Ataques de Canal Lateral em processadores Intel baseados em Aprendizagem Profunda

Guilherme Rodrigues Lourenço
Academia Militar
Lisboa
lourenco.gr@exercito.pt

Ricardo Chaves, Aleksandar Ilic
Instituto Superior Técnico
Lisboa
Ricardo.Chaves@tecnico.ulisboa.pt
aleksandar.ilic@edu.ulisboa.pt

Hoje em dia, as implementações criptográficas estão cada vez mais expostas aos chamados *ataques de canal lateral*, que analisam as fugas (de potência, tempo, etc.) produzidas pela implementação criptográfica, com o objetivo de descobrir informações pessoais. Até há pouco tempo, os *ataques de canal lateral* eram quase exclusivamente realizados com métodos estatísticos tradicionais que visavam as supostas fugas. Ultimamente, porém, os métodos de aprendizagem profunda (AP) têm mostrado grandes resultados em várias áreas da tecnologia, pelo que é natural que possam também ser utilizados para realizar *ataques de canal lateral*.

O objetivo deste trabalho é investigar a possibilidade e capacidade de um *ataque de canal lateral*, utilizando a potência do processador para adquirir as chaves de encriptação utilizadas pelo mesmo. Para o conseguir, vamos implementar um *ataque de canal lateral* num processador Intel, com o objetivo de conquistar acesso à interface *Running Average Power Limit* e de seguida aplicar técnicas de AP, utilizando diferentes arquitecturas de redes neuronais, para descobrir a chave secreta na totalidade.

Nesta dissertação, os dados provenientes das medições do *software* foram adquiridos a partir de versões simplificadas do *Advance Encryption Standard*. Quando colocados em condições de teste, os resultados obtidos pelos modelos de redes neuronais mostraram que ocorrem fugas no processador e que nas mesmas condições em que foram adquiridos os dados utilizados para criar os modelos, é possível descobrir o *byte* secreto pretendido.

Palavras chave: implementações criptográficas, *ataque de canal lateral*, segurança, aprendizagem profunda, redes neuronais

1. Introdução

Algoritmos criptográficos, incluindo encriptações de chaves públicas, encriptações simétricas e funções de hash, criam um conjunto de características que permitem construir mecanismos de segurança que visam objectivos específicos [11]. No entanto, deve assumir-se que os atacantes não irão passar directamente pela complexidade computacional e quebrar as primitivas criptográficas empregados nos mecanismos de segurança. Este tipo de ataque é chamado Ataque de Canal Lateral (SCA), e funciona porque existe uma correlação entre as medidas físicas tomadas durante os cálculos, tais como o consumo de energia, radiação electromagnética, e o tempo de computação e o estado interno do dispositivo de processamento, que por sua vez está relacionado com a chave secreta, que em vários casos, acontece estar fora do controlo dos sistemas de segurança. Vários estudos [7] demonstraram que as técnicas de aprendizagem automática (AA), e mais particularmente as técnicas de AP também têm sido aplicadas à classificação de dados de séries temporais com resultados excepcionais, e mais especificamente, estas técnicas também têm sido aplicadas experimentalmente a SCAs com resultados promissores [2]. Neste trabalho foi possível recuperar um byte de uma chave com apenas 8 powertraces.

2. Introdução Teórica e Estado da Arte

Nesta seção faz-se o enquadramento teórico essencial para se por compreender o trabalho desenvolvido.

2.1. Criptografia

Existem vários serviços criptográficos relevantes para este trabalho, que são garantidos por dois tipos chave: simétrica e assimétrica. Neste trabalho usar-se-à apenas a simétrica.

2.1.1. Criptografia Simétrica

Pelo menos duas partes comunicantes partilham chaves simétricas. As chaves simétricas são mais pequenas, e as operações são mais rápidas do que com chaves assimétricas. A mesma chave é utilizada para encriptar e decifrar. Um dos mais populares algoritmos de chave simétrica é o AES (Advanced Encryption Standard). Este algoritmo utiliza blocos de 128 bits para modos de encriptação de blocos, e as chaves podem ser de 128, 192, ou 256 bits. O algoritmo executa 10, 12, ou 14 rondas de transformação, dependendo do tamanho da chave de encriptação utilizada. Estas transformações são normalmente aplicadas em 16 bytes de dados, representados como uma matriz de 4x4 bytes "estado". Uma ronda é composta por várias operações que devem transformar a entrada numa saída com base nessas transformações. Durante uma ronda, podem ser efectuados quatro tipos de

operações: “AddRoundKey”, “SubBytes”, “ShiftRows”, “MixCollumns”. A última ronda deve ser a única em que não é totalmente igual às outras, uma vez que a última operação não é realizada.

2.2. Ataques de Canal Lateral

Num SCA, o atacante explora as fugas do sistema, com a intenção de recolher informações sobre as operações executadas. Se forem recolhidas informações suficientes, a sua análise posterior revelará a chave secreta.

2.3. Arquitetura do Computador

Cada processador Intel contém uma nova funcionalidade, o RAPL (Running Average Power Limit), que tem um domínio específico, o PP0 (Power Plane 0) que permite não só medir a potência consumida mas também o seu desempenho [1].

2.4. Aprendizagem Profunda: Classificação e Revisão

A aprendizagem profunda é uma subdivisão da aprendizagem automática que utiliza ferramentas e algoritmos baseados em redes neuronais artificiais, também conhecidas como redes neuronais (NN). Estas NNs são inspirados por cérebros biológicos, e os neurónios que os compõem. Cada NN é composto por neurónios, ou nós, que conduzem sinais com informação uns aos outros, tal como os neurónios biológicos transmitem informação através de sinapses.

2.4.1. Redes Neuronais Convolucionais

A rede neural convolucional (CNN) é uma arquitectura de rede neural composta por cinco tipos diferentes de camadas: camadas de entrada, convolução, pooling, totalmente ligadas e de saída, a fim de produzir uma decomposição hierárquica do sinal de entrada.

2.4.2. Redes Neuronais Recorrentes

A rede neural recorrente (RNN) é uma arquitectura de rede neural tipicamente melhor no tratamento de dados que evoluem no tempo, como dados sequenciais ou de séries temporais. Numa RNN, a informação passada é capaz de influenciar os resultados actuais, tomando saídas resultantes de entradas anteriores e inserindo-as novamente numa camada de rede onde as entradas actuais estão a ser processadas.

2.4.3. Redes Neuronais Convolucionais de Memória Longa e Curta

No caso de haver dados recolhidos durante períodos de tempo sucessivos, é habitual caracterizá-los como uma série temporal, e a abordagem de utilizar camadas ConvLSTM é mais do que aceitável. É uma combinação de RNN explicada na Secção 2.4.2 com CNN explicada na Secção 2.4.1 redes. É um modelo baseado em LSTM, onde as camadas de entrada são uma mistura de camadas convolutas com o tempo, oferecendo a capacidade de filtragem das CNN e a memória a longo prazo das LSTM.

2.5. Análise de Potência

Análise de Potência (PA) é um dos ramos de ataques de canal lateral mais conhecidos e utilizados actualmente. Os ataques de análise de potência são o principal foco de investigação actual dos ataques de canal lateral. Este ataque foi inicialmente proposto por Kocher, Jaffe, e Jun em 1999 [6], onde o consumo de energia foi medido com um osciloscópio e utilizado para ultrapassar completamente o algoritmo do Data Encryption Standard (DES). Os vestígios de energia libertam frequentemente quantidades significativas de informação sobre a chave criptográfica processada [11]. Os modelos de consumo de energia (também chamados modelos de fuga) permitem fazer exactamente isso. O modelo de peso Hamming Weight (HW) é o modelo de potência mais simples e o que será utilizado.

2.5.1. Ataques baseados em medições de energia recolhidas através de software

Até recentemente, os ataques de análise de potência tiveram duas limitações [8]: em vez de visar as CPUs mais complexas e de alto desempenho de secretária e de servidor, esse tipo de análise visa os pequenos microcontroladores incorporados; nenhum ataque baseado em software foi aplicado com sucesso, até agora, em x86 para apanhar bits chave criptográficos. Não existem muitos estudos científicos sobre vulnerabilidades introduzidos pela RAPL da Intel e características semelhantes, uma vez que esta área é um caso de estudo relativamente recente. Mais recentemente, O’Flynn et al. [10] foi capaz de alcançar segredos tratados no mundo seguro numa plataforma ARM TrustZone-M usando um ADC a bordo, e Mantel Mantel et al. [9] poderia diferenciar as chaves RSA medindo o consumo de energia nos processadores Intel. Ainda mais recentemente, Moritz Lipp et al. lipp2021platypus mostrou que os SCAs baseados em software são muito poderosos contra as Extensões de Guarda de Software INTEL (SGX), examinando a fuga do canal lateral e sendo capaz de demonstrar a diferença existente entre instruções, o peso de Hamming dos operandos e outros dados, recuperando chaves AES-NI e RSA em cenários de ataque não privilegiados e privilegiados.

2.5.2. SCAs baseados em Aprendizagem Profunda

Analisando a tendência mais recente na área de AP, os trabalhos recentes centraram-se mais nos algoritmos AP como as CNNs [4]. Martinasek et al. compararam métodos baseados em MLPs contra abordagens mais clássicas de SCA, tais como ataques de modelos ou Ataques Estocásticos. O alvo é uma implementação desprotegida do algoritmo AES-128 executado num microcontrolador PIC de 8 bits.

É assumido pelos autores que o modelo de fuga é baseado na etiqueta que é obtida através do cálculo do HW da saída dos valores da S-Box na primeira volta, o que gera nove classes diferentes, ou alcançado com base no modelo de classificação ID (originando 256 classes) da AES-128.

3. Metodologia e Resultados

Esta investigação apoiada por técnicas DL visa explorar a correlação entre o consumo de energia e os dados processados numa CPU, em particular durante a realização de operações criptográficas. Para completar o objectivo principal deste trabalho, é necessário delinear duas fases principais: recolha de dados através de software; avaliar os dados com modelos DL. Para a primeira fase principal, com base em trabalhos anteriores [3] e pesquisa na secção 2.5.1, foi decidido que em vez de se utilizar um framework específico para a aquisição de energia, a funcionalidade RAPL que vem em cada processador Intel será utilizada directamente para ler o módulo de kernel personalizado (o MSRdrv). Espera-se que esta escolha minimize a energia consumida pelo CPU, e diminua o tempo necessário para realizar as aquisições. Para a segunda fase deste trabalho, foi definido que, além dos modelos já utilizados, serão também aplicados novos modelos que, como mostrado no Capítulo 2, poderão obter resultados promissores.

3.1. Descobrir a Frequência de Amostragem

Só depois de descobrir a taxa de amostragem e a taxa de leitura utilizada para as aquisições, é então possível dar o parâmetro correcto ao software utilizado. Neste caso, tentou-se utilizar duas máquinas diferentes: 1) um desktop com um processador Intel I7 10^{th} com frequência de relógio de 3,8 GHz e 2) um portátil mais antigo com um Intel I7 6^{th} com frequência de relógio de 2,6 GHz. Analisando a Figura 1, verifica-se que os intervalos de amostragem na máquina portátil têm uma distribuição muito inferior, e a maioria destes intervalos são inferiores a 100 μs, o que não era o caso da máquina de secretária. Na máquina portátil o intervalo médio de amostragem é de 73,1 μs, o que resulta numa taxa média de amostragem de 16,7 KHz, e o intervalo médio de leitura é de 2,7 μs, o que resulta numa taxa média de leitura de 365,0 KHz.

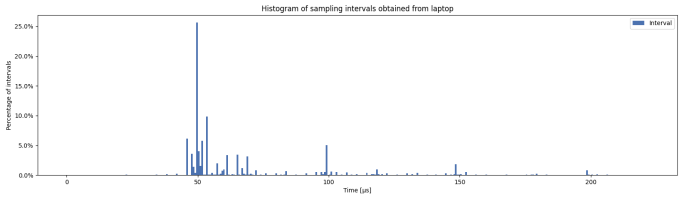

Figura 1: Histograma de Intervalo de Amostragem adquirido no portátil. Taxa média de amostragem de 16,7 KHz e taxa de leitura de 365,0 KHz.

Tendo em conta os dados acima referidos, optou-se por continuar este trabalho e realizar as aquisições na máquina portátil devido ao seu intervalo de amostragem médio inferior quando comparado com o desktop.

3.2. Algoritmo Simplificado baseado em AES

A utilização de um AES simplificado fornece o trabalho de base para tentar avaliar diferentes métodos AP e hiper parâmetros NN, com o objectivo de aumentar gradualmente a complexidade para um cenário de caso real. Está estruturada da seguinte forma: Para aumentar a fuga da S-Box, todos os bytes do texto da placa e todos os bytes da chave são os mesmos e previamente definidos; realiza apenas as duas primeiras operações da primeira ronda da AES, em vez das dez rondas completas regulares, o que proporciona as condições para explorar a fuga da primeira S-Box sem as seguintes operações e rondas que a obscurecem; os bytes serão representados com o modelo de peso Hamming, uma vez que neste caso só podem assumir 9 valores diferentes. As experiências seguintes aumentarão gradualmente os possíveis valores de peso de Hamming disponíveis: dois níveis (0 e 8); três níveis (0, 4, e 8); cinco níveis (0, 2, 4, 6, e 8); e nove níveis (0, 1, 2, 3, 4, 5, 6, 7, e 8).

3.3. Coletar e Processar Dados

Para este conjunto de dados, optámos por separar o número total de amostras adquiridas em blocos de um determinado tamanho, de modo a criar numerosos traços de potência menores e distintos para cada valor de peso de Hamming adquirido, em vez de um único traço de potência grande para cada valor de HW. Há alguns passos importantes e necessários a ter em conta, tendo em conta os valores de consumo de energia adquiridos: 1)Correcção de todos os valores nulos que possam ter sido lidos a partir do registo; 2)Aplicação de uma média móvel a fim de eliminar eventuais picos e outras disparidades; 2)Separar os dados em traços de potência menores com 700 ou 900 amostras de potência por traço de potência; 3)Atribuição de uma etiqueta a cada traço de potência, o valor do peso do Hamming à saída da operação S-Box; 4)Assegurar que todas as etiquetas têm o mesmo número de traços de potência, a fim de ter um conjunto de dados equilibrado; 5)Randomização da ordem dos traços de potência mais pequenos criados, de modo a obter os resultados mais precisos durante o treino do NN; 6)Dividir os traços de potência em dois grupos, com 70% incluindo o conjunto de formação e validação, e 30% incluindo o conjunto de testes;

3.4. Criar e Escolher Modelos de AP

O passo seguinte é a implementação das NNs. Serão implementadas várias arquitecturas diferentes de NNs: a CNN, treinada com 4, 5, 6, 7, e 8 camadas, com 64 kernels por camada; a RNN, treinada com 2, 3, e 4 camadas, com 100 unidades cada; a ConvLSTM, treinada com 1, 2, e 3, com 256 kernels cada um.

Durante este processo de treino há vários hyper parâmetros que podem ser modificados, e cada hyper parâmetro pode ter um impacto maior ou menor durante o treino. Com base em testes rápidos anteriores mais pequenos e em pesquisas anteriores [2, 5], foram definidos e fixados alguns parâmetros de hiper para todas as experiências. Para a avaliação do modelo, as métricas utilizadas foram: classificação de chave, matriz de confusão.

3.5. Nove níveis de Hamming weight

Tentou-se classificar todas as nove classes diferentes, provenientes do algoritmo mais simples.

3.5.1. Treino e Resultados

A matriz de confusão do modelo com melhor desempenho, a CNN com 7 camadas com uma precisão de 81,10% foi calculada. Mais especificamente, para o nível um de HW o modelo foi capaz de prever correctamente 98,10 % das vezes, o que significa que em 420 potência o modelo classificou correctamente 412 deles, que foi a classe com o melhor resultado de precisão;para o nível dois e três de HW, o modelo foi capaz de prever correctamente 60,77% e 64,29% das vezes. Resultou nos piores resultados obtidos. A principal causa para as más previsões no nível dois de HW foi a classificação incorrecta de 82 traços de potência como nível um de HW, e 65 valores como nível zero de HW. Ao analisar as previsões do nível três é evidente que correu mal ao classificar mal este nível como todos os níveis de HW inferiores a si próprio, com 164 más previsões num total de 170. Embora os resultados obtidos não sejam tão bons como se esperava, a divisão geral por classes feita pelo modelo foi bastante boa. É possível ver que quando o modelo está a classificar traços de potência que têm níveis maiores de HW associados, nunca prevê um valor a partir dos níveis mais pequenos de HW.

3.6. Classificação de Chave

A Figura 2 mostra a evolução do Rank médio utilizando este modelo, dados os sucessivos traços de potência para classificação. Como se pode observar, o modelo foi capaz de alcançar um Rank Médio igual a zero após apenas 5 traços, o que significa que com apenas 4500 amostras do conjunto de dados de teste, este modelo foi capaz de recuperar completamente um byte da chave secreta.

Este modelo pode ser utilizado para descobrir qualquer byte de chave secreta através do consumo de energia de um CPU adquirido nas mesmas condições que o conjunto de dados de treino.

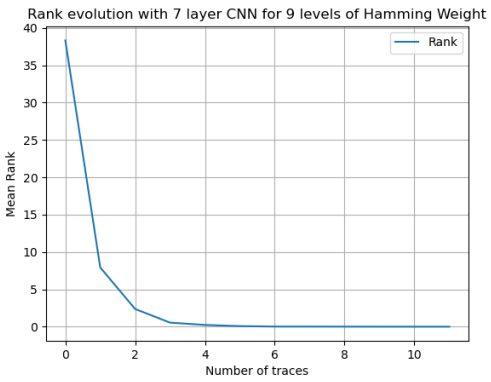

Figura 2: Evolução da Classificação Média do ataque realizado com o melhor modelo CNN treinado na Secção 3.5.

3.7. Resultados e Análises

Toda a informação recolhida foi reunida e apresentada em duas tabelas diferentes, da CNN (Tabela 1), das RNNs e ConvLSTM (Tabela 2).

Os resultados globais obtidos mostrados na Tabela 1 mostram uma boa precisão e não evoluíram de forma completamente uniforme, mas é possível ver que à medida que a complexidade aumentou, os modelos com menos camadas começaram a ter maus resultados. Ao mesmo tempo, quando a complexidade era menor e era utilizado um modelo com mais camadas, a precisão também sofria.

A CNN com quatro camadas teve um grande desempenho na formação e avaliação do conjunto de dados mais simples, conseguindo o melhor resultado de precisão para os dois níveis diferentes de HW. Além disso, a CNN com sete camadas conseguiu alcançar o mesmo resultado, tendo ambos os resultados sido alcançados utilizando 700 amostras por traço de potência. Para os três níveis diferentes de HW, a CNN com melhor precisão foi a CNN com seis camadas e 700 amostras por traço de potência.

Tabela 1: Resultados exactos para diferentes configurações CNN treinadas e testadas.

Hamming Weight	Samples/ Power Trace	CNN Layers 4	5	6	7	8
2	700	**99.7%**	99.1%	99.5%	**99.7%**	96.4%
	900	87.5%	99.0%	99.2%	98.4%	96.0%
3	700	90.3%	90.8%	**94.3%**	93.2%	70.6%
	900	93.6%	94.1%	93.6%	93.5%	71.0%
5	700	67.8%	85.2%	83.0%	**88.7%**	40.7%
	900	74.2%	74.0%	87.3%	86.5%	44.9%
9	700	8.6%	71.5%	72.3%	72.3%	11.0%
	900	12.6%	59.9%	67.1%	**81.1%**	16.0%

Um aspecto que é imediatamente evidente é que todos os resultados utilizando cinco e oito camadas nunca foram os melhores resultados. De facto, a maioria das precisões calculadas parecem melhorar de quatro para cinco camadas, e depois todas as precisões são piores com oito camadas em vez de sete camadas. Isto mostra que para os conjuntos de dados mais complexos, as CNNs com menos camadas não conseguem lidar com a complexidade e as pequenas diferenças que existem entre traços de potência de diferentes valores de HW. No entanto, se a complexidade da CNN aumentar demasiado (como na CNN de oito camadas), o modelo criado poderá ser propenso a sobreajustar-se aos dados de formação, devido a todos os parâmetros de formação disponíveis. Em geral, os modelos com menos camadas não conseguem lidar com os conjuntos de dados mais elaborados, e o modelo com mais camadas adapta-se demasiado ao conjunto de treino, tornando-o ineficaz na fase de treino, ou na fase de ataque. Pela diferença na utilização de 700 amostras ou 900 amostras por traço de potência, parece que os melhores resultados são quase sempre com o conjunto de dados de 700 amostras, mas olhando para casos individuais não há um padrão claro que

nos permita concluir que um conjunto de dados é muito melhor do que o outro.

Passando ao outro tipo de redes neuronais também implementado neste trabalho, a Tabela 2, mostra que nenhum valor de precisão obtido com um RNN foi capaz de superar qualquer uma das precisões das CNNs. Por outro lado, os resultados obtidos com a ConvLSTM para os dois diferentes níveis de HW foram semelhantes aos da CNN, mas para os outros conjuntos de dados os resultados de precisão foram bastante inferiores aos obtidos pelos modelos da CNN.

Tabela 2: Resultados exactos para diferentes configurações RNN e ConvLSTM treinadas e testadas.

Hamming Weight	Samples/ Power Trace	RNN (layers) 2	4	ConvLSTM (layers) 1	2	3
2	700	95.2%	73.7%	97.6%	97.2%	**98.0%**
	900	71.7%	93.8%	97.8%	**98.0%**	97.5%
3	700	79.2%	79.0%	85.3%	**86.1%**	84.3%
	900	61.1%	53.1%	82.7%	83.9%	84.1%
5	700	25.2%	19.4%	19.8%	53.7%	54.5%
	900	25.1%	19.9%	19.5%	19.5%	**56.0%**
9	700	11.0%	11.0%	11.2%	**51.6%**	11.0%
	900	22.8%	10.7%	10%	10%	48.2%

Com o modelo obtido para os nove diferentes níveis de HW, o SCA realizado foi notavelmente bem sucedido, levando cinco traços de potência para descobrir o byte chave, o que neste caso revela toda a chave, uma vez que todos os bytes são iguais. Isto está de acordo com o resultado desejado para esta experiência, uma vez que o byte completo foi recuperado.

3.8. Nove níveis de Hamming Weight

Tentou-se classificar todas as nove classes diferentes, provenientes do algoritmo mais complexo usado neste trabalho.

3.8.1. Treinos e Resultados

A última experiência mostra que para os níveis dois e sete de HW, os NNs foram capazes de prever correctamente 86,50% e 93,74% das vezes, respectivamente, o que significa que em 437 e 431 potência traça o modelo correctamente classificado 378 e 404, respectivamente; para o nível cinco do HW, o NN foi capaz de prever correctamente apenas 6,05% das vezes, o que foi a classe com o pior resultado de precisão. Neste caso, a principal contribuição para esta baixa precisão foi que os traços de potência com o nível cinco de HW foram mais vezes classificados individualmente como qualquer outro nível de HW (excepto o nível sete), em vez de como o nível cinco em si.

Ao contrário da experiência equivalente na Secção 3.5, este modelo não mostra uma separação tão boa dos traços de potência associados a valores de HW maiores e os associados a valores de HW mais baixos. Isto é esperado, uma vez que a primeira ronda completa do algoritmo AES esconde a associação entre o traço de potência e o valor de HW que representa.

3.9. Classificação de Chave

A Figura 3 mostra a evolução do Rank médio utilizando este modelo, dados os sucessivos traços de potência para classificação. Como se pode observar, o modelo foi capaz de alcançar um Rank Médio de zero em menos de 15 traços, o que significa que com menos de 13500 amostras do conjunto de dados de teste este modelo foi capaz de recuperar completamente um byte da chave secreta. Além disso, com este modelo, o SCA apenas necessita, em média, de 7 traços de potência para recuperar o byte da chave.

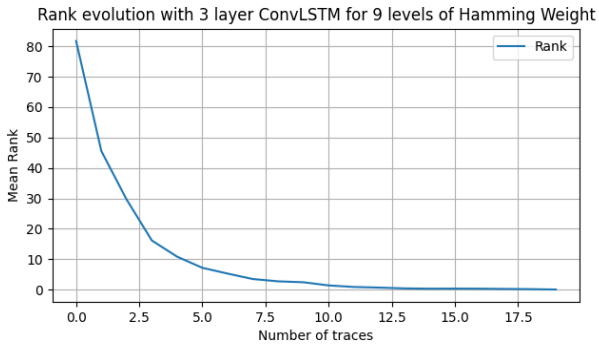

Figura 3: Classificação média do ataque realizado com o melhor modelo ConvLSTM treinado na Secção 3.8.

Este modelo pode ser utilizado para descobrir o byte chave secreto através do consumo de energia de um CPU adquirido exactamente nas mesmas condições que o conjunto de dados de formação.

3.10. Resultados e Análises

Nas Tabelas 3 e 4 encontram-se os valores globais da exactidão obtida são inferiores aos anteriores obtidos ao aplicar as AES simplificadas durante a encriptação.

Tabela 3: Resultados exactos para diferentes configurações CNN treinadas e testadas.

Hamming Weight	Samples/ Power Trace	CNN (layers) 4	5	6	7	8
2	700	**88.6%**	88.5%	88.3%	88.1%	87.7%
	900	87.5%	88.1%	87.5%	86.0%	87.1%
3	700	60.9%	56.7%	29.6%	**66.2%**	65.7%
	900	53.0%	54.0%	60.0%	57.0%	32.5%
5	700	27.2%	18.4%	19.4%	24.2%	24.6%
	900	19.2%	19.0%	26.6%	**30.7%**	30.1%
9	700	25.0%	25.8%	23.1%	**27.7%**	13.2%
	900	19.5%	16.2%	25.6%	18.1%	24.2%

Também se pode observar, com excepção dos dois níveis diferentes de HW, que também obtiveram o melhor valor

de precisão utilizando a CNN com quatro camadas e 700 amostras por traço de potência, que os melhores resultados para as CNNs foram todos obtidos ao aplicar sete camadas utilizando 700 ou 900 amostras por traço de potência. Isto mostra que o aumento da complexidade com o ciclo completo de AES obriga-nos a considerar geralmente a CNN com sete camadas mais capaz de realizar esta tarefa de classificação, e que as arquitecturas mais simples não têm parâmetros treináveis suficientes para capturar toda a informação necessária para classificar correctamente os traços de potência de teste. O que é novidade, são os bons resultados obtidos pelas redes ConvLSTM quando comparadas com as CNNs. Para todos os diferentes níveis de HW, excepto cinco, a ConvLSTM teve um desempenho comparável ao da CNN, ou ainda melhor em alguns casos. Como tal, a melhor precisão para a gama completa de valores de HW com uma ronda completa de AES foi com a arquitectura ConvLSTM, utilizando três camadas com as 900 amostras por conjunto de dados de rastreio de potência. Para a ConvLSTM, também se pode observar que ela beneficia principalmente do processamento de uma série temporal maior, ou seja, de traços de potência de processamento com 900 em vez de 700 amostras, o que pode ser explicado pelas suas capacidades de lidar com dados relacionados com o tempo. Finalmente, a classificação de chave calculada neste capítulo também mostra resultados muito bons, só é necessário menos de 15 traços de potência para uma recuperação de chave correcta utilizando o modelo com melhor desempenho, o ConvLSTM com 3 camadas treinadas com as 900 amostras por traço de potência.

Tabela 4: Resultados exactos para configurações ConvLSTM treinadas e testadas.

Hamming Weight	Samples/ Power Trace	ConvLSTM (layers) 1	2	3
2	700	87.6%	**88.3%**	73.7%
	900	87.3%	87.4%	87.0%
3	700	32.5%	65.1%	68.1%
	900	33.2%	33.6%	**68.4%**
5	700	19.0%	19.0%	19.1%
	900	19.2%	**20.4%**	19.2%
9	700	10.9%	10.3%	11.0%
	900	10.9%	10.3%	**38.7%**

4. Conclusões e Trabalho Futuro

Analisando os resultados preliminares obtidos a partir dos vestígios de encriptação de energia nestas condições, que representa a ligação entre os dados processados e o consumo de energia detectado com o RAPL, podemos concluir que existe e é possível detectar uma fuga, e que poderia eventualmente ser explorada de uma forma que poderia pôr em risco a segurança do sistema.

Há mais passos que podem ser dados para a realização de um SCA sobre o AES, que se relaciona mais de perto com um cenário do mundo real, que não foram implementados neste trabalho. Estes passos seriam, executando o algoritmo AES na sua totalidade, com as dez rondas completas (embora este objectivo possa também ser dividido em passos incrementais menores), aleatorizando todos os 16 bytes do texto da placa ou da chave e aleatorizando todos os 16 bytes tanto do texto da placa como da chave.

Referências

[1] rapl. `https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/advisory-guidance/running-average-power-limit-energy-reporting.html`. Accessed: 2022-01-05.

[2] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas. Deep learning for side-channel analysis and introduction to ascad database. *Journal of Cryptographic Engineering*, 10(2):163–188, 2020.

[3] A. L. N. da Silva. Hacking the systems from within: Using rapl for power analysis. Master's thesis, Instituto Superior Técnico, January 2020.

[4] W. Fischer and N. Homma. *Cryptographic Hardware and Embedded Systems–CHES 2017: 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529. Springer, 2017.

[5] S. Ghandali, S. Ghandali, and S. Tehranipoor. Deep k-tsvm: A novel profiled power side-channel attack on aes-128. *IEEE Access*, 9:136448–136458, 2021.

[6] P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Annual international cryptology conference*, pages 388–397. Springer, 1999.

[7] S. B. Kotsiantis, I. Zaharakis, P. Pintelas, et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.

[8] M. Lipp, A. Kogler, D. Oswald, M. Schwarz, C. Easdon, C. Canella, and D. Gruss. Platypus: Software-based power side-channel attacks on x86. In *IEEE Symposium on Security and Privacy (SP)*, 2021.

[9] H. Mantel, J. Schickel, A. Weber, and F. Weber. How secure is green it? the case of software-based energy side channels. In *European Symposium on Research in Computer Security*, pages 218–239. Springer, 2018.

[10] C. O'Flynn and A. Dewar. On-device power analysis across hardware security domains. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 126–153, 2019.

[11] Y. Zhou and D. Feng. Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. *IACR Cryptol. ePrint Arch.*, 2005:388, 2005.