11

ENTRANDO NO MODO NERD: ANÁLISE AVANÇADA DE POTÊNCIA

Os dois capítulos anteriores, assim como a literatura de análise de potência em geral, focaram na compreensão teórica dos ataques e na aplicação deles em condições de laboratório. Como pessoas que testemunharam uma infinidade desses ataques, podemos dizer que, para a maioria dos alvos reais, 10 por cento do tempo é gasto configurando a medição corretamente; 10 por cento do tempo é dedicado à execução dos ataques de análise de potência reais, e os outros 80 por cento do tempo são gastos tentando entender por que os ataques não estão mostrando vazamentos. Isso ocorre porque seu ataque mostrará vazamentos apenas se você tiver realizado corretamente cada etapa, desde a aquisição do traço até a análise do traço, e até você realmente encontrar vazamentos, pode ser difícil determinar qual etapa estava errada desde o início. Na realidade, a análise de potência requer paciência, com muita análise de etapas, muita tentativa e erro, e é complementada com poder computacional. Este capítulo trata mais da arte da análise de potência do que da ciência.

Na prática, você precisará de algumas ferramentas extras para superar os diversos obstáculos que um alvo da vida real lançará em seu caminho. Esses obstáculos determinarão em grande parte o quão difícil será extrair um segredo de um dispositivo com sucesso. Algumas propriedades inerentes ao alvo que você está testando afetarão as características de sinal e ruído, assim como propriedades como programabilidade, complexidade do dispositivo e velocidade do relógio, tipo de canal lateral e contramedidas. Ao medir uma implementação de software de AES em um microcontrolador, você provavelmente será capaz de identificar as rondas de criptografia individuais de um único traço com um olho fechado e uma mão atrás das costas. Quando você está medindo um AES de hardware funcionando a 800 MHz incorporado em um System-on-Chip (SoC) completo, esqueça de ver as

rondas de criptografia em um único traço. Muitos processos paralelos causam ruído de amplitude, sem mencionar que o sinal de vazamento é extremamente pequeno. As implementações mais simples de AES podem ser quebradas em menos de 100 traços e 5 minutos de análise, enquanto os ataques mais complexos que vimos tiveram sucesso após passarem além de um bilhão(!) de traços e meses de análise — e, às vezes, o ataque ainda falha.

Nas próximas seções, forneceremos ferramentas para aplicar em várias situações e uma receita geral sobre como abordar todo o tópico de análise de potência. Equipado com essas ferramentas, cabe a você descobrir se, quando e como aplicá-las em seu alvo favorito. Como tal, este capítulo é um pouco de tudo. Primeiro, discutimos uma série de ataques mais poderosos e fornecemos referências. Em seguida, mergulhamos em várias maneiras de medir o sucesso na extração de chaves e como medir melhorias em sua configuração. Depois, falamos sobre a medição de dispositivos reais, em oposição a alguns alvos fáceis de laboratório, com controle total. Depois disso, há uma seção sobre análise e processamento de traços, e, finalmente, fornecemos algumas referências adicionais.

Os Principais Obstáculos

A análise de potência se apresenta em várias formas. Neste capítulo, faremos referência à análise de potência simples (SPA), à análise de potência diferencial (DPA) e ao ataque de potência por correlação (CPA), ou simplesmente à análise de potência quando uma afirmação se aplica a todos os três.

As diferenças entre a teoria e a realização de ataques em dispositivos reais são significativas. Você encontrará seus principais obstáculos ao realizar análises de potência reais. Esses obstáculos incluem os seguintes:

Ruído de Amplitude

Este é o chiado que você ouve ao escutar transmissões de rádio AM, o ruído de todos os outros componentes elétricos em sua configuração, ou o ruído aleatório adicionado como uma contramedida. Várias partes de sua configuração de medição irão causá-lo, mas operações não interessantes, mas paralelas, no dispositivo real também acabarão em sua medição. Você encontrará ruído de amplitude em todas as medições que realizar, e é um problema para seu ataque de potência porque ele obscurece as variações reais de potência devido ao vazamento de dados. Para CPA, isso faz com que o pico de correlação diminua em amplitude.

Ruído Temporal (também conhecido como Desalinhamento)

Jitter de Temporização causado pelo acionamento do osciloscópio ou trajetórias de tempo não constantes até a operação alvo resultam na operação de interesse aparecendo em tempos diferentes em cada traço. Este jitter afeta um ataque de potência por correlação porque o ataque pressupõe que o vazamento sempre apareça no mesmo índice de tempo. O jitter tem o efeito indesejado de ampliar o pico de correlação e diminuir sua amplitude.

Contramedidas de Canal Lateral

Sim, os fabricantes de chips e dispositivos também leem este livro. As fontes de ruído não intencionais descritas anteriormente também podem ser introduzidas intencionalmente pelos projetistas de dispositivos para diminuir a eficácia de um ataque de potência. Não apenas são introduzidas fontes de ruído, mas os sinais de vazamento são diminuídos usando algoritmos e designs de chips como mascaramento e ofuscação (consulte "Securing the AES Finalists Against Power Analysis Attacks", de Thomas S. Messerges), rotação constante de chave em um protocolo (consulte "Leakage Resistant Encryption and Decryption", de Pankaj Rohatgi), assim como circuitos de potência constante (consulte "Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints", de Thomas Popp e Stefan Mangard) e bibliotecas de células resistentes a SCA (consulte "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation", de Kris Tiri e Ingrid Verbauwhede).

Não se desespere, no entanto. Para cada fonte de ruído ou contramedida, existe uma ferramenta para recuperar pelo menos uma fração do vazamento. Como atacante, seu objetivo é combinar todas essas ferramentas em um ataque bem-sucedido; como defensor, seu objetivo é apresentar contramedidas suficientes que façam com que seu atacante fique sem recursos, como habilidade, tempo, paciência, poder computacional e espaço em disco.

Ataques Mais Poderosos

O que descrevemos até agora sobre análise de potência são na verdade alguns dos ataques mais básicos no campo. Existem uma variedade de ataques mais poderosos, e muitos estão bem além do escopo deste capítulo. No entanto, não queremos deixá-lo no lado errado da curva de Dunning-Kruger do conhecimento real versus conhecimento percebido. Queremos garantir que você tenha conhecimento suficiente para saber que não possui todo o conhecimento.

NOTA: *O efeito Dunning-Kruger é o que ocorre quando você aprende sobre um assunto pela primeira vez e pensa consigo mesmo: "Isso não é tão difícil." David Dunning resumiu brevemente esse efeito da seguinte maneira: "Se você é incompetente, não pode saber que é incompetente. [. . .] as habilidades que você precisa para produzir uma resposta correta são exatamente as habilidades que você precisa para reconhecer o que é uma resposta correta."*

Tudo o que você aprendeu até agora usou um modelo de vazamento. Esse modelo fez algumas suposições básicas, por exemplo, que um maior consumo de energia pode significar que mais fios estão ativados. Um método mais poderoso é o ataque de modelo (veja "Template Attacks", de Suresh Chari, Josyula R. Rao e Pankaj Rohatgi). Em um ataque de modelo, ao invés de assumir um modelo de vazamento, você o mede diretamente de um dispositivo para o qual você conhece os dados (e a chave!) sendo processados. O conhecimento dos dados e da chave fornece uma indicação do poder usado para uma série de valores de dados conhecidos, que é codificado em um modelo para cada valor. Um modelo de

valores de dados conhecidos ajuda a reconhecer os valores de dados desconhecidos no mesmo ou em dispositivos similares.

Fazer tal modelo de modelo significa que você precisa de um dispositivo que possa ser completamente controlado definindo seus próprios valores de chave e permitindo que a criptografia desejada ocorra. A praticidade desse abordagem varia porque pode ser difícil reprogramar o dispositivo alvo, ou você pode ter apenas uma cópia do alvo que você não pode reprogramar para gerar modelos. Em outras ocasiões, como com microcontroladores genéricos, você pode acessar tantos dispositivos programáveis quanto precisar.

A vantagem dos ataques de modelo é que eles operam em um modelo mais preciso do que CPA e, portanto, podem realizar a recuperação de chave em menos traços, possivelmente revelando uma chave de criptografia inteira com apenas uma operação de criptografia. Outra vantagem é que, se o dispositivo que você está atacando estiver executando algum algoritmo não padrão, um ataque de modelo não exige que você tenha um modelo para o vazamento. A desvantagem desses ataques mais poderosos é a complexidade computacional e os requisitos de memória, que são maiores do que uma simples correlação com um peso de Hamming. Portanto, escolher se usar modelos ou outras técnicas, como regressão linear (veja "Univariate Side Channel Attacks and Leakage Modeling", de Julien Doget, Emmanuel Prouff, Matthieu Rivain e François-Xavier Standaert), análise de informação mútua (veja "Mutual Information Analysis", de Benedikt Gierlichs, Lejla Batina, Pim Tuyls e Bart Preneel), aprendizado profundo (veja "Lowering the Bar: Deep Learning for Side-Channel Analysis", de Guilherme Perin, Baris Ege e Jasper van Woudenberg) ou análise de cluster diferencial (veja "Differential Cluster Analysis", de Lejla Batina, Benedikt Gierlichs e Kerstin Lemke-Rust), depende do que é necessário ou disponível em suas circunstâncias de ataque, como ter o menor número de traços, o menor tempo de relógio, a menor complexidade computacional, menor análise humana e qualquer número de outras circunstâncias.

Em termos de dicas mais práticas, Victor Lomné, Emmanuel Prouff e Thomas Roche escreveram "Behind the Scene of Side Channel Attacks — Extended Version", que contém muitas dicas sobre vários ataques. Especificamente, a média de vazamento condicional para CPA pode economizar muito tempo. Você pode encontrar uma implementação disso e vários outros algoritmos como parte do projeto Jlsca de código aberto da Riscure em https://github.com/Riscure/Jlsca/.

No final deste capítulo, discutiremos mais referências.

Medindo o Sucesso

Como medimos o sucesso na vida é um tópico propenso a divagações filosóficas. Felizmente, engenheiros e cientistas têm pouco tempo para divagações, então aqui estão vários métodos que nos permitem medir o sucesso de ataques de análise de canal lateral. Discutiremos diversos tipos de dados e gráficos com os quais você provavelmente se deparará durante suas pesquisas adicionais.

Métricas Baseadas na Taxa de Sucesso

Uma das métricas originais usadas na academia era baseada na taxa de sucesso do ataque. A versão mais básica disso poderia ser testar quantos traços são necessários para um ataque que recupera completamente a chave de criptografia. Geralmente, essa métrica não é muito útil. Se você estiver fazendo apenas um único teste, pode ser que você tenha tido uma sorte excepcional; normalmente, seriam necessários mais traços do que os que você relatou.

Para combater essa situação irrealista, utilizamos gráficos da taxa de sucesso versus número de traços. Primeiro, nos referiremos à taxa de sucesso global (GSR), que fornece a porcentagem de ataques que recuperaram com sucesso a chave completa para um determinado número de traços. A Figura 11-1 mostra um exemplo de gráfico GSR.

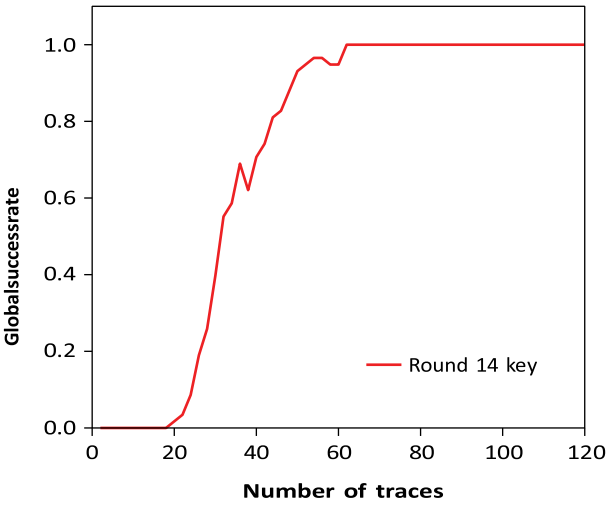


Figura 11-1: Gráfico de exemplo da taxa de sucesso global para um alvo AES-256 vazado

O gráfico na Figura 11-1 mostra que se tivéssemos 40 traços registrados do dispositivo, esperaríamos recuperar a chave de criptografia completa cerca de 80% das vezes. Podemos encontrar essa métrica simplesmente realizando o experimento no dispositivo muitas vezes, idealmente com chaves de criptografia diferentes, caso determinados valores da chave gerem mais vazamento do que outros.

Em vez de usar a GSR, também podemos plotar a taxa de sucesso parcial. Aqui, parcial significa que estamos considerando cada um dos 16 bytes na chave AES128 independentemente dos outros bytes, o que fornece 16 valores, cada um representando a probabilidade de recuperar o valor correto para um byte específico, dado um número fixo de traços.

A taxa de sucesso global pode ser enganosa porque, em algumas implementações específicas, um dos bytes da chave pode não vazar. A GSR, portanto, sempre será zero, uma vez que a chave de criptografia inteira nunca é recuperada, mas os gráficos da taxa de sucesso parcial revelarão se apenas um dos

16 bytes não pode ser recuperado. Então, poderíamos forçar bruta o último byte dentro de 1 segundo, enquanto uma GSR zero não teria revelado uma probabilidade real de recuperar a chave.

Métricas Baseadas em Entropia

As métricas baseadas em entropia são baseadas no princípio de que podemos fazer algumas suposições para recuperar a chave. A chave original AES-128 exigiria, em média, 0,5 × 2^128 suposições para recuperar a chave sem nenhum conhecimento prévio. Este número é tão grande que a chave não pode ser computada antes que o cluster quebrador de chaves seja derretido e/ou consumido pelo sol à medida que se transforma em uma gigante vermelha (cerca de 5 bilhões de anos a partir de agora).

O resultado de um ataque de análise de canal lateral fornece mais informações do que um simples "a chave é XYZ" ou "chave não encontrada". Na verdade, cada suposição de chave tem um nível de confiança associado a ela - a confiança de que uma suposição de chave está correta em relação a um método de análise específico. Na CPA, esse valor de confiança é o valor absoluto da correlação daquela suposição de chave específica. O resultado de um ataque CPA em um byte de uma chave AES-128 é, portanto, uma lista classificada de suposições de chave com níveis de confiança, com nossa melhor suposição no topo e a pior suposição na parte inferior.

Digamos que, usando um ataque de análise de potência, sabemos que o byte de chave real está entre os três primeiros de cada lista. Então, há um total de 3^16 suposições a serem feitas para a chave, o que equivale a cerca de 43 milhões, então isso pode ser facilmente feito em um smartphone. Reduzimos, assim, a entropia. A chave original era uma coleção aleatória de bits, mas agora temos algumas informações sobre o estado mais provável de certos bits e podemos usar isso para acelerar o ataque de força bruta.

O gráfico mais fácil para representar isso é a entropia de suposição parcial (PGE, do inglês "Partial Guessing Entropy"). A PGE faz a seguinte pergunta: após você realizar o ataque com um certo número de traços, quantas suposições de chave foram classificadas incorretamente como mais prováveis do que o valor de chave correto? Se você estiver fazendo suposições de chave para cada byte, você terá um valor de PGE para cada byte da chave; para o AES-128, você acabará com 16 gráficos de PGE. A PGE fornece informações sobre a redução no espaço de busca de chaves feita pelo ataque de canal lateral. A Figura 11-2 mostra um exemplo desse tipo de gráfico.

O gráfico na Figura 11-2 também faz a média de todos os 16 gráficos de PGE para obter um PGE médio para o ataque. A entropia de suposição parcial pode ser um pouco enganosa, pois talvez não tenhamos uma maneira ideal de combinar as suposições em todas as chaves. Por exemplo, se para um byte de chave o valor correto for classificado em primeiro lugar e para um segundo byte de chave for classificado em terceiro lugar, ainda precisamos fazer uma suposição de pior caso e forçar bruta todos os três candidatos principais. No entanto, um ataque de força bruta assim rapidamente se torna impossível se a PGE não for uniforme em todos os bytes.

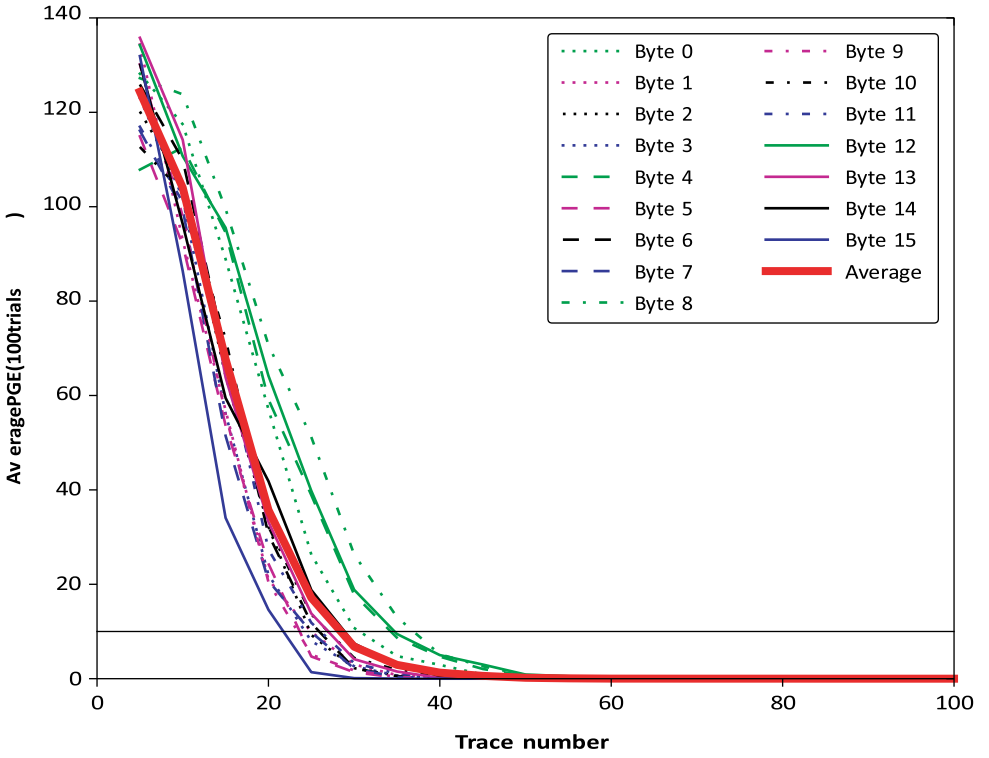


Figura 11-2: Entropia de Suposição Parcial

Algoritmos para combinar idealmente a saída do ataque existem e podem ser usados para gerar uma verdadeira entropia de suposição total (consulte "Security Evaluations Beyond Computing Power", de Nicholas Veyrat-Charvillon, Benoît Gérard, François-Xavier Standaert). A entropia de suposição total fornece detalhes exatos da redução do espaço de suposição da chave que resultou da execução do algoritmo de ataque.

Progressão do Pico de Correlação

Outro formato é plotar a correlação de cada suposição de chave ao longo de um número de traços. Este método é projetado para mostrar a progressão da amplitude dos picos de correlação ao longo do tempo; veja a Figura 11-3 como exemplo. Ela mostra para cada suposição de chave qual é o pico de correlação quando aumentamos o número de traços. Para suposições de chave incorretas, essa correlação tenderá a zero, enquanto para a suposição de chave correta, ela tenderá ao nível real de vazamento.

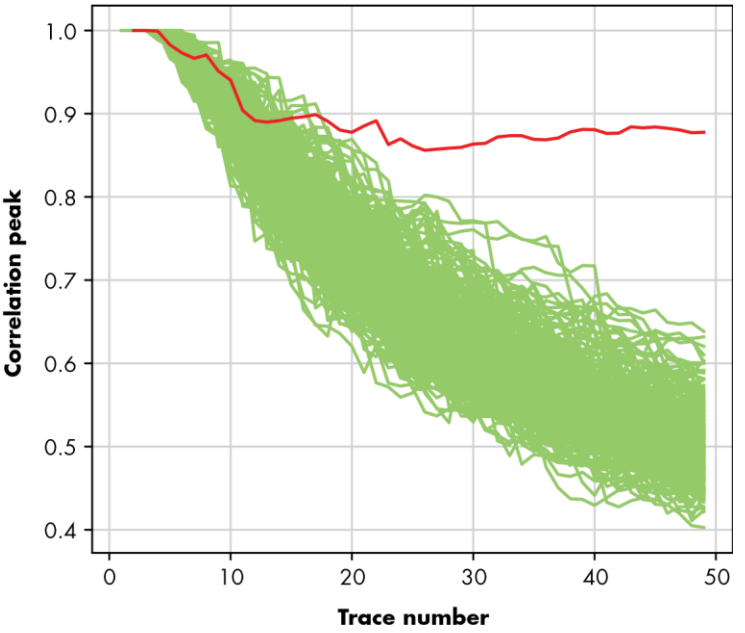


Figura 11-3: Gráficos do pico de correlação vs. número de traços mostram a suposição correta.

Este gráfico remove informações sobre o momento em que o pico máximo de correlação ocorreu, mas agora mostra como esse pico se diferencia das "suposições erradas". O ponto em que o pico correto ultrapassa todas as suposições incorretas é considerado onde o algoritmo foi quebrado. Gráficos da saída de correlação em relação ao número de traços mostram a suposição correta da chave evoluindo lentamente fora do ruído das suposições incorretas da chave.

Uma vantagem do gráfico mostrado na Figura 11-3 é que ele indica a margem entre a suposição incorreta e a correta. Se essa margem for grande, você pode ter mais confiança de que o ataque será bem-sucedido em geral.

Altura do Pico de Correlação

As métricas de sucesso descritas até agora fornecem uma ideia de quão perto você está da extração da chave, mas não ajudam muito na depuração da sua configuração ou abordagem de processamento de traços. Para essas tarefas, há uma abordagem simples: olhar para os traços de saída do algoritmo de ataque, como os traços de correlação para CPA (ou t-traços para TVLA, que discutiremos mais tarde). Esses traços de saída são uma das principais maneiras de melhorar sua configuração ou processamento.

O gráfico que você cria, como na Figura 11-4, destaca todos os traços de correlação das suposições de chave incorretas em uma cor e a suposição de chave correta em outra cor.

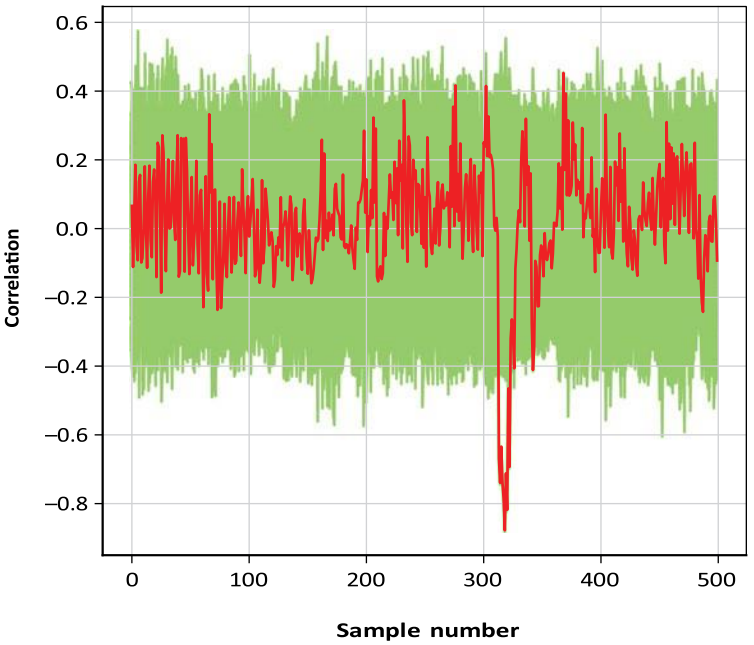


Figura 11-4: Gráfico da saída bruta do algoritmo de ataque

A Figura 11-4 mostra que a suposição correta da chave tem o maior pico de correlação, e também fornece o índice de tempo deste pico. Este gráfico mostra a correlação como uma função do tempo, onde a suposição correta da chave é destacada em cinza escuro na figura, e as suposições incorretas são cinza claro. Sobrepor este gráfico com traços de potência pode ser útil para visualizar onde ocorre o vazamento.

Esse tipo de gráfico é muito útil quando você está otimizando sua configuração. Basta calcular o gráfico antes e depois de alterar um dos seus parâmetros de aquisição ou etapas de processamento. Se o pico ficar mais forte, você melhorou seu ataque de canal lateral; se diminuir, piorou.

Medições em Dispositivos Reais

Quando chega a hora de medir um dispositivo real - não uma plataforma experimental simples projetada para análise de canal lateral - você precisa fazer algumas considerações adicionais. Esta seção as esboça brevemente.

Operação do Dispositivo

O primeiro passo para atacar um dispositivo real é operá-lo. Os requisitos para fazer isso dependem do ataque que você está realizando, mas podemos fornecer algumas orientações gerais e dicas sobre como executar operações criptográficas e escolher quais entradas enviar.

Dispositivos reais podem não fornecer uma função "criptografar este bloco". Parte do trabalho em ataques de análise de canal lateral é determinar exatamente como atacar tais dispositivos. Por exemplo, se estivermos atacando um bootloader que autentica o firmware antes de descriptografá-lo, não podemos simplesmente enviar dados de entrada aleatórios para descriptografar. No entanto, para análise de potência, muitas vezes apenas saber o texto cifrado ou o texto simples é suficiente. Nesse caso, podemos simplesmente alimentar a imagem do firmware original, que passará na verificação de autenticidade e será então descriptografada. Como conhecemos o texto cifrado do firmware, ainda podemos realizar um ataque de potência.

Da mesma forma, muitos dispositivos terão uma função de autenticação baseada em desafio-resposta. Essas funções geralmente exigem que você responda a um valor de nonce aleatório criptografando-o. O dispositivo também criptografará separadamente o nonce. Agora, o dispositivo pode verificar se a resposta de você foi criptografada corretamente, comprovando assim que você compartilha a mesma chave que o dispositivo. Se você enviar ao dispositivo um valor aleatório ou lixo, a verificação de autenticação acabará falhando. No entanto, essa falha é irrelevante; capturamos o nonce e o sinal de energia do dispositivo durante a criptografia. Se coletarmos um conjunto desses sinais, isso pode nos fornecer informações suficientes para um ataque de análise de potência. Implementações adequadas incluirão limitação de taxa ou um número fixo de tentativas para evitar esse tipo de ataque.

Outro problema ao lidar com a comunicação do dispositivo será sincronizar a aquisição. Como demonstrado anteriormente, não nos importamos em encontrar o momento exato em que a criptografia ocorreu, pois o ataque CPA revelará isso para nós (assumindo alinhamento, mas falaremos sobre isso mais tarde). Precisamos estar dentro da vizinhança geral do timing correto (por exemplo, acionando nosso osciloscópio com base no momento em que enviamos o último pacote de um bloco criptografado). Não sabemos quando a criptografia ocorre, mas sabemos que claramente deve ocorrer em algum momento entre o envio desse bloco e o dispositivo enviar de volta uma mensagem de resposta.

Acionar com base no monitoramento das linhas de E/S será mais difícil. Muitas vezes, a maneira mais fácil é implementar um dispositivo personalizado que monitore as linhas de E/S para a atividade relevante. Você poderia programar um microcontrolador simplesmente para ler todos os dados enviados e definir um pino de E/S como alto quando detectar o byte(s) desejado(s), o que por sua vez acionaria o osciloscópio.

Iniciar e capturar a operação é principalmente um obstáculo de engenharia, mas é importante torná-lo o mais estável e livre de jitter possível. Comportamento de temporização instável resulta em ruído de temporização e outros problemas ao longo do tempo, o que pode tornar impossível realizar uma análise adequada dos traços posteriormente.

Repetindo e Separando Operações

Outro truque a lembrar é que se você tiver controle programático sobre seu alvo, ajuda a obter muitas operações em um único traço. Você pode fazer isso tornando o número de vezes que a operação de destino é chamada dentro de um traço uma variável de entrada em seu protocolo. O truque mais simples é colocar um loop em torno da chamada à operação no próprio alvo. Em alguns casos, você pode fazer com que ele faça um loop em um nível mais baixo, dando, por exemplo, a um mecanismo de criptografia AES-ECB um grande número de blocos para criptografar.

Agora, se você realizar aquisições com um número crescente de chamadas à operação de destino (por exemplo, dobrando-o a cada traço), logo começará a ver uma expansão onde as operações criptográficas estão sendo realizadas. Isso acontece porque, embora uma única operação criptográfica possa ser um pequeno sinal invisível, quanto mais operações você realizar, mais tempo elas levarão. Em algum momento, isso se torna visível no seu traço. Então, você pode facilmente identificar o momento da operação e calcular a duração média de uma única operação.

Também pode valer a pena experimentar com um loop de atraso variável (ou slide de nop; nop significa uma operação nula, que efetivamente faz com que o processador não faça nada por um período de tempo muito específico) entre as operações. Uma vez que o truque anterior tenha mostrado o timing, você pode usar essa informação para separar as chamadas de operação individuais, o que pode realmente ajudar a detectar vazamentos, porque o vazamento de uma operação não se mistura então em operações sucessivas.

De Entradas Aleatórias para Entradas Escolhidas

Até agora, temos inserido dados totalmente aleatórios em nossos algoritmos criptográficos, o que proporciona boas propriedades para o cálculo do CPA. Alguns ataques específicos exigem entradas escolhidas, como certos ataques ao AES (consulte "A CollisionAttack on AES: Combining Side Channel- and Differential-Attack" de Kai Schramm, Gregor Leander, Patrick Felke e Christof Paar) ou para a variante de rodada intermediária do teste de avaliação de vazamento de vetor de teste (TVLA) usando o teste t de Welch (mais detalhes na seção "Test Vector Leakage Assessment" mais adiante neste capítulo).

Sem entrar nos detalhes do porquê (falaremos sobre isso depois), durante a aquisição de traços, você pode criar vários conjuntos diferentes, como medições associadas a dados de entrada constantes ou aleatórios, e vários dados de entrada cuidadosamente escolhidos.

Você estará realizando várias análises estatísticas nesses conjuntos, então é crucialmente importante que as únicas diferenças estatisticamente relevantes entre seus conjuntos sejam causadas por diferenças nos seus dados de entrada. Na realidade, campanhas de aquisição de traços que duram mais do que algumas horas terão mudanças detectáveis, talvez no nível médio de potência (consulte a seção "Técnicas de Análise" mais adiante neste capítulo). Se você medir o conjunto A no minuto 0 e o conjunto B no minuto 60, suas estatísticas certamente mostrarão

diferenças de potência entre esses conjuntos. Essas diferenças de potência podem parecer insignificantes até você descobrir que o vazamento suspeito é de fato devido ao seu ar-condicionado ligando no minuto 59 e resfriando o dispositivo de destino, e não devido a um dispositivo de destino com vazamento. Sempre que você fizer análise estatística em vários conjuntos, você deve garantir que não haja correlação acidental com nada além dos dados de entrada. Isso significa que, para cada traço que você mede, você deve selecionar aleatoriamente para qual conjunto deseja gerar entrada. Você também não quer que o alvo saiba para qual conjunto está fazendo uma medição; tudo o que ele precisa saber é sobre os dados nos quais operar. Se você enviar ao alvo informações sobre o conjunto, elas aparecerão nos seus traços. Se você intercalar os conjuntos em vez de escolhê-los aleatoriamente, isso aparecerá nos seus traços. Essas correlações não interessantes são extremamente difíceis de depurar, pois aparecerão como vazamento (falso), então você deve trabalhar duro para evitá-las. Você está detectando mudanças extremamente pequenas na potência, e uma instrução switch executada no alvo com base no conjunto de traços vai ofuscar qualquer vazamento interessante.

A Sonda de Medição

Para realizar o ataque de canal lateral, você precisa medir o consumo de energia do seu dispositivo. Fazer essa medição era trivial ao atacar uma placa de destino que você projetou, mas requer mais criatividade em dispositivos reais. Discutiremos os dois principais métodos: usando um resistor de derivação física e usando uma sonda eletromagnética.

Inserir Resistores de Derivação

Se estiver tentando medir a energia em uma placa "padrão", você precisará fazer algumas modificações na placa para as medições de consumo de energia. Isso variará de placa para placa, mas, como exemplo, veja a Figura 11-5, que mostra como você pode levantar a perna de um pacote fino de montagem em superfície (TQFP, na sigla em inglês) para inserir um resistor de montagem em superfície.

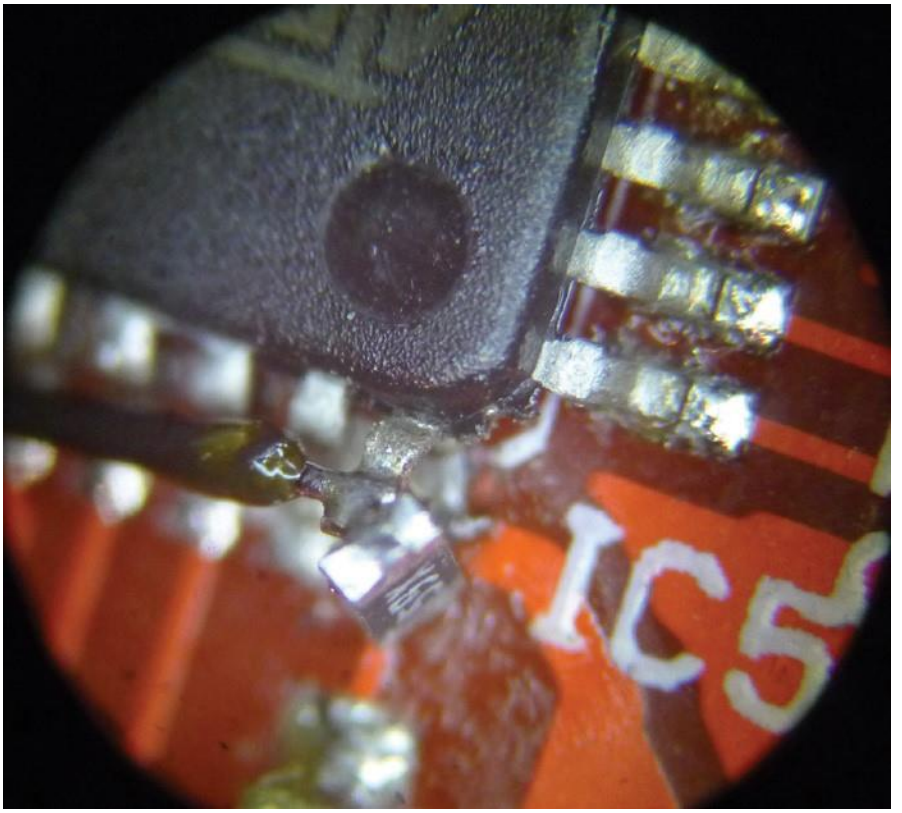


Figura 11-5: Inserindo um resistor na perna de um pacote TQFP

Você então precisa conectar sua sonda de osciloscópio a cada lado do resistor, o que permite medir a queda de tensão através do resistor e, assim, o consumo de corrente de uma rede de tensão específica.

Sondas Eletromagnéticas

Uma alternativa mais avançada é usar uma sonda eletromagnética (também chamada de sonda de campo H, sonda de campo próximo ou sonda de campo magnético), que pode ser posicionada acima ou próxima à área de interesse. A análise resultante é chamada de análise eletromagnética (EMA, na sigla em inglês). A EMA não requer modificações no dispositivo sob ataque, pois a sonda pode ser colocada diretamente sobre o chip ou acima dos capacitores de desacoplamento ao redor do chip. Essas sondas são vendidas em conjuntos conhecidos como conjuntos de sondas de campo próximo e geralmente incluem um amplificador.

A teoria por trás desse método é simples. A física do ensino médio nos ensina que uma corrente fluindo por um fio cria um campo magnético ao redor do fio. A regra da mão direita nos diz que, se segurarmos o fio de modo que o polegar esteja apontando na direção da corrente, as linhas do campo magnético circundarão o fio

na direção dos dedos. Agora, qualquer atividade dentro do chip é simplesmente a alternância de correntes. Em vez de medir a corrente alternada diretamente, sondamos o campo magnético alternado ao seu redor. Isso se baseia no princípio de que um campo magnético alternado induz uma corrente em um fio. Podemos medir esse fio com um osciloscópio, o que reflete de forma indireta a atividade alternada no chip.

Fazendo Sua Própria Sonda Eletromagnética

Como uma alternativa à compra de uma sonda, você pode construir uma sonda EM simples por conta própria. Construir sua própria sonda EM é divertido para toda a família, desde que a família goste de trabalhar com objetos afiados, ferros de solda e produtos químicos. Além da sonda, você precisará construir um amplificador de baixo ruído para aumentar a intensidade do sinal que seu osciloscópio ou outro dispositivo está medindo.

A própria sonda é construída a partir de um cabo coaxial semi-flexível. Você pode adquiri-lo em várias fontes (Digi-Key, eBay) procurando por "cabos SMA para SMA", como o Número da Peça CCSMA-MM-086-8 da Crystek, que está disponível na Digi-Key por cerca de US$10. Cortando este cabo ao meio, você obtém dois comprimentos de cabo semi-flexíveis, cada um com um conector SMA em uma das extremidades (um dos quais é mostrado na Figura 11-6).

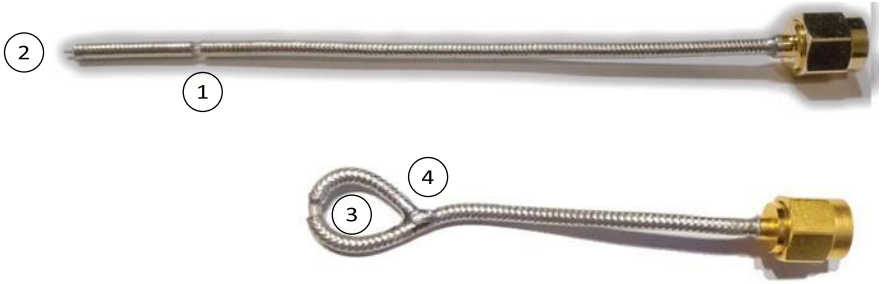


Figura 11-6: Sondas eletromagnéticas caseiras feitas a partir de um cabo SMA semiflexível

Faça uma abertura **1** ao redor de todo o escudo externo. Retire alguns milímetros da extremidade **2**. Arredonde gentilmente isso em um círculo **3**, segurando a fenda com alicate para evitar que o fio condutor interno se dobre. Para completar a sonda básica, solde o círculo fechado **4**, garantindo que o fio condutor interno esteja incluído na conexão de solda entre os escudos externos.

Como o escudo externo é condutor, você pode querer revestir a superfície com um material não condutor, como um revestimento de borracha, como Plasti Dip, ou envolvê-lo com fita autoadesiva.

O sinal captado na pequena lacuna desta sonda será minúsculo, então você precisará de um amplificador para ver qualquer sinal no seu osciloscópio. Você pode usar um CI simples como base para um amplificador de baixo ruído. Ele requer uma fonte de alimentação limpa de 3,3 V, então considere também construir o regulador de voltagem na placa de circuito. Se o seu osciloscópio não for suficientemente sensível, você pode precisar até mesmo encadear dois

amplificadores para obter ganho suficiente. A Figura 11-7 mostra um exemplo de um amplificador simples construído em torno de um CI de $0,50 (número de peça BGA2801,115).

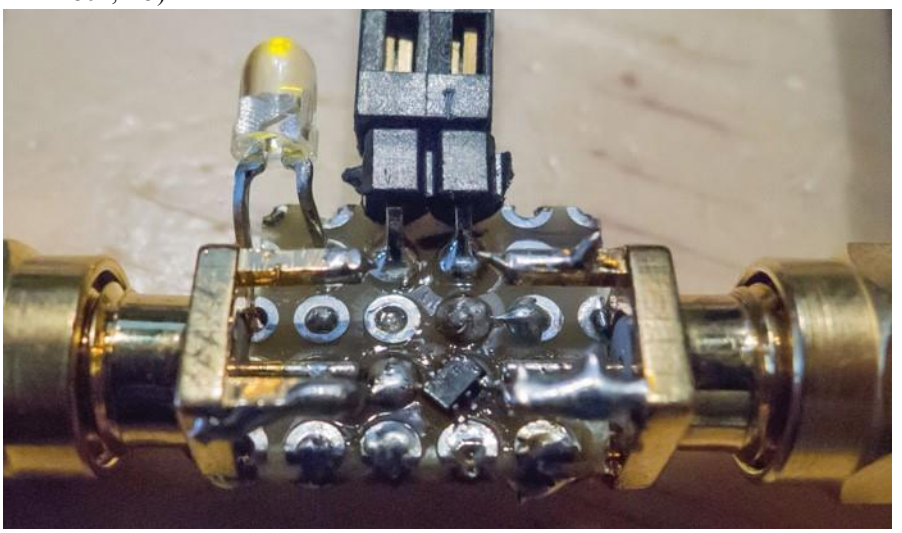


Figura 11-7: Amplificador simples para uma sonda EM

Se quiser construir o amplificador você mesmo, veja a Figura 11-8 para o esquemático.

A escolha da medida de canal lateral pode afetar significativamente as características de sinal e ruído. Geralmente, há pouco ruído ao medir diretamente a energia consumida por um chip, em comparação, por exemplo, ao ruído em uma medição eletromagnética, ou em um canal lateral acústico (consulte "RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis" de Daniel Genkin, Adi Shamir e Eran Tromer), ou em uma medição do potencial do chassi (consulte "Get Your Hands Off My Laptop: Physical Side-Channel Key-Extraction Attacks on PCs" de Daniel Genkin, Itamar Pipman e Eran Tromer). No entanto, uma medição direta de energia significa que você mede toda a energia consumida, incluindo a energia consumida por processos nos quais você não está interessado. Em um SoC, você pode obter um sinal melhor com uma medição eletromagnética se sua sonda estiver cuidadosamente posicionada sobre a localização física do vazamento. Você pode encontrar contramedidas que minimizam o vazamento na medição direta de energia, mas não o limitam na medição eletromagnética, ou vice-versa. Como regra geral, tente EM primeiro em chips complexos e SoCs e tente energia primeiro em microcontroladores menores.

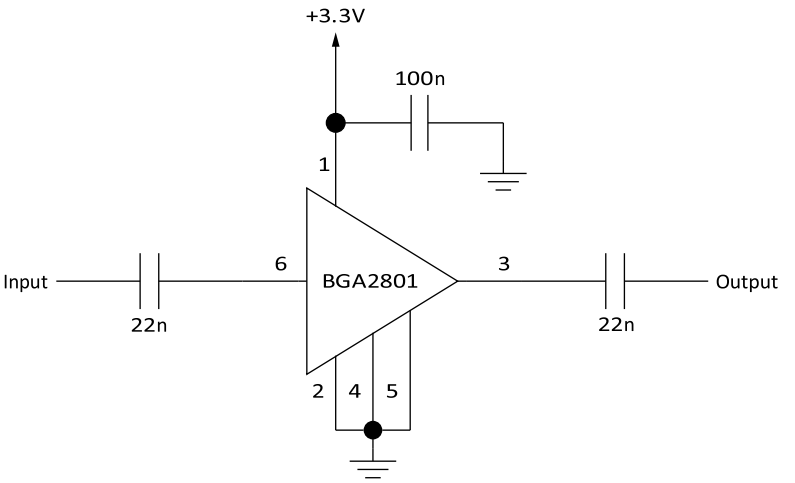


Figura 11-8: Esquemático para um amplificador simples para uma sonda eletromagnética

Determinando Redes Sensíveis

Seja utilizando um resistor de derivação resistiva ou uma sonda eletromagnética (EM), é necessário determinar qual parte do dispositivo deve ser medida. O objetivo é medir o consumo de energia do circuito lógico que realiza a operação sensível — seja um periférico de hardware ou o núcleo de propósito geral que executa um programa de software.

No caso do resistor de derivação resistiva, isso significa olhar para os pinos de energia no CI. Aqui, você precisa medir em um dos pinos que alimentam os núcleos internos, e não nos pinos que alimentam os drivers dos pinos de E/S. Microcontroladores pequenos podem ter uma única fonte de alimentação usada para todas as partes do microcontrolador. Mesmo esses microcontroladores simples podem ter vários pinos de alimentação com o mesmo nome, então selecione aquele que é mais facilmente acessado. Certifique-se de não selecionar uma fonte dedicada à parte analógica, como a fonte de alimentação do conversor analógico-digital, pois isso provavelmente não alimentará os componentes de interesse.

Dispositivos mais avançados podem ter quatro ou mais fontes de alimentação. Por exemplo, a memória, a CPU, o gerador de clock e a seção analógica podem ter fontes de alimentação separadas. Novamente, você pode precisar fazer algumas experimentações, mas quase certamente, a fonte de alimentação desejada será uma das fontes com a palavra CPU ou CORE no nome. Você pode usar os dados que coletou com a ajuda do Capítulo 3 para identificar os alvos mais prováveis.

Se estiver mirando um dispositivo usando uma sonda EM, você precisará experimentar para determinar a orientação e a localização corretas da sonda. Também vale a pena colocar a sonda próxima aos capacitores de desacoplamento que cercam o alvo, já que correntes elevadas tendem a fluir por essas partes. Nesse caso, você precisaria determinar quais capacitores de desacoplamento estão

associados aos componentes principais do dispositivo, de forma semelhante à determinação da fonte de alimentação a ser alvejada.

Deixar seu alvo executar criptografias enquanto exibe capturas de traços ao vivo em uma tela pode ser esclarecedor. Conforme a sonda se move, você verá as capturas de traços variarem drasticamente. Uma boa regra prática é encontrar um local onde o campo seja fraco antes e depois da fase de criptografia e forte durante a rotina de criptografia. Também ajuda exibir um disparador que "acompanha" a operação. Não custa mover a sonda manualmente para ter uma ideia rápida do vazamento em várias partes do chip.

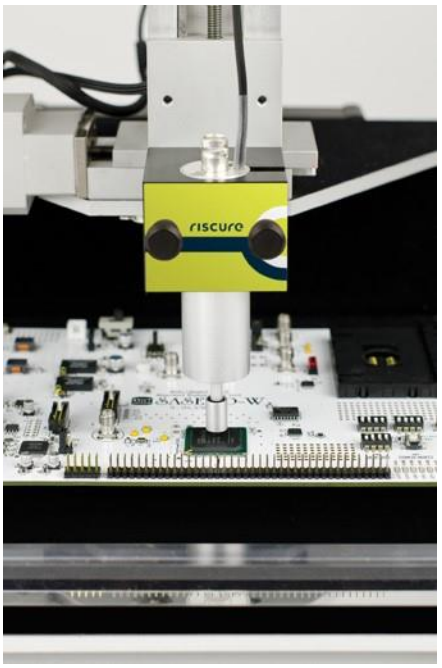
Varredura Automatizada da Sonda

Montar a sonda em um estágio XY e capturar automaticamente traços em várias posições no chip permite uma localização mais precisa de áreas de interesse. A Figura 11-9 mostra uma configuração de exemplo.

Você pode usar o TVLA para obter outra visualização interessante, conforme explicado na seção "Test Vector Leakage Assessment" mais adiante neste capítulo. O TVLA mede vazamentos sem realizar um ataque de CPA, então, se você visualizar o resultado do TVLA, verá um gráfico de vazamento real sobre a área do chip. A desvantagem é que, para calcular os valores do TVLA, você precisa ter dois conjuntos completos de medições para cada ponto no chip, o que aumenta significativamente o comprimento da sua campanha de aquisição de traços.

Sondar mais pontos aumenta as chances de encontrar o local certo, mas diminui sua eficiência. Faça a varredura com uma resolução espacial que forneça gradientes de dados mais contínuos na visualização, garantindo que o tamanho do passo de varredura XY seja menor que a área sensível da sua sonda.

A varredura é de particular interesse quando combinada com a técnica descrita posteriormente neste capítulo na seção "Filtragem para Visualização". Se você conhece a frequência de vazamento da operação de destino, pode visualizar a intensidade do sinal nessa frequência como uma função da posição sobre o chip. Isso leva a imagens interessantes, como a mostrada na Figura 11-10, que mostra uma visualização de varredura XY da intensidade de vazamento em diferentes áreas no chip na banda de 31 a 34 MHz. Esses tipos de imagens podem ajudar a localizar áreas de interesse e podem ser feitos com apenas um traço por localização.



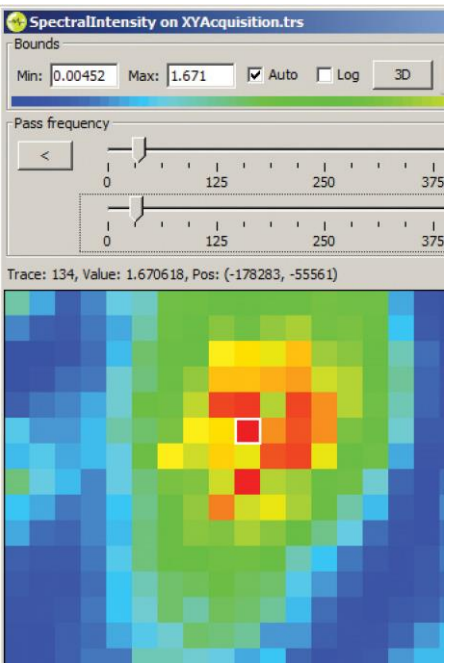


Figura 11-9: Exemplo de uma sonda eletromagnética da Riscure montada em um estágio XY

Figura 11-10: Visualização de varredura XY de áreas de vazamento de um chip

Configuração de Osciloscópio

Um osciloscópio é uma ferramenta ideal para capturar e apresentar os sinais de vazamento de uma sonda magnética. Você precisará configurar seu osciloscópio cuidadosamente para obter boas informações. Discutimos os vários tipos de entrada disponíveis para o seu osciloscópio no Capítulo 2, juntamente com o conselho geral de evitar o uso de sondas que introduzirão considerável ruído em um sinal muito pequeno. Para reduzir ainda mais o ruído, algum tipo de amplificação é frequentemente necessária na entrada do osciloscópio para aumentar o sinal.

Você pode usar um amplificador diferencial para fazer isso, que amplifica apenas a diferença entre os dois pontos de sinal. Além de apenas aumentar o sinal, o amplificador diferencial remove o ruído presente em ambos os pontos de sinal (chamado ruído de modo comum). Na realidade, isso significa que o ruído gerado pela fonte de alimentação será principalmente removido, deixando apenas a variação de tensão que é medida através do seu resistor de medição.

Os fabricantes de osciloscópios vendem sondas diferenciais comerciais, mas geralmente são extremamente caras. Como alternativa, você pode simplesmente construir um amplificador diferencial usando um amplificador operacional comercial (ou op-amp). Uma sonda diferencial pode medir o consumo de energia

através do resistor para reduzir a contribuição de ruído. Um design de código aberto de amostra está disponível como parte do projeto ChipWhisperer, que usa o Analog Devices AD8129. A Figura 11-11 é uma foto dessa sonda em uso em um dispositivo físico.

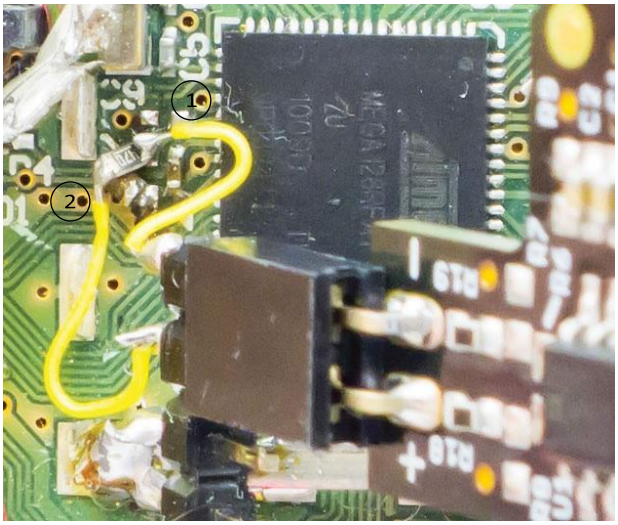


Figura 11-11: Uma sonda diferencial em uso em uma placa de destino

Na Figura 11-11, a sonda diferencial tem um pino positivo (+) e um negativo (–). Esses pinos são marcados no lado inferior direito na serigrafia da PCB preta da sonda. Os fios **2** e **1** conectam o pino positivo e o negativo, respectivamente, a dois lados do resistor de derivação montado na PCB de destino. A sonda diferencial é usada neste exemplo porque a energia fluindo para o resistor de derivação é ruidosa, e queremos remover esse ruído de modo comum.

O esquemático da sonda diferencial é mostrado na Figura 11-12, caso você esteja curioso sobre os detalhes de sua conexão.

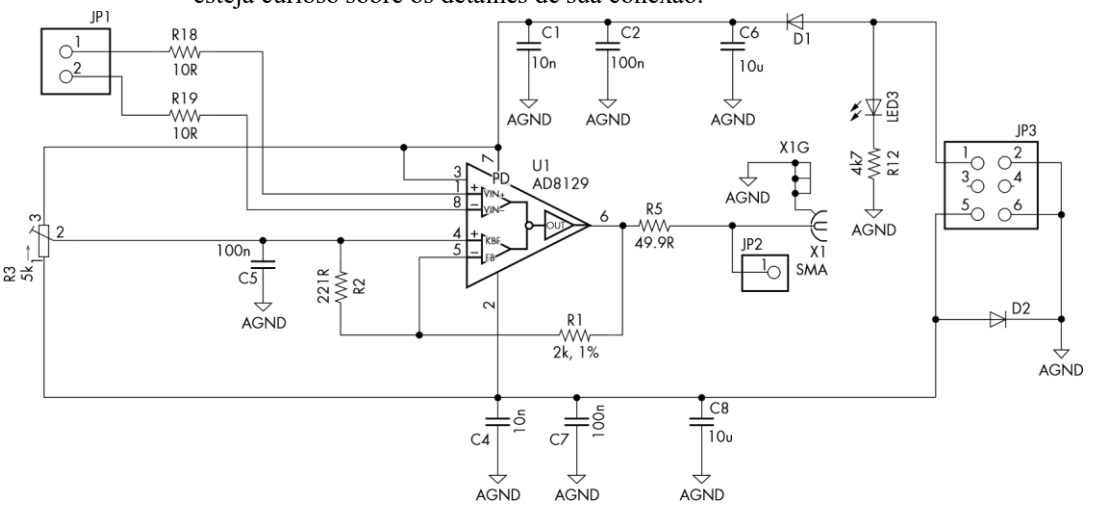


Figura 11-12: Esquemático da sonda diferencial

Taxa de Amostragem

Até agora, supomos que você magicamente conseguiu ler suas medições no computador. Os capítulos anteriores explicaram brevemente que, ao configurar seu osciloscópio, você precisa selecionar uma taxa de amostragem apropriada. O limite superior dessa taxa de amostragem é baseado em quanto você pagou pelo seu osciloscópio; se você tiver dinheiro suficiente, pode comprar dispositivos de 100 GS/s (giga-amostras por segundo) ou mais rápidos.

Mais nem sempre é melhor. Traces mais longos significam muito espaço de armazenamento e tempos de processamento muito mais longos. Você pode querer amostrar a uma taxa muito alta e depois reduzir a taxa de amostragem (ou seja, média de amostras consecutivas) ao armazenar seus dados, o que melhorará consideravelmente suas formas de onda. Primeiro, a redução da taxa de amostragem resulta em um aumento virtual na resolução de quantização do seu osciloscópio. Se o seu osciloscópio tiver um ADC de 8 bits funcionando a 100 MHz e você tirar a média de cada duas amostras, você terá efetivamente um osciloscópio de 9 bits funcionando a 50 MHz. Isso ocorre simplesmente porque, se um valor de amostra de 55 e um valor de amostra de 56 forem médios, eles produzirão 55,5. A inclusão desses valores "meio" efetivamente adiciona 1 bit de resolução. Ou, você pode fazer a média de quatro amostras consecutivas para ter um osciloscópio efetivo de 10 bits a 25 MHz.

Em segundo lugar, a amostragem rápida reduz o desvio de tempo nas medições. Um evento de disparo ocorre em algum momento durante um período de amostragem, e o osciloscópio começará a medir apenas no próximo período de amostragem. O fato de o evento de disparo ocorrer de forma assíncrona com relação ao relógio de amostragem do osciloscópio significa que há desvio de tempo entre o evento de disparo e o próximo período de amostragem. Esse desvio se manifesta como um desalinhamento nos traços.

Considere a situação em que o osciloscópio está amostrando a uma taxa mais lenta, como 25 MS/s, o que significa que as amostras estão sendo obtidas a cada 40 ns. Sempre que o evento de disparo ocorre (ou seja, o início da criptografia), haverá algum atraso até o início da próxima amostra. Esse atraso seria, em média, de 20 ns (metade do período de amostragem), uma vez que a base de tempo do osciloscópio é completamente independente da base de tempo no dispositivo alvo.

Se você amostrar muito mais rápido (digamos, a 1 GS/s), aquele atraso do disparo até o início da primeira amostra será de apenas 0,5 ns, ou 40 vezes melhor! Uma vez que você gravou os dados, pode então reduzi-los para reduzir seus requisitos de memória. A forma de onda resultante terá o mesmo número de pontos como se você realizasse a captura a 25 MS/s, mas agora o jitter é de no máximo 0,5 ns, melhorando consideravelmente o resultado de um ataque de canal lateral (veja Colin O'Flynn e Zhizhang Chen, "Amostragem Síncrona e Recuperação de Clock de Osciladores Internos para Análise de Canal Lateral e Injeção de Falhas").

O downsampling real de uma perspectiva de processamento de sinal digital (DSP) usa um filtro, e qualquer rotina de downsampling integrada a um framework DSP para sua linguagem de escolha ofereceria suporte a isso. No entanto, na

prática, o downsampling pela média de pontos consecutivos, ou mesmo mantendo apenas cada 40º ponto de amostra, tende a manter vazamentos exploráveis.

Alguns osciloscópios podem realizar essa operação para você; alguns dispositivos PicoScope têm uma opção de downsampling que é realizada em hardware. Verifique o manual de programação detalhado do seu osciloscópio para ver se essa opção existe.

Finalmente, você pode usar hardware que captura de forma síncrona com o relógio do dispositivo. No Apêndice A, descrevemos o hardware ChipWhisperer, projetado especificamente para realizar essa tarefa. Alguns osciloscópios terão uma capacidade de referência, que geralmente permite a entrada de até uma referência de sincronização de 10 MHz. Essa capacidade é menos útil na realidade, pois significa que você teria que alimentar seu dispositivo a partir de um relógio de 10 MHz (o mesmo que a referência de sincronização indo para o osciloscópio) para alcançar a capacidade de amostragem síncrona.

Análise e Processamento de Traços

Até agora, a suposição tem sido que você grava traços de energia e depois realiza um algoritmo de análise. Realisticamente, você incluirá uma etapa intermediária: pré-processar os traços, o que significa realizar alguma ação neles antes de passá-los para o algoritmo de análise (como CPA). Todos esses passos visam diminuir o ruído e/ou aumentar o nível do sinal de vazamento. Sua configuração de medição e scripts CPA neste ponto devem ser configurados e esquecidos. O processamento de traços é em grande parte um processo de tentativa e erro e depende da experimentação para encontrar o que funciona melhor no seu alvo. Nesta seção, supomos que você tenha feito um conjunto de medidas de traços, mas ainda não começou CPA.

Quatro técnicas principais de pré-processamento que você pode usar incluem normalização/descarte, ressincronização, filtragem e compressão (consulte a seção "Técnicas de Processamento" mais adiante neste capítulo). Para determinar se sua etapa de pré-processamento está realmente ajudando, primeiro descreveremos algumas técnicas de análise, como calcular médias e desvios padrão, filtragem (sim, novamente), análise de espectro, correlação intermediária, CPA com chave conhecida e TVLA (listadas na ordem típica em que você as aplica). Você não necessariamente precisará de todas elas e, ao fazer análises em uma plataforma experimental simples e vazada que você controla totalmente, provavelmente poderá ignorar a maioria delas completamente. Todas essas técnicas são ferramentas padrão de processamento digital de sinais (DSP), aplicadas em um contexto de análise de energia. Consulte a literatura de DSP para obter inspiração sobre técnicas mais avançadas.

As técnicas de análise se tornam mais valiosas à medida que você se afasta de uma plataforma experimental e passa a realizar medições na vida real, feitas sob situações não ideais. Você usará uma técnica de pré-processamento e então verificará seu resultado usando uma técnica de análise. Se você conhece a chave, sempre pode verificar se seu ataque melhorou usando CPA com chave conhecida

ou TVLA. Se você não conhece a chave, repete o processo até pensar que está pronto para realizar a CPA. Se funcionar, ótimo; se não, você terá que retroceder em cada etapa para descobrir se deve tentar algo diferente. Infelizmente, não é uma ciência exata, mas as técnicas de análise descritas aqui podem lhe fornecer alguns pontos de partida.

Técnicas de Análise

Esta seção descreve algumas técnicas de análise padrão que fornecem uma medida de quão próximo você está de ter um sinal bom o suficiente para CPA. Com CPA, você realizou medições usando diferentes dados de entrada. Muitas das visualizações na seção seguinte devem ser realizadas primeiro com a mesma operação e com os mesmos dados e, posteriormente, você pode usar informações diferentes à medida que se aproxima de um ataque de CPA.

Médias e Desvios Padrão ao Longo de uma Campanha de Aquisição de Dados (por Traço)

Vamos lá! Cada traço pode ser representado por um único ponto, ou seja, a média de todos os valores amostrados nesse traço. Vamos relembrar $t_{d,j}$, onde $j = 0,1,\ldots,T - 1$ é o índice de tempo no traço, e $d = 0,1,\ldots,D - 1$ é o número do traço. A fórmula para o cálculo é:

$$traceavg(d) = \frac{1}{T}\sum_{j=0}^{T-1} t_{d,j}$$

Plotar todos esses pontos mostra mudanças na média dos traços ao longo do tempo e pode ajudá-lo a encontrar anomalias em sua campanha de aquisição de traços; veja, por exemplo, a Figura 11-13.

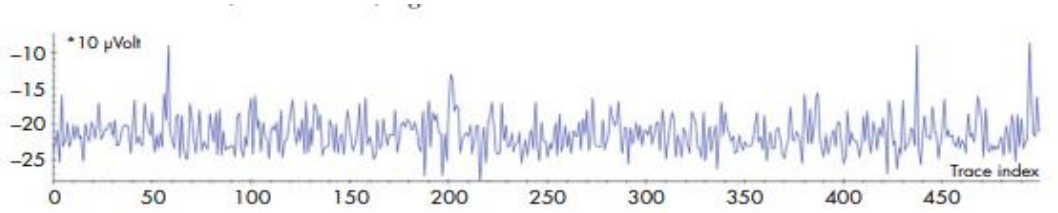


Figura 11-13: Valor médio de todas as amostras por traço, mostrando que os traços 58, 437 e 494 são valores discrepantes.

Um tipo de anomalia é uma média flutuante, por exemplo, devido a mudanças de temperatura (sim, você verá o ar condicionado entrar em funcionamento) ou devido a um valor discrepante completo causado, talvez, por um gatilho perdido. Você pode querer corrigir esses traços ou descartá-los completamente. (Consulte a seção "Normalizando Traços" mais adiante neste capítulo para obter detalhes sobre o que fazer com essa informação.) O desvio padrão lhe dará uma perspectiva diferente sobre a mesma campanha de aquisição. Recomendamos calcular ambos, pois a sobrecarga computacional é insignificante.

Médias e Desvios Padrão ao Longo das Operações (por Amostra)

A outra maneira de calcular uma média é por amostra:

$$sampleavg(j) = \frac{1}{D}\sum_{d=1}^{D} t_{d,j}$$

Essa média pode ajudar a fornecer uma visão mais clara de como é a operação que você está capturando, pois reduz o ruído de amplitude. A Figura 11-14 mostra um traço bruto no gráfico superior e um traço médio por amostra no gráfico inferior.

O traço médio por amostra torna os passos do processo mais óbvios. No entanto, sua utilidade diminui com o aumento do ruído temporal. Um pequeno desalinhamento geralmente não é um problema para a visualização, pois você perde apenas os sinais de alta frequência, mas quanto mais desalinhados estiverem os traços, menor será a frequência mais alta que você poderá ver. Um pequeno desalinhamento pode ser prejudicial para CPA se o vazamento estiver apenas nas frequências mais altas. Você pode usar a média para julgar o desalinhamento visualmente, observando o conteúdo de frequência mais alta.

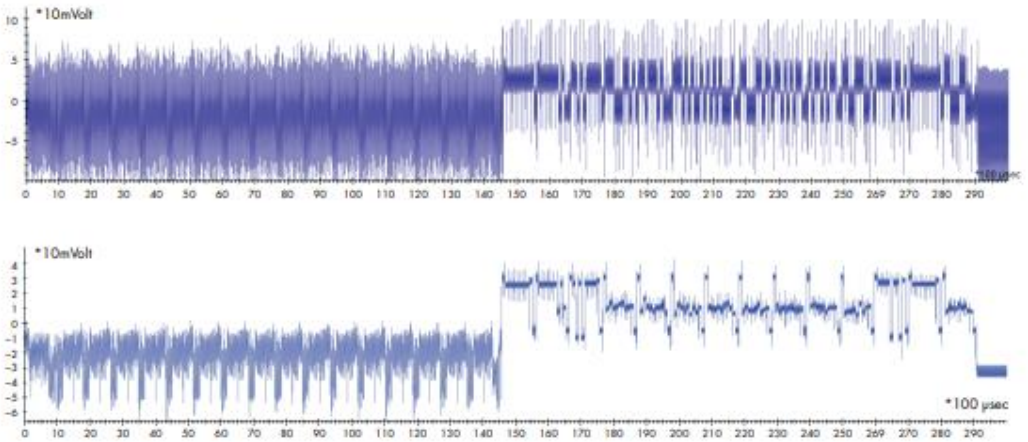


Figura 11-14: Traço bruto (topo) e traço médio por amostra (inferior)

Outro método eficaz é calcular o desvio padrão por amostra. Como regra geral, quanto menor o desvio padrão, menos desalinhamento você tem, como mostrado na Figura 11-15. Neste exemplo, o intervalo entre 300 e 460 amostras tem um baixo desvio padrão, indicando pouco desalinhamento.

Traços perfeitamente alinhados com as mesmas operações ainda podem mostrar diferenças tanto para a média quanto para o desvio padrão, o que se deve a diferenças nos dados e, portanto, é um indicativo de vazamento de dados.

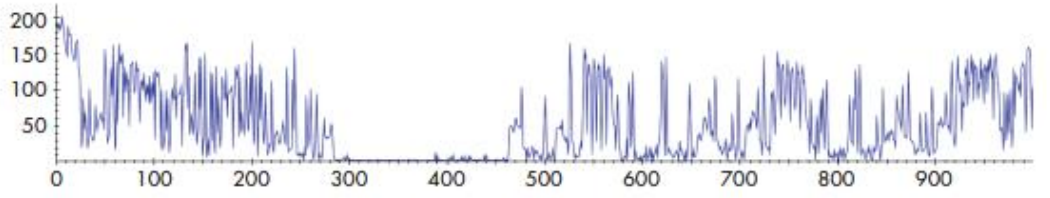


Figura 11-15: Desvio padrão ao longo de um conjunto de traços

Filtragem para Visualização

A filtragem de frequência pode ser usada como um método para gerar representações visuais dos dados de traço. Você pode cancelar agressivamente certas frequências (geralmente frequências altas) para obter uma visão melhor das operações sendo realizadas, sem ter que calcular uma média sobre um conjunto completo de traços. Um filtro passa-baixa simples pode ser implementado tomando uma média móvel sobre as amostras (veja a Figura 11-16). Um filtro passa-baixa é uma maneira rápida de limpar uma representação visual de dados de traço.

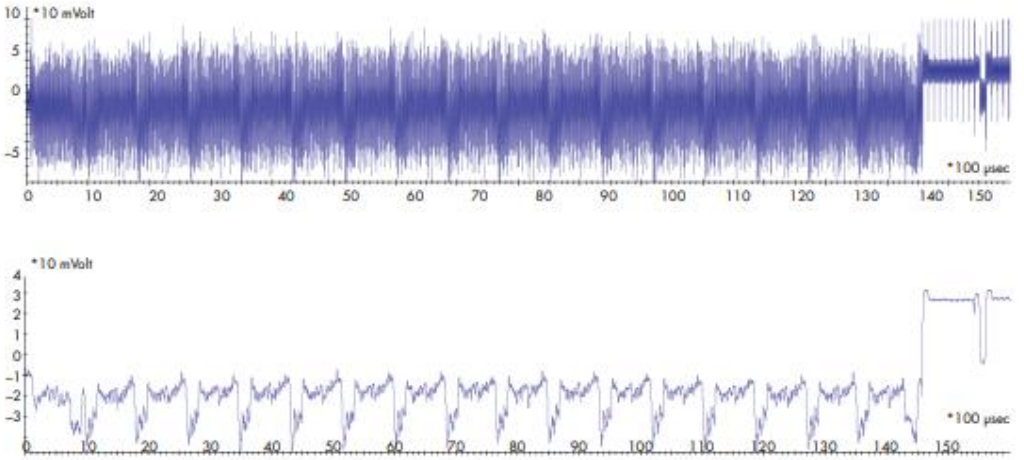


Figura 11-16: Traço bruto (topo) e traço filtrado por passa-baixa (inferior)

Você também pode usar filtros mais precisos e computacionalmente complexos (consulte a seção "Filtragem de Frequência" mais adiante neste capítulo), mas fazer isso pode ser exagerado para fins de visualização. Esta etapa de visualização serve apenas para fornecer uma ideia do que está acontecendo abaixo do ruído; não é uma etapa de pré-processamento, pois provavelmente você removerá também o sinal de vazamento. Uma exceção é para alguns tipos simples de ataques de análise de potência: a visualização de operações dependentes de segredo, como quadrado/multiplicação em RSA, pode comprometer a chave privada!

Análise Espectral

O que você não pode ver no domínio do tempo pode ser visível no domínio da frequência. Se você não sabe o que significa o domínio da frequência, pense em música e som. Se você grava música, captura as informações do domínio do tempo: a pressão do ar causada por ondas sonoras ao longo do tempo. Mas quando você ouve música, percebe o domínio da frequência: diferentes alturas de sons ao longo do tempo.

Duas visualizações geralmente são úteis: o espectro médio, que é o domínio de frequência "puro" sem nenhuma representação para o tempo, e o espectrograma médio, que é uma combinação de informações de frequência e tempo. O espectro mostra a magnitude de cada frequência em um único traço e é um sinal unidimensional. É obtido calculando a transformada rápida de Fourier (FFT) de um traço. O espectrograma mostra a progressão ao longo do tempo de todas as frequências para um único traço. Por adicionar uma dimensão de tempo, é um sinal bidimensional. É calculado fazendo uma FFT sobre pequenos trechos de um traço.

O espectro médio e o espectrograma médio representam a média desses sinais ao longo de todo o conjunto de traços. Quando dizemos que olhamos para a média, queremos dizer que primeiro calculamos o sinal para cada traço individual e depois calculamos a média de todos eles por amostra.

O espectro do chip mostrado na Figura 11-17 tem um clock em torno de 35 MHz, o que pode ser observado a partir dos picos de frequência a cada 35 MHz. Existem picos menores a cada 17,5 MHz, indicando que existem processos repetitivos que levam dois ciclos de clock.

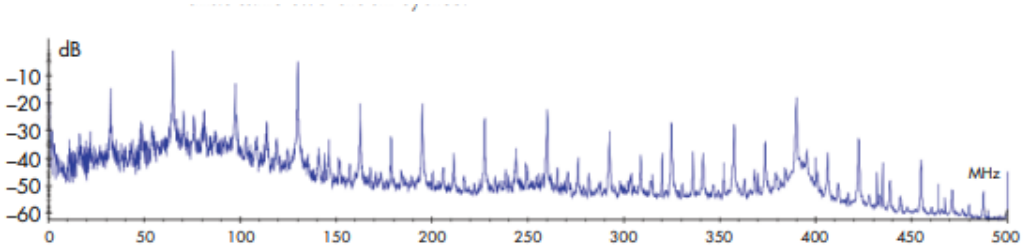


Figura 11-17: Espectro médio sobre todo o conjunto de traços

Você pode realizar algumas análises interessantes. Os picos de frequência a cada 35 MHz são causados por harmônicos de uma onda quadrada a 35 MHz; em outras palavras, eles são causados por um sinal digital que alterna ligado e desligado a 35 MHz. Você sugeriria que este é o clock? Correto. O espectro pode ser usado para identificar um ou mais domínios de clock em um sistema.

Essa análise pode ser particularmente útil se a operação alvo (criptográfica) estiver sendo executada em uma frequência de clock diferente da de outros componentes. Fica ainda melhor quando você faz uma análise diferencial de dois espectros médios. Vamos supor que você saiba que uma seção de tempo do seu traço contém a operação alvo, e o restante do traço não contém. Agora, você calcula independentemente o espectro médio para cada uma das duas seções e subtrai um

do outro; ou seja, você calcula a diferença entre essas duas médias. Você obterá um espectro diferencial, mostrando exatamente quais frequências estão mais (ou menos) ativas durante a operação alvo, o que pode ser um ótimo ponto de partida para filtragem de frequência (consulte a seção "Filtragem de Frequência" mais adiante neste capítulo).

Outra maneira de encontrar a frequência de uma operação é realizar CPA com chave conhecida no domínio de frequência dos traços. CPA com chave conhecida é explicado na seção com o mesmo nome mais adiante neste capítulo, mas em poucas palavras, porque você conhece a chave, você pode determinar o quão próximo um CPA com chave desconhecida está de recuperar uma chave. Para encontrar a frequência de uma operação, primeiro transforme todos os traços usando FFT e depois realize CPA com chave conhecida nos traços transformados. Agora você pode ser capaz de ver em que frequências o vazamento aparece. Você pode fazer o mesmo truque com TVLA. Estes métodos nem sempre funcionam, e você pode precisar de (significativamente) mais traços para obter um sinal.

A vantagem da análise espectral é que ela é relativamente independente de temporização e, portanto, de desalinhamento, pois não estamos observando o componente de fase do sinal. Em vez de ressincronização de traços, você pode realmente fazer CPA no espectro, embora a eficiência dependa do tipo de vazamento (consulte "Análise de Potência de Correlação no Domínio de Frequência" por O. Schimmel et al., apresentado na COSADE 2010).

O espectrograma, que contém informações de temporização, também pode ajudar a identificar eventos interessantes. Se você souber quando sua operação alvo começa, pode ser possível ver certas frequências aparecerem ou desaparecerem. Alternativamente, se você não souber quando a operação alvo começa, pode ser útil observar um ponto no tempo em que o padrão de frequência muda. Veja a Figura 11-18, onde todo o espectro claramente muda em, por exemplo, 5ms e 57ms.

A mudança nas características de frequência do sinal pode ser devido ao início de um mecanismo criptográfico. Ao contrário da análise espectral, você está olhando para informações baseadas em tempo, então esse método de espectrograma é mais sensível ao ruído de temporização.

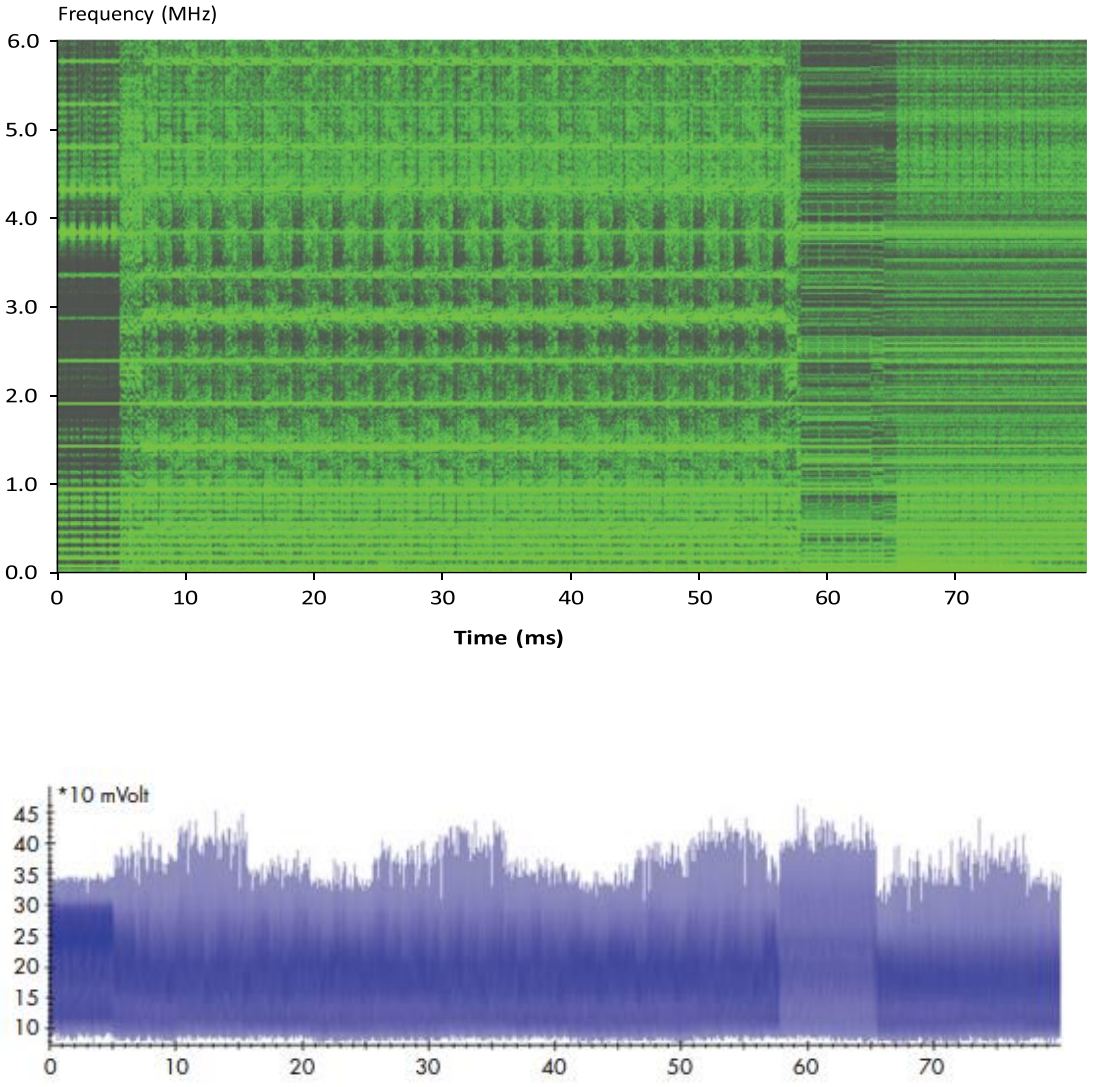


Figura 11-18: Espectrograma sobre uma operação criptográfica (topo) e o traço original (inferior)

Correlações Intermediárias

Agora você sabe que pode usar CPA para determinar chaves calculando um traço de correlação para cada hipótese de chave. Você também pode usar o traço de correlação para outros fins: para detectar outros valores de dados que estão sendo processados pelo alvo, por exemplo, onde o texto simples ou o texto cifrado estão sendo usados em uma operação. Nesta seção, presumimos que você realmente conhece os valores de dados que deseja correlacionar, portanto, nenhum teste de hipótese é necessário. Os candidatos mais imediatos e interessantes são o texto simples e o texto cifrado consumidos e produzidos por um algoritmo de cifragem.

Com valores de dados conhecidos e um modelo de vazamento, você pode correlacionar traços e descobrir se e quando esses valores de dados vazam.

Vamos supor que você tenha uma criptografia AES para a qual você conhece o texto simples de cada execução, e você sabe que ela vaza o peso de Hamming (HW) dos valores de 8 bits. Agora você pode correlacionar o HW de cada byte de texto simples com suas medições e ver quando o algoritmo os consome; isso também é conhecido como correlação de entrada. Dependendo da janela de aquisição de traços, você pode ver muitos momentos de correlação: cada transferência de barramento, cópia de buffer ou outro processamento do texto simples pode causar um pico. No entanto, um desses picos poderia ser o verdadeiro input para o primeiro **AddRoundKey**, logo após o qual você vai querer atacar a operação Substitute.

Outro truque é calcular a correlação com o texto cifrado; isso também é conhecido como correlação de saída. Embora os picos de texto simples possam teoricamente aparecer ao longo de seu traço, os picos de texto cifrado só podem aparecer após o término do processo criptográfico. Portanto, o primeiro pico de texto cifrado indica que a criptografia deve ter ocorrido antes desse pico. Uma boa regra prática é procurar operações criptográficas entre o primeiro pico de texto cifrado e o pico de texto simples imediatamente antes dele.

Observar um pico na correlação do texto cifrado é algo positivo. É uma indicação de que você tem traços suficientes, desalinhamento insignificante e um modelo de vazamento que captura o texto cifrado. Claro, não ver um pico significa que você precisa corrigir qualquer um dos itens acima, e você pode não necessariamente saber qual. A abordagem geralmente é de tentativa e erro. Observe que com CPA, você está atacando intermediários criptográficos, e não texto simples ou texto cifrado. A correlação com o texto simples ou texto cifrado é, portanto, apenas uma indicação de que você tem seu processamento correto; os intermediários criptográficos reais podem precisar de um alinhamento ligeiramente diferente, um filtro diferente ou mais traços.

O último truque de correlação que você pode usar se souber a chave de uma execução criptográfica é a correlação intermediária. Se você conhece a chave, o texto cifrado ou o texto simples, e o tipo de implementação criptográfica, você pode calcular todos os estados intermediários do algoritmo de cifragem. Por exemplo, você pode correlacionar com o HW de cada uma das saídas de 8 bits de MixColumns em AES, para cada rodada. Dessa forma, você deve ver 16 picos para cada rodada, ligeiramente atrasados em relação uns aos outros. Essa ideia pode ser estendida para correlacionar com o HW de um estado de rodada AES inteiro de 128 bits de uma vez, o que funciona em implementações paralelas do AES.

Você também pode usar esse truque para forçar a modelagem de vazamento - por exemplo, não apenas calculando o HW, mas também a distância de Hamming (HD) e vendo qual produz os maiores picos. A desvantagem é que você precisa saber a chave, mas a vantagem é que se você ver picos aqui, está se aproximando de um CPA bem-sucedido. (A razão pela qual você não pode concluir que já chegou lá é porque o CPA se preocupa com "picos corretos" versus "picos incorretos", e nós analisamos apenas "picos corretos" aqui.)

Known-Key CPA

A técnica de CPA com known-key combina os resultados do CPA e dos princípios de entropia parcial de suposição abordados anteriormente neste capítulo para determinar se você realmente pode extrair uma chave. Você calcula um CPA completo e depois usa a análise de EGP (para cada subchave) para analisar o ranking do candidato a chave correto versus o número de traços. Uma vez que você percebe que as subchaves caem estruturalmente no ranking, você sabe que está no caminho certo.

Não se empolgue demais quando apenas algumas de suas chaves caírem para posições muito baixas no ranking. Estatísticas podem produzir resultados estranhos. Elas podem muito bem subir novamente com um conjunto de traços crescente. Somente se a maioria das chaves cair e permanecer em baixa posição, você pode estar em algo promissor. Também observamos o efeito oposto: 9 de 10 bytes de chave em posição 1, enquanto o último demora muito para ser encontrado. Novamente, estatísticas podem produzir resultados estranhos. Somente quando todas as subchaves estiverem em uma posição baixa no ranking você entra no território de poder forçar seu caminho para fora.

Em contraste com a correlação intermediária, este método realmente lhe diz se você pode extrair uma chave. No entanto, a complexidade computacional é significativamente maior; você precisa calcular 256 valores de correlação para cada byte de chave, em vez de um valor de correlação no caso da correlação intermediária. Assim como na correlação intermediária, não ver picos pode ser causado por traços insuficientes, desalinhamento significativo ou um modelo de vazamento ruim. Pode ser necessário tentativa e erro para determinar isso.

Avaliação de Vazamento de Vetor de Teste

O teste t de Welch é um teste estatístico usado para determinar se dois conjuntos de amostras têm valores médios iguais. Usaremos este teste para responder a uma pergunta simples: se você agrupou traços de energia em dois conjuntos, esses conjuntos são estatisticamente distinguíveis? Ou seja, se realizamos 100 operações de criptografia com a chave A e 100 operações de criptografia com a chave B, há uma diferença detectável nos traços de energia? Se o consumo médio de energia do dispositivo em um determinado momento no traço difere para a chave A e a chave B, pode sugerir que o dispositivo está vazando informações.

Aplicamos este teste a um determinado ponto no tempo para cada um dos dois conjuntos de traços de energia. O resultado é a probabilidade de que os dois conjuntos de traços de energia tenham médias iguais naquele ponto no tempo, independentemente do desvio padrão. Criaremos intencionalmente dois conjuntos de traços e, em cada conjunto, o alvo processa diferentes valores. Se esses valores derem origem a mudanças no nível médio de energia, então sabemos que temos vazamento. Consulte a seção "Análise e Processamento de Conjuntos de Traços" anterior neste capítulo para notas sobre aquisição de múltiplos conjuntos e para aprender mais sobre a escolha dos dados de entrada. Não podemos enfatizar o suficiente: se você gerou dois conjuntos executando 100 traços com a chave A e depois sequencialmente depois disso 100 com a chave B, seus traços são inúteis. O

teste estatístico quase certamente encontrará uma diferença entre eles, pois mudanças físicas (como temperatura) são bastante prováveis de ocorrer entre os momentos em que cada conjunto foi capturado. Antes da aquisição de cada traço, decida aleatoriamente no PC (não no alvo) se será com a chave A ou a chave B. Pergunte-nos como sabemos.

PESQUISA ADICIONAL

Para obter mais informações sobre a aplicação deste teste para fins de detecção de vazamentos, "Uma Metodologia de Teste para Validação de Resistência a Canais Laterais", de Gilbert Goodwill, Benjamin Jun, Josh Jaffe e Pankaj Rohatgi, é um bom ponto de partida, e "Metodologia de Avaliação de Vazamento de Vetor de Teste (TVLA) na Prática", de G. Becker et al., é outra excelente referência. O TVLA foi projetado para padronizar a medição de vazamentos de forma que possa ser utilizada em um cenário de certificação de aprovação/reprovação, sem depender das qualidades de um analista de canal lateral individual. Consulte o Capítulo 14 para obter mais informações sobre certificação.

Podemos traçar o valor do teste t de Welch ao longo do tempo e observar picos onde o vazamento é detectado, semelhante a um traço de correlação. O valor do teste t de Welch é calculado por:

$$w_j = \frac{\overline{t_j^A} - \overline{t_j^B}}{\sqrt{\frac{var(t_j^A)}{D^A} + \frac{var(t_j^B)}{D^B}}}$$

onde $\bar{t}jA$ é o valor médio da amostra no tempo jj para o conjunto de traços A, var() é a variância da amostra e DA é o número de traços no conjunto de traços A. Quanto maior for Wj, mais provável é que o conjunto de traços A e o conjunto de traços B sejam realmente gerados por um processo com uma média diferente no tempo j. Em nossa experiência, para conjuntos de traços de pelo menos algumas centenas de

traços, valores absolutos para Wj de, digamos, 10 ou mais indicam que provavelmente há vazamento, e um ataque CPA pode ter sucesso se Wj for 80 ou mais. Em outras literaturas, você frequentemente verá o valor de 4.5, que em nossa experiência resultou em alguns falsos positivos.

Nós vamos te fornecer alguns conjuntos de amostras para AES que você pode testar, para que você tenha uma ideia do que estamos procurando aqui:

Crie um conjunto com dados de entrada aleatórios e outro conjunto com dados de entrada constantes. A ideia é que, se o alvo não vazar informações, as medições de energia dentro do algoritmo de criptografia devem ser estatisticamente indistinguíveis, mesmo que as características dos dados processados claramente variem. Observe que as medições de energia do transporte dos dados de entrada para o mecanismo de criptografia provavelmente vazarão, o que este teste detectará. Obviamente, diferenças nos dados de entrada não são vazamentos reais e não podem ser explorados, então cuidado com picos falsos de t causados por esse "vazamento de entrada".

Crie um conjunto em que um bit de dados intermediário X tenha o valor 0 e outro conjunto em que X tenha o valor 1. Este exemplo é de maior interesse ao testar um bit em uma rodada intermediária do AES, como qualquer bit de estado AES após as operações SubBytes ou MixColumns na rodada 5. Com este teste, não haverá falsos positivos como "vazamento de entrada"; bits na rodada 5 do AES têm efetivamente nenhuma correlação com os bits de entrada ou saída do AES. Se você deseja testar o vazamento de distância de Hamming, também pode calcular o bit X como o XOR entre, por exemplo, a entrada e a saída de uma rodada inteira do AES. Você deve realizar este teste com uma chave conhecida, mas pode fazê-lo com entradas totalmente aleatórias. Como você não sabe qual bit X realmente vaza, pode calcular as estatísticas para todos os bits intermediários imagináveis - por exemplo, para os 3 × 128 bits de estado após AddRoundKey, SubBytes e MixColumns (ShiftRows não inverte bits) na rodada 5.

Crie um conjunto em que o intermediário Y seja A e outro conjunto em que Y não seja A. Esta é uma extensão da ideia anterior. Você pode, por exemplo, testar se um byte de saída do SubBytes tem um viés nas medições de energia quando seu valor é, por exemplo, 0x80. Novamente, você pode calcular o teste t para qualquer Y intermediário e valor A, para que você possa executar 16 × 256 testes para o estado de saída do SubBytes na rodada 5.

Crie um conjunto em que o estado completo da rodada R de 128 bits do AES tenha exatamente N bits definidos como 1 e, em seguida, crie outro conjunto aleatório. Este é um método inteligente. Vamos dizer que escolhemos a rodada R = 5 e geramos um estado de 128 bits com, digamos, N = 16 bits selecionados aleatoriamente definidos como 1. Isso é um viés significativo: sob circunstâncias normais, em média, 64 bits são definidos como 1, e é altamente improvável que o estado enviesado apareça. No entanto,

usando a chave conhecida, podemos calcular qual texto simples teria gerado esse estado enviesado sob essa chave. Devido às propriedades da criptografia, os bytes desses textos simples aparecerão uniformemente aleatórios. Isso vale para o texto cifrado. Na verdade, ao calcular t, o único viés que você pode teoricamente detectar está realmente na rodada R, porque não deve haver nenhum outro viés (exceto por algum viés menor das rodadas R - 1 e R + 1). Portanto, você não obterá picos de t causados pela transferência de texto simples ou texto cifrado. Como você está enviesando um estado de rodada inteira, você pode detectar vazamentos com menos traços do que com os métodos anteriores; portanto, é uma ótima maneira de detectar vazamentos antes que qualquer método CPA possa detectá-los.

Como você pode ver, é possível usar o teste t para detectar vários tipos de vazamento. Note que não especificamos um modelo de energia explícito, o que torna o teste t um detector de vazamento mais genérico do que CPA e similares. O enviesamento de uma rodada interna especialmente amplifica o vazamento. O teste t é uma ótima ferramenta para determinar o momento dos vazamentos, a localização dos vazamentos de EM ou para melhorar os filtros ajustando-os para o maior valor de t. Um truque legal que pode ajudar se houver muito desalinhamento é primeiro fazer uma FFT e depois calcular t no domínio da frequência para descobrir em que frequência está o seu vazamento.

As desvantagens dos testes t são que você pode precisar da chave e que esses testes na verdade não extraem a chave. Em outras palavras, você ainda precisará usar CPA e descobrir um modelo de energia, e talvez não tenha sucesso. Assim como CPA, não ver picos significa que você pode precisar melhorar o processamento de seus traços.

Como você na verdade não está recuperando a chave, também é fácil para o teste t produzir falsos positivos. Isso pode ocorrer porque há uma diferença estatística entre os grupos de traces não relacionada ao vazamento criptográfico (por exemplo, devido a não randomizar adequadamente sua campanha de aquisição). Além disso, o teste t detectará vazamentos relacionados ao carregamento ou descarregamento de dados do núcleo criptográfico, o que pode ser inútil para o ataque. O teste t simplesmente indica se dois grupos têm médias iguais ou diferentes, e você deve entender corretamente o que isso implica. No entanto, é uma ferramenta muito útil para ajustar suas técnicas de processamento: se o valor t aumenta, você está indo na direção certa.

Técnicas de Processamento

Na seção "Técnicas de Análise" anterior neste capítulo, apresentamos alguns métodos padrão que fornecem uma medida de quão próximo você está de ter um sinal bom o suficiente para CPA. Nesta seção, descreveremos algumas técnicas para processar conjuntos de traces. Um conselho prático: verifique seus resultados após cada etapa e duas vezes no domingo. Caso contrário, é fácil cometer um erro e perder o sinal de vazamento para sempre. É mais eficiente em termos de tempo detectar problemas mais cedo do que mais tarde, quando você precisa depurar toda a sua cadeia de processamento.

Normalização dos Traços

Depois de adquirir um conjunto de traces, sempre é útil calcular a média e o desvio padrão por trace, conforme explicado na seção "Médias e Desvios Padrão sobre Operações (por Amostra)" mais cedo neste capítulo. Você verá duas coisas: outliers que em apenas um trace saltarão fora da faixa "normal" e um lento desvio da faixa normal devido a condições ambientais, bem como possíveis erros/falhas na sua aquisição. Para melhorar a qualidade do seu conjunto de traces, você desejará excluir traces que são outliers, permitindo apenas uma certa faixa de valores de média/desvio padrão. Depois disso, você pode corrigir o desvio normalizando as traces. Uma estratégia típica de normalização é subtrair a média por trace e dividir todos os valores de amostra pelo desvio padrão desse trace. O resultado é que cada trace tem um valor médio de amostra de 0 e um desvio padrão de 1.

Filtragem de Frequência

Ao capturar dados com o osciloscópio, podemos utilizar filtros analógicos na entrada do osciloscópio. Esses filtros também podem ser calculados digitalmente: uma variedade de ambientes fornece bibliotecas que permitem facilmente passar traces por filtros. Exemplos incluem scipy.signal para Python e SPUC para C++. Filtros digitais constituem a base da maioria dos trabalhos de processamento de sinal digital, então a maioria das linguagens de programação possui excelentes bibliotecas de filtragem.

Ao fazer filtragem de frequência, seu objetivo é aproveitar o fato de que o sinal de vazamento que você está interessado, ou alguma fonte específica de ruído, pode estar presente em uma parte distinta do espectro de frequência. (A seção "Análise Espectral" mais cedo neste capítulo contém uma descrição de como analisar o espectro para ruído ou sinal.)

Ao passar o sinal ou bloquear o ruído, você pode melhorar a eficácia do CPA. Provavelmente, você deseja aplicar o mesmo filtro aos harmônicos do sinal base; por exemplo, se o seu relógio alvo for de 4 MHz, provavelmente será útil manter 3,9–4,1, 7,9–8,1, 11,9–12,1 MHz, e assim por diante. Se o seu sistema tiver um regulador de comutação adicionando ruído às suas medições, você pode precisar de um filtro passa-alta ou passa-banda para eliminar esse ruído. Frequentemente, a filtragem passa-baixa pode ajudar a aliviar o ruído de alta frequência presente nesses sistemas, mas em alguns casos, o seu sinal de vazamento está inteiramente nos componentes de alta frequência, então a filtragem passa-alta eliminaria qualquer chance de sucesso! Em outras palavras, isso requer algum teste e erro.

Para DPA, você provavelmente estará usando filtros (multi-)notch para passar ou bloquear frequências base e seus harmônicos. O projeto de um filtro de resposta finita ao impulso (FIR) ou de resposta infinita ao impulso (IIR) para filtragem de entalhe pode ser complicado; você sempre pode recorrer ao método mais complexo computacionalmente de fazer uma FFT e, em seguida, bloquear/passar partes arbitrarias do espectro definindo a amplitude para 0 e fazendo uma FFT inversa.

Ressincronização

Idealmente, nós saberíamos quando a operação de criptografia ocorre e acionaríamos nosso osciloscópio para registrar esse momento exato no tempo. Infelizmente, pode ser que não tenhamos um disparador tão preciso, mas, em vez disso, estamos acionando nosso osciloscópio com base em uma mensagem enviada para o microcontrolador. A quantidade de tempo que passa entre o microcontrolador receber a mensagem e realizar a criptografia não é constante, já que ele pode não agir imediatamente na mensagem.

Essa discrepância significa que precisamos ressincronizar múltiplos traços. A Figura 11-19 mostra três traços antes da ressincronização (traços desalinhados) e os mesmos três traços após a ressincronização (traços alinhados).

Os três traços no topo não estão sincronizados. Ao realizar um processo de soma da diferença absoluta (SAD) nos três traços, a saída sincronizada mostra um traço claro na parte inferior.

Ao aplicar o método SAD, você seleciona um traço que servirá como sua referência. Este é o traço ao qual você então alinhará todos os outros. A partir desse traço de referência, você seleciona um grupo de pontos, geralmente alguma característica que aparece em todos os traços. Por fim, você tenta deslocar cada traço de modo que a diferença absoluta entre os dois traços seja minimizada. Este capítulo vem com um pequeno notebook Jupyter (https://nostarch.com/hardwarehacking/) que implementa o SAD e produz. Figura 11-19.

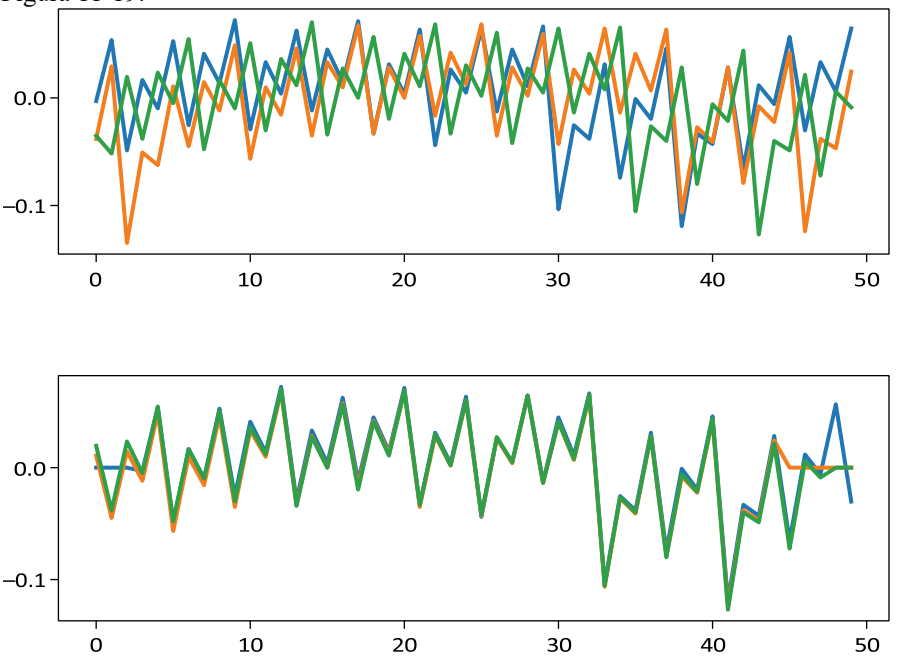


Figura 11-19: Sincronizando traços usando o método de soma da diferença absoluta (SAD).

Uma alternativa é usar o teorema da convolução circular. A convolução entre dois sinais é basicamente a multiplicação ponto a ponto de dois sinais em diferentes deslocamentos n. O valor de n no qual essa multiplicação tem o valor mais baixo é o deslocamento "melhor ajustado" para esses sinais. O cálculo ingênuo é muito caro. Felizmente, você pode obter uma convolução realizando uma FFT em ambos os sinais, multiplicando os sinais ponto a ponto e, em seguida, fazendo uma FFT inversa. Esse processo lhe dará o resultado da convolução entre dois sinais para cada valor de deslocamento n, após o qual você só precisa procurar pelo mínimo.

Vários outros módulos simples de ressincronização podem ser encontrados no software ChipWhisperer. A ressincronização pode se tornar mais avançada do que simplesmente aplicar um deslocamento estático. Você pode precisar distorcer os traços no tempo ou remover seções de um traço onde ocorreu uma interrupção em apenas alguns traços. Não cobrimos esses detalhes aqui, mas consulte "Improving Differential Power Analysis by Elastic Alignment", de Jasper G. J. van Woudenberg, Marc F. Witteman e Bram Bakker, para mais detalhes sobre o alinhamento elástico.

Compressão de Traços

Capturar traços longos pode ocupar muito espaço em disco e memória. Utilizando um osciloscópio de alta velocidade com amostragem na ordem de GS/s ou mais, você rapidamente descobrirá que o tamanho dos seus traços cresce de forma irritantemente grande. Ainda pior, a análise torna-se muito lenta, pois é realizada em cada amostra sucessiva.

Se o objetivo real é encontrar informações de vazamento em cada ciclo de clock, você pode supor que não precisa de cada amostra individual de cada ciclo de clock. Em vez disso, muitas vezes é suficiente manter apenas uma amostra de cada ciclo de clock. Isso é chamado de compressão de traços, pois você reduz significativamente o número de pontos de amostra.

Como mencionado anteriormente na seção "Taxa de Amostragem" deste capítulo, você pode realizar a compressão de traços simplesmente reduzindo a taxa de amostragem, mas fazer isso não resultará em economia tão grande quanto a verdadeira compressão de traços.

A verdadeira compressão de traços utiliza uma função para determinar o valor pelo qual representar cada ciclo de clock. Isso pode ser o valor mínimo, máximo ou médio ao longo de um ciclo de clock inteiro ou apenas de uma parte do ciclo de clock inteiro. Se o dispositivo de destino tiver um oscilador de cristal estável, você pode realizar essa compressão de traços tomando amostras em um determinado deslocamento a partir do gatilho, já que o dispositivo e o relógio de amostragem devem ser estáveis. Para relógios não estáveis, você precisará realizar alguma recuperação de clock, por exemplo, encontrando picos que indiquem o início do clock. Depois de ter o clock, você pode descobrir que apenas os primeiros x por cento de um ciclo de clock contêm a maior parte do vazamento, então você pode ignorar o restante.

Ao comprimir medidas de sonda EM, leve em consideração que o sinal EM é a derivada do sinal de potência. Portanto, para um único pico de potência, haverá um pico EM positivo seguido por um negativo. Você não deseja calcular a média das partes positiva e negativa das ondas capturadas; por sua natureza, elas se cancelam! Nesse caso, você só quer pegar a soma dos valores absolutos das amostras para esse clock.

Aprendizado Profundo usando Redes Neurais Convolucionais

Permanecer relevante requer que um campo como a análise de canal lateral acompanhe as tendências de aprendizado de máquina (ML). Existem na verdade duas formas aparentemente promissoras de enquadrar o problema de canal lateral em termos de aprendizado de máquina: a primeira sendo a análise de canal lateral como uma sequência de etapas realizadas por um agente (inteligente), e a segunda forma sendo a análise de canal lateral como um problema de classificação. Este tópico de pesquisa ainda está em sua fase inicial no momento da escrita, mas é importante. A análise de canal lateral está se tornando cada vez mais importante, e não há pessoas suficientes para acompanhar as demandas do mercado. Qualquer automação, como o aprendizado de máquina, é crucial.

Considerando o quadro do agente: os agentes observam seu ambiente, realizam uma ação e são punidos/recompensados em relação a como suas ações alteram o mundo. Poderíamos treinar um agente para decidir quais passos tomar em seguida, como decidir se deve usar alinhamento, filtragem ou reamostragem com base em quão alto é um pico de T. O futuro dirá se isso é brilhante ou tolo, já que este tópico atualmente não é estudado.

Agora, considere o problema de classificação. A classificação é a ciência de receber um objeto e atribuí-lo a uma classe. Por exemplo, classificadores de aprendizado profundo modernos podem receber uma imagem arbitrária e, com alta precisão, detectar se há um gato ou um cachorro na imagem. As redes neurais usadas para realizar a classificação são treinadas apresentando milhões de imagens que já estão rotuladas como "gato" ou "cachorro". O treinamento significa ajustar os parâmetros da rede de modo que ela detecte características nas imagens representativas de gatos ou cachorros. A parte interessante das redes neurais é que o ajuste acontece puramente por observação; nenhum especialista precisa descrever as características necessárias para detectar "gato" ou "cachorro". (No momento da escrita, ainda são necessários especialistas para projetar a estrutura da rede e como a rede é treinada). A análise de canal lateral é essencialmente um problema de classificação: tentamos classificar valores intermediários a partir dos traços que nos são apresentados. Sabendo os valores intermediários, podemos calcular a chave.

A Figura 11-20 ilustra o processo em que uma rede neural está sendo treinada para realizar análise de canal lateral.

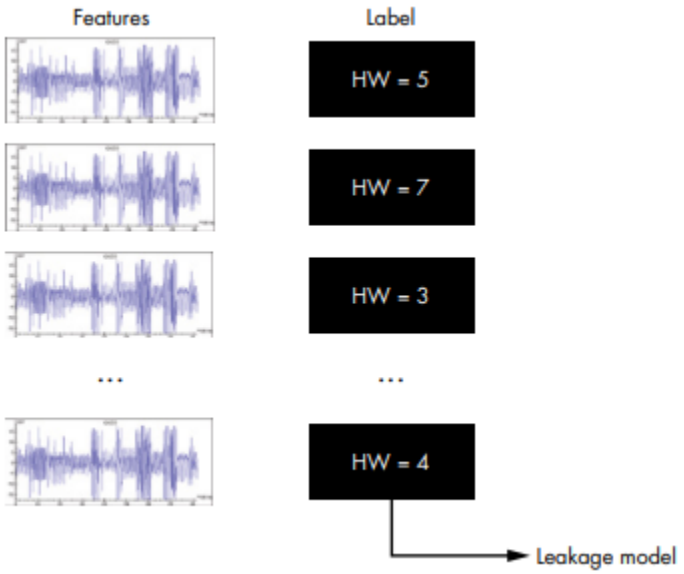


Figura 11-20: Treinando uma rede neural para análise de canal lateral

Substituímos nossos adoráveis gatos e cachorros por um conjunto fofo de traços, que rotulamos individualmente com o peso de Hamming do valor intermediário que estamos visando. Para AES, esse rótulo poderia ser o peso de Hamming de uma saída específica da caixa S. Este conjunto de traços rotulados será o conjunto de treinamento para a rede neural, que então, esperamos, aprenderá a determinar o peso de Hamming a partir de um traço dado. O resultado é um modelo treinado que pode ser usado para atribuir probabilidades sobre os pesos de Hamming para um novo traço.

A Figura 11-21 mostra como a classificação de uma rede pode ser usada para obter valores de confiança para intermediários (e, portanto, chaves).

Este diagrama mostra a rede neural processando um único traço. O traço passa pela rede neural, o que resulta em uma distribuição de probabilidade sobre os pesos de Hamming. Neste exemplo, o peso de Hamming mais provável é 6, com uma probabilidade de 0.65.

Podemos treinar uma rede neural apresentando-a com traços e valores intermediários conhecidos, conforme mostrado na Figura 11-20, e, em seguida, permitir que a rede classifique um traço com um valor intermediário desconhecido, conforme mostrado na Figura 11-21, o que, na prática, é um método de Análise de Poder Lateral (SPA). Uma análise de SPA desse tipo pode ser útil para ECC ou RSA, onde precisamos classificar partes de traços que representam o cálculo sobre um ou alguns bits de chave.

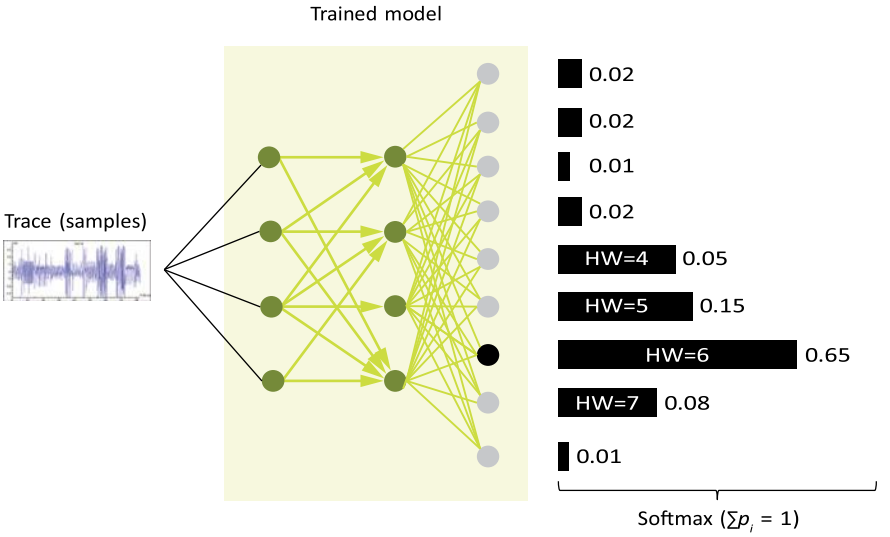


Figura 11-21: Utilizando a classificação da rede para auxiliar na busca por chaves

A abordagem de DPA consiste em usar a distribuição de probabilidade (que é a saída da rede neural) para valores intermediários, transformar essa distribuição de probabilidade em valores de confiança sobre os bytes de chave e atualizar essas confianças para cada traço observado. Aqui é onde divergimos da classificação usual de redes neurais: não nos importamos em classificar cada traço perfeitamente, desde que, em média, viesemos o valor de confiança para o byte de chave relevante. Em outras palavras, não pretendemos identificar perfeitamente um gato ou cachorro em cada imagem, mas temos uma infinidade de imagens extremamente ruidosas de um animal, e tentamos determinar se é um gato.

Redes neurais adequadamente treinadas, especificamente redes neurais convolucionais, detectam objetos independentemente da orientação, escala, mudanças de cor irrelevantes e algum nível de ruído. Portanto, hipoteticamente, essas redes seriam capazes de reduzir o esforço humano analisando traços que precisam de filtragem e alinhamento. Na palestra da Black Hat de 2018 de Jasper, "Lowering the Bar: Deep Learning for Side Channel Analysis" (disponível no YouTube), ele mostra o trabalho de seus coautores Guilherme Perin e Baris Ege. Ele demonstra que as redes neurais são uma abordagem viável para analisar traços de criptografia assimétrica e implementações de software de cifras simétricas onde há desalinhamento e algum ruído. Ainda é uma questão em aberto o quão bem isso se estende à implementação de hardware com contramedidas mais difíceis. Um resultado interessante do trabalho foi que ele quebrou uma implementação mascarada de segunda ordem ao detectar vazamento de primeira ordem com a rede.

O objetivo deste trabalho é eliminar a necessidade de um analista humano interpretar traços. Ainda não alcançamos esse objetivo, embora argumentavelmente tenhamos facilitado ao transferir o esforço para o projeto de redes, em vez das complexidades multidomínio da análise de canal lateral.

Sumário

Na introdução deste capítulo, mencionamos que seria sobre a arte da análise de potência, em contraste com a ciência da análise de potência. A ciência é a parte fácil - apenas tentar entender o que as ferramentas fazem. A arte está em aplicá-las no momento certo da maneira certa ou até mesmo projetar suas próprias ferramentas. Alcançar expertise nesta arte requer experiência, que você só ganhará por meio de experimentação. Para cada nível de habilidade, há alvos interessantes para brincar. Em nosso laboratório, analisamos SoCs multi-GHz, mas isso requer uma equipe de pessoas que já fizeram esse tipo de análise profissionalmente por alguns anos, e pode levar alguns meses para começar a ver qualquer vazamento. No outro extremo do espectro, em apenas algumas horas, somos capazes de ensinar como quebrar a chave em um microcontrolador simples para pessoas sem experiência.

Outro ótimo exercício é construir suas próprias contramedidas. Pegue um alvo com o qual você se sinta confortável em quebrar e que permita carregar seu próprio código. Tente pensar no que realmente tornaria difícil para você, como atacante, quebrar a implementação; um dos truques a empregar é tomar uma das etapas em sua análise e quebrar as suposições que essa etapa faz. Um simples é randomizar o timing do algoritmo, o que quebra o DPA e força você a alinhar as traces. Dessa forma, você melhora a segurança do seu sistema, melhora suas habilidades de atacante e dá a si mesmo algo para fazer no próximo fim de semana.

RECURSOS

Este livro dedicou três capítulos à análise de canais laterais e mal arranhou a superfície dos recursos disponíveis. Coletamos algumas ferramentas e recursos que você pode achar úteis.

"Power Analysis Attacks: Revealing the Secrets of Smart Cards" (Springer, 2010), de Stefan Mangard, Elisabeth Oswald e Thomas Popp, deve ser sua referência principal para mais diversão com análise de canais laterais. Este livro inclui detalhes de ataques mais avançados, como ataques de modelo, e tem vários espaços de trabalho de exemplo disponíveis em http://www.dpabook.org/.

"Serious Cryptography" (No Starch Press, 2018), de Jean-Philippe Aumasson, fornece uma visão geral e detalhes de vários algoritmos criptográficos. Aplicar ataques de análise de canais laterais exigirá compreensão de vários aspectos do algoritmo, e este livro é uma boa referência para a maioria dos algoritmos que você provavelmente encontrará.

A área da análise de canais laterais é um grande campo acadêmico, resultando em pesquisadores universitários e comerciais frequentemente publicando resultados sobre novos ataques e contramedidas. Se você estiver interessado em mais detalhes deste campo, certamente desejará consultar esses recursos acadêmicos.

A oficina sobre Hardware Criptográfico e Sistemas Embarcados (CHES) continua sendo um dos principais eventos na área de análise de canais laterais e segurança de hardware embarcado em geral. Normalmente, é co-localizado com uma conferência sobre tolerância a falhas (FDTC) e provas de segurança (PROOFS), e ocasionalmente co-localizado com uma das principais conferências CRYPTO. CHES geralmente atrai centenas de participantes.

A oficina sobre Análise de Canais Laterais Construtiva e Design Seguro (COSADE) também trata de análise de canais laterais. Esta conferência é muito menor que a CHES, mas tem um forte foco no design seguro de hardware embarcado.

A Conferência de Pesquisa Avançada em Cartões Inteligentes e Aplicações Avançadas (CARDIS) concentra-se na pesquisa de cartões inteligentes, mas isso inclui análise de canais laterais e injeção de falhas. Esta conferência é menor que a CHES.

O CT-RSA é uma faixa específica (a Trilha de Criptógrafos) da principal Conferência RSA, que teve cerca de 42.000 participantes no passado.