigem recursos específicos para realizar a medição de potência tal como plataforma de prototipação adequada e osciloscópio de precisão para medição e aquisição de dados. Além disso, um grande esforço computacional é exigido para a execução das análises, podendo ser concluídas em algumas horas ou até mesmo em alguns dias, dependendo do tipo de análise, da quantidade de medições realizadas e a configuração do sistema computacional disponível para executar as análises. Um método rápido e de baixo custo para estimar a vulnerabilidade de um circuito é analisar a atividade de chaveamento do circuito durante a seu processamento. Circuitos com mesma atividade de chaveamento para todas combinações possíveis de valores de entrada são desejados para implementar sistemas criptográficos imunes a estes tipos de ataques. Embora analisar o chaveamento do circuito seja uma maneira de estimar a vulnerabilidade de um circuito, o método proposto não possui a precisão nem substitui as análises DPA ou SPA.

Neste trabalho é proposta uma abordagem para avaliar a atividade de chaveamento de circuitos digitais usando o ambiente Quartus II da Altera. A proposta visa automatizar o processo de analisar o número de chaveamentos realizados por um circuito para processar todos os possíveis dados de entrada.

A abordagem proposta é empregada para avaliar o funcionamento do submódulo SBOX do algoritmo criptográfico DES. Este submódulo é uma parte vulnerável do algoritmo DES, sendo o principal alvo de ataques SCA tal como proposto por Kocher em [KOC99]. A interface da SBOX é composta por 6 bits de entrada e 4 bits de saída. A informação de entrada é o resultado de uma operação lógica XOR entre 6 dos 64 bits da mensagem de entrada e 6 dos 64 bits da chave criptográfica. Deste modo, as análises se restringem a avaliar apenas 2^6 possibilidades de uma parte da chave criptográfica do algoritmo. O algoritmo de criptografia AES, por exemplo, possui o SBOX com 8 bits de entrada, o que aumenta o número de possibilidades de valores de entrada e, com isso, o processo de avaliação, justificando o uso da abordagem proposta.

Duas implementações em hardware de um SBOX, ambas descritas em linguagem VHDL, são avaliadas neste trabalho. A primeira implementação é obtida pela descrição tradicional de um circuito digital usando lógica em trilha simples. A segunda implementação da SBOX é obtida usando a lógica STTL (Secure Triple Track Logic), um estilo lógico proposto para evitar a fuga de informações através da uniformização do consumo de potência [SOA08]. Em STTL cada bit de informação é representado por 3 trilhas definidas como (B_T,B_F,B_V). A representação do valor lógico ‘1’ é definida como (1,0,1). Já o valor lógico ‘0’ é representado por (0,1,1). As transições de valores lógicos ‘0’ para ‘1’ ou ‘1’ para ‘0’ necessariamente devem ser intercaladas por um espaçador, definido como (0,0,0). A terceira trilha, chamada trilha de validade, garante um tempo de propagação constante na computação do circuito. Qualquer

combinação diferente das mencionadas é considerada um valor inválido. Deste modo a transição de um sinal lógico qualquer sempre envolve o chaveamento de duas trilhas. Isto garante um consumo de potência uniforme independente do valor lógico de entrada. Maiores detalhes sobre o funcionamento de STTL é encontrado em [SOA08].

O projeto e validação das S-BOX é realizado no ambiente Quartus II da Altera que até sua versão 9 dispõe de uma ferramenta de simulação. Esta ferramenta permite validar o circuito através de uma interface gráfica representada por formas de onda e salvas em um arquivo com a extensão "vwf". O Quartus II também permite executar a simulação diretamente pela linha de comandos, usando o comando *quartus_sim <nome_projeto>*. Como resultado, entre as informações fornecidas, encontra-se o número total de transições da simulação.

A ferramenta Quartus II permite realizar simulações de um projeto tantas vezes quanto necessário sem a necessidade de sintetizá-lo novamente. Isto permite a elaboração de um script que faça automaticamente a avaliação das transições de um circuito para um conjunto de entradas. Para tanto basta que se tenha um arquivo de teste para cada dado de entrada do circuito que se deseja avaliar.

Ao se analisar o formato dos arquivos de teste, nota-se uma estrutura bastante regular, formada por um cabeçalho contendo a descrição de cada sinal, as transições de cada sinal ao longo do tempo de simulação, e, por último, uma descrição da forma de representação de cada sinal para o usuário no ambiente Quartus II. Essa organização da estrutura do arquivo permite que seja criado um programa para a geração automática desses arquivos. Para isso é desenvolvido um programa em C capaz de produzir os arquivos de teste para as 64 possibilidades de arquivos de entrada, alterando apenas os dados de entrada para cada arquivo.

Um script para a execução das simulações também foi desenvolvido. O script é responsável por selecionar o arquivo de simulação, executá-la e obter a informação do número de transições. Para pesquisar e obter o número de transições utiliza-se o comando *grep*, do sistema operacional Unix. Como o ambiente Quartus II usado nos experimentos encontrava-se disponível apenas para o sistema operacional Windows, foi necessária a instalação do software C*ygwin*, que simula o ambiente Unix sob o Windows. Uma alternativa a esta solução é o uso do pacote "*microsoft for unix"*, da Microsoft, ou ainda, a instalação do ambiente Quartus II sob o sistema operacional Linux, mantendo apenas um sistema operacional para o sistema.

3. RESULTADOS E DISCUSSÕES

O número de transições dos circuitos das S-BOX para lógica simples e lógica STTL encontram-se na Figura 1. Os resultados demonstram que a diferença entre o número máximo e mínimo de transições na implementação utilizando lógica STTL é menor em relação a SBOX definida em lógica simples. Isto demonstra o potencial da lógica STTL em manter quase independente de dados sua atividade de chaveamento. O número médio de transições entre as implementações difere significativamente devido à diferença no número de trilhas usadas para representar os dados. A execução dos scripts avaliou os 64 valores de entrada em 2 horas usando um PC core I5 2GHZ e 8 GB de RAM.

k_in	transitions									
1	10	17	28	32	36	47	81	max=	115
2	28	18	48	33	29	48	64	min=	10
3	28	19	53	34	42	49	52		
4	37	20	47	35	46	50	82		
5	31	21	44	36	63	51	77		
6	47	22	76	37	59	52	86		
7	49	23	71	38	61	53	75		
8	33	24	63	39	68	54	100		
10	49	25	47	40	62	55	98		
11	46	26	69	41	52	56	88		
12	58	27	69	42	63	57	68		
13	54	28	68	43	64	58	92		
14	60	29	60	44	74	59	88		
15	62	30	93	45	67	60	92		
16	31	31	82	46	75	61	82		

input	transitions								
0	534	16	528	32	528	48	526	max=	536
1	532	17	522	33	524	49	526	min=	510
2	526	18	524	34	530	50	526		
3	526	19	522	35	528	51	524		
4	536	20	524	36	526	52	522		
5	530	21	522	37	526	53	518		
6	534	22	520	38	532	54	514		
7	532	23	520	39	526	55	510		
8	532	24	524	40	530	56	526		
9	530	25	522	41	528	57	524		
10	530	26	526	42	516	58	522		
11	528	27	522	43	514	59	520		
12	532	28	524	44	530	60	528		
13	528	29	524	45	528	61	526		
14	522	30	526	46	526	62	526		
15	518	31	524	47	526	63	522		

Figura 1: Resultado das simulações. À esquerda, lógica simples, à direita, lógica STTL

4. CONCLUSÕES

Neste trabalho foi proposta uma abordagem para estimar a vulnerabilidade de circuitos criptográficos a ataques por análise de consumo de potência. Experimentos realizados sob duas implementações em hardware do submódulo SBOX do algoritmo DES demonstram que uma simples avaliação da atividade de chaveamento permite estimar a robustez de um circuito criptográfico às análises por consumo de potência. O método usado emprega o ambiente Quartus II, um programa em C e scripts que permitem de maneira simples avaliar a atividade de chaveamento de um circuito para um significativo número de possibilidades de combinações de entrada. A abordagem, porém, limita-se a avaliação de circuitos combinacionais, não sendo efetiva quando aplicada a sistemas complexos, tal como um núcleo dedicado ao processamento completo de um algoritmo de criptografia. Em trabalhos futuros pretende-se investigar como tornar análises de consumo de potência mais eficientes para avaliar sistemas criptográficos a fim de produzir sistemas digitais seguros e confiáveis.

5. REFERÊNCIAS BIBLIOGRÁFICAS

KOCHER, P.; Jaffe, J.; Jun, B. **Differential Power Analysis**. In: 19th International Cryptology Conference on Advances in Cryptology (CRYPTO'99), Aug 1999, pp. 388-397.

SINGH, Simon. **O livro dos códigos**. Rio de janeiro, Editora Record, 2001.

SOARES, Rafael, CALAZANS, Ney, LOMNE, Victor, TORRES, Lionel, MAURINE, Philippe., ROBERT, Michel. Evaluating the Robustness of Secure Triple Track Logic through Prototyping. In: 21st Symposium on Integrated Circuits and Systems Design (SBCCI 2008), Gramado, Sep 2008. pp 193-198.

STALLINGS, W. **Criptografia e segurança de redes – princípios e práticas**. 4 ed. Pearson Education do Brasil, 2008