Universidad de Valencia



**Texto

Descripción generada automáticamente**

**Grado en Ingeniería Electrónica Industrial**

Trabajo de Fin de Grado

**Seguridad en Microcontroladores:**

**Análisis de Side-Channel Attacks y Contramedidas**

**Autor: ABDALAH EL KENNOUSSI**

**Tutor: José Rafael Lajara**

**Valencia, Octubre – 2023**

# DECLARACION DE AUTORÍA

**Yo, ABDALAH EL KENNOUSSI declaro la autoría del Trabajo Fin de Grado titulado “Seguridad en Microcontroladores: Análisis de *Side-Channel Attacks* y Contramedidas” y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual. El material no original que figura en este trabajo ha sido atribuido a sus legítimos autores.**

**Valencia, 1 de enero de 2023**

**Fdo:**

# RESUMEN

En los microcontroladores existen diversos tipos de ataques. Algunos se pueden clasificar como directos y hacen operaciones como un volcado de memoria por JTAG. Otros buscan mecanismos indirectos, son los *side-channel attacks*.

En este proyecto haremos el estudio de dos tipos de ataques indirectos y propondremos unas contramedidas para evitarlos. Nos centraremos en el “*Power Analysis*” y “*Clock Glitching*” que generalmente son los ataques más conocidos por su eficacia a la hora de deducir contraseñas y códigos encriptados.

El primer paso es estudiar el funcionamiento del sistema “CHIPWHISPERER LITE” y entender su funcionamiento con los dos tipos de ataques. A partir de aquí se planteará y ejecutarán los dos tipos de ataques para ver cómo reacciona nuestro *Target*, que en este caso es el ARM CORTEX-M.

Una vez realizados los ataques y obtenidos los resultados se procederá a realizar un estudio de prevención en contra de los dos tipos de ataques y se pondrán a prueba para ver si surquen efecto.

# AGRADECIMIENTOS

Quiero expresar mi sincero agradecimiento a todos los profesores por el apoyo y el gran trabajo y conocimientos aportados durante la carrera, pero en especial a José Rafael Lajara que ha sido mi tutor y apoyo durante la realización de este trabajo de fin de grado. Su experiencia y conocimientos en este sector me han servido de guía durante todo el proceso de investigación y redacción de mi tesis. Su compromiso con la excelencia académica y su disposición a invertir tiempo y esfuerzo en este proyecto me han ayudado a mejorar muchas cualidades en este sector y cada vez interesarme más.

Además, quiero agradecerle por su paciencia y disponibilidad para responder a mis preguntas, aclarar mis dudas y ofrecer sugerencias constructivas que han mejorado significativamente este proyecto.

Espero tener la oportunidad de mantener el contacto y continuar aprendiendo de usted en el futuro. Gracias una vez más por su apoyo inquebrantable.

# Contents

[DECLARACION DE AUTORÍA 3](#_Toc157262031)

[RESUMEN 5](#_Toc157262032)

[AGRADECIMIENTOS 7](#_Toc157262033)

[Contents 9](#_Toc157262034)

[1 Introducción 11](#_Toc157262035)

[1.1 Introducción 11](#_Toc157262036)

[1.2 Motivación 11](#_Toc157262037)

[1.3 Objetivos 12](#_Toc157262038)

[2 Materiales y métodos 15](#_Toc157262039)

[2.1 ChipWhiperer Lite 32 Bits con Target Device Incorporado 15](#_Toc157262040)

[2.2 Máquina Virtual de ChipWhisperer 18](#_Toc157262041)

[2.3 Chipwhisperer Lite 32 BITS 20](#_Toc157262042)

[2.4 Tipos de targets 20](#_Toc157262043)

[3 ESTUDIO TEÓRICO 23](#_Toc157262044)

[3.1 Tipos de side channel attacks 23](#_Toc157262045)

[3.1.1 Análisis de potencia (*Power Analysis*) 23](#_Toc157262046)

[3.1.2 Análisis de tiempo (*Clock Glitching* o *Timing Analysis*) 27](#_Toc157262047)

[3.1.3 Otros 27](#_Toc157262048)

[4 Trabajo práctico 29](#_Toc157262049)

[4.1 Ataques simples de Power analysis 29](#_Toc157262050)

[4.2 Ataques simples de Clock glitching 29](#_Toc157262051)

[4.3 Implementación y prueba de contramedidas para ataques de power analysis 29](#_Toc157262052)

[4.4 Implementación y prueba de contramedidas para ataques de clock glitching 29](#_Toc157262053)

[5 Conclusiones 31](#_Toc157262054)

[6 Bibliografía 33](#_Toc157262055)

[7 Anexos 35](#_Toc157262056)

[7.1 Código Fuente X 35](#_Toc157262057)

[7.2 Código Fuente Y 35](#_Toc157262058)

# Introducción

## Introducción

Los *Side Channel Attacks* son un tipo de ataque informático que explora las fugas involuntarias de información que se producen durante el funcionamiento de un sistema, un ordenador o cualquier dispositivo electrónico [1][2][3][4]. A diferencia de los ataques tradicionales, que se centran en vulnerabilidades en el software o hardware, este tipo de ataques se basan en la observación de señales indirectas o canales colaterales, como el consumo de energía, el tiempo de ejecución o las emisiones electromagnéticas para descifrar información encriptada, de ahí su nombre.

Estos ataques pueden tener como objetivo la obtención de datos confidenciales, como claves de cifrado, contraseñas o información personal, sin acceder directamente a la fuente de esos datos. Los *Side Channel Attacks* pueden ser especialmente peligrosos en entornos donde la seguridad de la información es crítica, como sistemas de criptografía, sistemas de pago electrónico o dispositivos médicos.

Los investigadores y expertos en ciberseguridad trabajan constantemente en la detección y mitigación de los *Side Channel Attacks*, desarrollando técnicas de contramedidas, como el enmascaramiento de datos, la introducción de ruido en las señales o el diseño de hardware resistente a este tipo de ataques. La comprensión de las amenazas asociadas a estos ataques es esencial para garantizar la integridad y la confidencialidad de los sistemas y datos en un mundo cada vez más conectado y dependiente de la tecnología.

Los dos ataques más comunes y que además se van a tratar en este proyecto son: “*Power Analysis Attack*” y “*Clock Glitching Attack*”.

Los ataques por análisis de potencia [5] se basan en que el atacante estudia el consumo de energía de un dispositivo hardware criptográfico para poder desencriptarlo. En cambio, para los ataques de reloj podemos manipular el reloj que se presenta al dispositivo para provocar un comportamiento no deseado.

En general, los ataques por canales laterales se aprovechan de la implementación hardware de un sistema informático en algoritmos como el *Advanced Encrypton Standard* (AES) o el *Triple Data Ecryption Algorithm* (DEA) para explotarlos por medio de diferentes ataques laterales. Además, es común que estos ataques vayan dirigidos a microcontroladores.

## Motivación

Investigar el tema de los *Side Channel Attacks* es fundamental en el mundo actual de la ciberseguridad. Estas amenazas, que se basan en el análisis de información indirecta, representan un desafío constante y una oportunidad de mejora en la protección de datos y sistemas.

En la era digital en la que vivimos, la información es el recurso más valioso. La seguridad de esta información es esencial para proteger nuestras vidas, nuestra privacidad y nuestras instituciones. En este contexto, la investigación de los *Side Channel Attacks* se vuelve una tarea crítica.

Al comprender en profundidad cómo funcionan los *Side Channel Attacks* podemos desarrollar contramedidas efectivas para prevenirlos y proteger nuestros sistemas y datos. La investigación en esta área nos permite anticiparnos a los posibles agresores y fortalecer nuestras defensas.

Además, la investigación en *Side Channel Attacks* fomenta la innovación en la seguridad cibernética. A medida que los atacantes desarrollan nuevas técnicas, los investigadores también deben estar un paso adelante, diseñando sistemas más robustos y seguros. Esta constante competencia es lo que impulsa el avance de la tecnología y nos permite mantenernos a salvo en un mundo cada vez más interconectado.

En resumen, la investigación en *Side Channel Attacks* es crucial porque nos ayuda a proteger nuestra información, promueve la innovación en seguridad cibernética y puede tener un impacto positivo en muchas áreas de la tecnología. Al trabajar juntos para comprender y mitigar estas amenazas, podemos garantizar un futuro digital más seguro y confiable. Se espera que este trabajo sirva como base para otros futuros trabajos de investigación que avancen en nuevos conocimientos de este tipo de ataques y contramedidas en nuevos *targets*.

## Objetivos

En este TFG, nuestro objetivo es estudiar dos tipos de ataques laterales: el *Power Analysis Attack* y el *Clock Glitching*.

Para este estudio emplearemos el hardware CHIPWHISPERER LITE como herramienta de generación y análisis de ataques y nuestro *Target* víctima será un microcontrolador ARM CORTEX – M.

El objetivo principal es realizar los dos ataques utilizando nuestro micro y ver cómo reacciona nuestro *Target*. Una vez obtenidos los resultados, se estará en condiciones de proponer contramedidas y ponerlas a prueba para evitar este tipo de brechas.

Hay varios sub-objetivos, como son:

* Mejorar la seguridad informática: estudiar los *Side Channel Attacks* permite comprender las vulnerabilidades de los sistemas y dispositivos electrónicos frente a técnicas de ataque que van más allá de los métodos convencionales. Al entender cómo se pueden explotar canales secundarios para obtener información confidencial, podemos desarrollar contramedidas más efectivas y fortalecer la seguridad de los sistemas.
* Evaluación de la resistencia de sistemas y dispositivos: Estudiar los *Side Channel Attacks* proporciona una herramienta valiosa para evaluar la resistencia de sistemas y dispositivos a posibles amenazas.
* Protección de la privacidad: Estudiar los *Side Channel Attacks* es esencial para proteger la privacidad de los usuarios.
* Estudio y evaluación con diferentes programas: Esto nos permite el estudio de dichos ataques empleando varios programas diferentes con los que podemos analizar todos los comportamientos a la hora de ejecutar dichos ataques.
* Desarrollar softwares de contramedida: La posibilidad de desarrollar nuevos softwares para poder evitar todo este tipo de ataques y poder implementarlos en los micros.

En general, el principal objetivo es entender todo el funcionamiento de este tipo de ataques laterales y poder implementar las contramedidas necesarias para poder evitar o atenuar ataques.

# Materiales y métodos

## ChipWhiperer Lite 32 Bits con Target Device Incorporado

El ChipWhisperer-Lite [6] representa la búsqueda más agresiva de NewAE Technology Inc. para proporcionar, de la forma más fácil y eficaz, el estudio de los ataques laterales, tanto los ataques de potencia como los ataques de *Glitching* de reloj a cualquier persona con conocimientos en el sector, de estudiantes a ingenieros de software. Es completamente de código abierto (hardware, software, firmware, código FPGA) y está proporcionando una revolución en la seguridad de hardware.

El ChipWhisperer-Lite integra hardware para realizar mediciones de análisis de potencia, programación de dispositivos, *Glitching*, comunicaciones serie y un ejemplo de objetivo que puede cargarse con algoritmos criptográficos, todo en una sola placa. La versión de placa única viene en dos variantes: Atmel XMEGA o STM32F3 como objetivo víctima.

Se puede separar la parte de captura y el *target* para conectarlos a otros objetivos. La parte de captura está disponible como un componente independiente y viene lista con conectores SMA y para el objetivo, pero requiere un objetivo externo.

**Especificaciones:**

**Tabla

Descripción generada automáticamente**

Ilustración 1: Especificaciones del Chipwhisperer [6]

Estas son las especificaciones que adjunta nuestra placa y podemos observar que en este caso los dos tipos de ataques disponibles son el de reloj y el de la potencia. Por otra parte, se puede observar cual es nuestro *Target Device* o víctima, varía según el tipo de placa, pero en una de ellas nos ofrece el STM32Fx, un microcontrolador ARM Cortex-M.

Por otra parte, se pueden observar los tiempos de medida que ofrece la placa con un *Offset* de 200 ps como mínimo. Tiene rangos de frecuencia de 5 a 200 MHz y una salida de reloj con salida de *Glitch* incorporada.

**Arquitectura síncrona:**

**Imagen de la pantalla de una computadora

Descripción generada automáticamente con confianza media**

Ilustración 2: Ejemplo de arquitectura síncrona vista en osciloscopio

ChipWhisperer puede utilizarse para inyectarse en la señal de reloj de un dispositivo objetivo y aplicar multiplicaciones y desplazamientos de fase para muestrear en el punto deseado(s) durante el ciclo de reloj. Esto asegura que los puntos de muestreo estén directamente relacionados con el reloj digital que genera las señales de interés. El resultado es que muchos dispositivos pueden ser atacados con una frecuencia de muestreo de 5 a 100 veces más lenta en comparación con un osciloscopio regular.

**Tipos de placa:**

Un circuito electrónico

Descripción generada automáticamente con confianza media

Ilustración 3: Imagen de ChipWhisperer. A la izquierda la placa de generación y análisis de ataques, a la derecha la víctima

La compañía ofrece varias posibilidades a la hora de realizar la compra. En la Ilustración 3 se puede observar la parte de captura y el objetivo en una misma placa, pero cabe la posibilidad de realizar la compra por partes separadas si el objetivo que queremos atacar es diferente o si queremos incorporar osciloscopios externos u otros tipos de medidores.

En las siguientes imágenes podemos observar los diferentes tipos de pedidos que se pueden realizar:

Tipo 1:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Ilustración 4: ChipWhisperer single board

Este modelo es con el cual vamos a trabajar. Incorpora en la misma placa el objetivo y la placa de medida, todo interconectado entre sí. Es más útil si se tiene claro el tipo de ataques que se quieren realizar y el objetivo deseado, aunque te limita el acceso a otras funciones como el poder conectar objetivos externos diferentes al que ya tenemos incorporado en la placa. También nos limita el no poder conectar osciloscopios para visualizar el comportamiento sin tener que soldar o hacerlo manualmente desde la placa.

Tipo 2:

Texto

Descripción generada automáticamente

Ilustración 5: ChipWhisperer placa de ataque con conectores soldados

Este modelo solo incorpora la parte de captura por lo que se necesitaría comprar aparte un objetivo para atacar. Por eso incorpora cables y conectores para poder realizar las medidas y las conexiones necesarias.

Tipo 3:

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Ilustración 6: ChipWhisperer, placa de ataque y víctima separadas y con conectores

Este modelo incorpora las dos partes anteriores, pero por separado. Es decir, tanto la placa de captura como el objetivo, pero en diferentes placas además de aportar cables de conexionado para poder interconectarlo todo.

En resumen, dependiendo del tipo de pruebas que se quieran realizar y el nivel de conocimiento que se tenga, se elige un tipo diferente de producto.

**Software:**

El ChipWhisperer ofrece un software incorporado que te permite realizar todo tipo de medidas y programación necesaria. ChipWhisperer es una cadena de herramientas de código abierto para la investigación en seguridad embebida. Todos los targets y los hardware de captura en este catálogo son compatibles con una aplicación de captura basada en Python. La naturaleza de código abierto significa que se pueden realizar modificaciones según necesidades específicas, ya sea desarrollando tus propios algoritmos o realizando validaciones en objetivos propietarios.

Actualmente ya no existe la versión visual de Windows, en la cual se tenía acceso a varios programas gráficos de captura y medida. Ahora todo es realizado basándonos en la programación en Python lo cual nos permite tener aún más libertad y expansión a la hora de querer adaptarlo a nuestros requerimientos. Aunque, por otra parte, requiere más conocimientos y dedicación.

## Máquina Virtual de ChipWhisperer

Hay varias formas de realizar la instalación de todo lo necesario para poder ejecutar todas las funciones del Micro, pero existe la posibilidad de descargar la máquina virtual en la cual está incorporado todas las funciones y programas de captura y ataque para el ChipWhisperer.

En este caso, accederemos a la máquina virtual para poder acceder a los archivos, pero trabajaremos con Jupyter con acceso mediante un *host* desde nuestro navegador. Tiene directamente implementado un intérprete Python en el cual podemos ejecutar nuestros códigos para poder estudiar el comportamiento del micro.

En este TFG trabajaremos con VirtualBox y la extensión o la imagen de la computadora preconfigurada del ChipWhisperer con Jupyter, intérprete Python y todas las librerías necesarias.

Aquí explicaremos mediante una serie de imágenes cuál es la forma de acceder a la máquina virtual y cómo poder tener acceso a Jupyter.

Primero hay que introducir el usuario y la contraseña, que por defecto ambos son “vagrant”, ver Ilustración 7. Una vez accedemos nos indica que introduzcamos una contraseña nueva para el servidor de Jupyter.

Una vez creada la contraseña ya tendríamos acceso a Jupyter. Ahora hay que ver en la barra de la parte inferior de nuestra máquina virtual la IP de red a la que estamos conectados y copiarla (Ilustración 8). Introducir esta IP en nuestro navegador seguido por el puerto de Jupyter (8888). Al final la URL quedaría algo parecido a XXX.XXX.XXX.XXX:8888 como en la Ilustración 9.

Accedemos a Jupyter e introducimos la contraseña que hemos definido en la máquina virtual. En este punto ya estaríamos dentro de Jupyter y tendríamos acceso a todos los ejemplos y archivos de chipwhisperer, tal y como se ve en la Ilustración 10.

* Primer paso:

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 7: Máquina virtual de Chipwhisperer

* Segundo paso:



Ilustración 8: Icono de red de la máquina virtual

* Tercer paso:



Ilustración 9: IP y el host en nuestra barra de direcciones del navegador web

* Cuarto paso:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ilustración 10: Panel principal de Jupyter desde el navegador web

Una vez realizados todos estos pasos ya tendríamos acceso a los archivos de Jupyter y ya podríamos realizar las pruebas de iniciación para conectar nuestro microprocesador y comprobar que hay comunicación tanto a la placa de ataques como a la víctima.

## Chipwhisperer Lite 32 BITS

El "ChipWhisperer Lite" es una versión específica del proyecto ChipWhisperer, que es un conjunto de herramientas de código abierto desarrolladas para realizar análisis de seguridad en dispositivos embebidos y sistemas integrados. La herramienta se centra en la detección y evaluación de ataques de canales laterales, como los ataques de análisis de potencia, que buscan explotar las fugas de información de consumo de energía para obtener datos confidenciales. Este es solo un ejemplo de los muchos otros tipos de ataques que se pueden realizar con esta herramienta y los cuales explicaremos con más detalles posteriormente.

ChipWhisperer Lite está diseñado para proporcionar una plataforma de bajo costo y fácil acceso para aquellos interesados en aprender y experimentar con ataques de canales laterales en sistemas embebidos.

Ofrece capacidades para realizar ataques de análisis de potencia y de reloj, así como otras técnicas de evaluación de seguridad. Incluye hardware y software que permiten a los usuarios interactuar con dispositivos integrados para evaluar su resistencia a ataques de canales laterales. Es un proyecto de código abierto, lo que significa que su hardware y software son accesibles al público [7].

## Tipos de targets

Existen varios *targets* con los cuales podemos adaptar nuestro ChipWhisperer para realizar diferentes tipos de ataques [8]. Normalmente según el tipo de ataques existen unos *targets* u otros, por ejemplo, en nuestro caso usaremos el ARM CORTEX-M.

Existen varios tipos de targets, citaremos los más usados o los más comunes:

* Targets Integrados: este tipo de targets normalmente vienen incorporados ya en la placa de captura, es decir, viene todo incorporado en la misma placa. Nosotros emplearemos el CW303 – ARM.

En la siguiente imagen os adjuntaremos los 3 targets integrados que existen.

Pantalla de un celular con letras

Descripción generada automáticamente

Ilustración 12: Diferentes Targets Integrados

* Targets Externos: este tipo de targets viene incorporado en una placa totalmente diferente a la de captura, por lo que tenemos dos placas diferentes las cuales pueden interactuar entre sí conectándolas mediante cableado y otro tipo de conexiones como (20-pin o cables SMA).

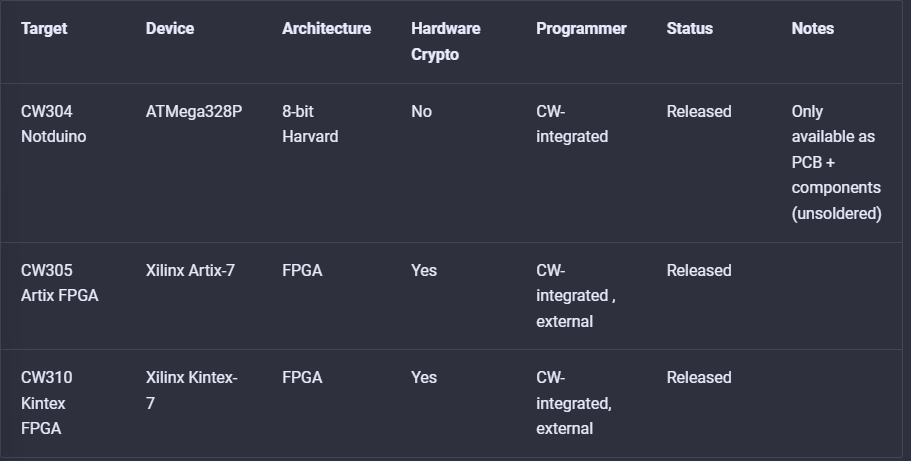


Ilustración 13: Diferentes Targets Externos

* Targets CW308 UFO: este tipo de targets suele ser empleado para atacar otros tipos de targets. Ofrecen muchas comodidades para alimentar otras placas target incluyendo reguladores de voltaje, circuitos de accionamiento cristal, selección de reloj, etc.

Estos algunos de ellos, pero en [8] se pueden encontrar todos los diferentes targets.

32 bits

Pantalla de un celular de un mensaje en letras blancas

Descripción generada automáticamente con confianza media

Ilustración 14: CW308 Targets 32 bits

8/16 bits



Ilustración 15: CW308 Targets 8/16bits

En conclusión, en función del tipo de uso que le queremos dar o qué tipo de ataques queremos realizar, utilizaremos una placa u otra o varias a la vez. Cabe destacar que normalmente, al ser integradas, se entiende que es para realizar el tipo de ataque que se ha elegido, pero existe la posibilidad de poder separarla de la placa de captura cortando las conexiones y luego soldar otro tipo de targets si queremos ampliar nuestra búsqueda.

# 

# ESTUDIO TEÓRICO

## Tipos de side channel attacks

Existen varios tipos de ataques laterales en los cuales cada uno de ello va dirigido a aprovechar diferentes aspectos y características e los sistemas. Los tipos de ataques más comunes son los dos en que se centra este TFG, que es estudiarán más en detalle, para el resto sólo esbozaremos una introducción.

### Análisis de potencia (*Power Analysis*)

Este tipo de ataque se basa en medir y analizar el consumo de energía de un dispositivo durante su operación. Puede revelar información sobre las operaciones internas, como cálculos criptográficos, a través de patrones en el consumo de energía. Básicamente, son ataques que utilizan la variación en el consumo de energía del hardware durante los cálculos. Existen todos tipos de análisis o de ataques cuando hablamos de la interpenetración del consumo de energía del hardware:

* El Análisis de Potencia Simple (SPA): consiste en la interpretación visual de trazas de potencia, o gráficos de la actividad eléctrica a lo largo del tiempo.
* El Análisis de Potencia Diferencial (DPA): es una forma más avanzada de análisis de potencia, que permite a un atacante calcular los valores intermedios dentro de las operaciones criptográficas a través del análisis estadístico de datos recopilados de múltiples operaciones criptográficas.

Diagrama

Descripción generada automáticamente

Ilustración 16: Diagrama de análisis de Potencia Diferencial

El ataque DPA consta de tres fases principales: en primer lugar, la fase de recopilación de datos, durante la cual se recopila un conjunto de trazas. Luego, tenemos una fase de entrenamiento, en la cual el atacante calibra la plantilla (ataque clásico) o entrena el modelo de aprendizaje automático.

Finalmente, está la fase de ataque, donde la plantilla/modelo se utiliza para recuperar un conjunto de claves secretas y evaluar la resistencia de la implementación a un Ataque de Canal Lateral (SCA). Durante esta fase, los atacantes utilizan múltiples trazas para cada clave atacada y combinan sus probabilidades para hacer la mejor suposición posible.

Un aspecto técnico importante y, a menudo pasado por alto, de este tipo de ataque de canal lateral es que no recuperará la clave de una sola vez; en cambio, el ataque se centra en recuperar un solo byte de clave. Por esta razón, para AES128, se necesitan 16 ataques (y 16 modelos) para recuperar la clave completa.

Captura de pantalla de un celular con letras

Descripción generada automáticamente

Ilustración 17: Fases para un Ataque de Potencia

Para explicar con detalle cómo funciona el *Power Analysis Attack* explicaremos las 3 fases con más detalle.

FASE 1. Recopilar trazas de potencia:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 18: recopilación de Datos de Potencia

1. El programa responsable de recopilar todas las gráficas o trazas de potencia, en este caso viene dado por el Chipwhisperer, nos ofrece directamente la posibilidad de tener un osciloscopio integrado dentro de las herramientas que ofrece.
2. El código provoca la encriptación de una clave y un texto plano elegidos al azar en el hardware objetivo.
3. Al finalizar la encriptación, el programa detiene la captura y recopila la traza de potencia del osciloscopio. La traza y su etiqueta (la clave y el texto plano utilizados) se añaden al conjunto de datos del ataque.

Fase 2. Configuración del Hardware y Evaluación de resultados.

En la práctica, para facilitar las cosas, el código encargado de recopilar las trazas no controla directamente el osciloscopio ni interactúa con el chip objetivo. En su lugar, utiliza hardware (ChipWhisperer) que estandariza esas interacciones y las hace fáciles de programar. Para tener un poco de idea genérica de cómo se vería una traza de potencia podemos observar la Ilustración 19.

Imagen que contiene Gráfico

Descripción generada automáticamente

Ilustración 19: Traza de potencia

Al final de cada captura se obtiene un conjunto de trazas de potencia que se asemeja al que se muestra en la captura de pantalla anterior. Esta imagen muestra la traza de potencia para una implementación de AES no protegida. Es evidente que esta implementación tiene muy pocas contramedidas contra ataques de canal lateral (SCA), ya que se pueden ver claramente los 10 picos de AES. Las implementaciones mejor protegidas aún muestran este patrón de 10 picos, pero no necesariamente de manera tan clara.

Una vez recopilada toda esta información, pasamos a la fase de entrenamiento y evaluación de información capturada.

El objetivo de recopilar trazas es construir dos conjuntos de datos de ejemplos: uno de entrenamiento y otro de evaluación. El conjunto de entrenamiento es utilizado por algoritmos de ataque, ya sean basados en aprendizaje profundo o no, para aprender cómo cambia el consumo de energía en relación con la clave utilizada durante la encriptación. El conjunto de evaluación, como se discutirá más adelante, se utiliza una vez que el algoritmo está entrenado/calibrado para evaluar la eficiencia del ataque.

Para el conjunto de entrenamiento se puede elegir libremente qué claves y textos planos se utilizarán durante el cálculo, ya que se asume que este paso se ejecuta en dispositivos que están bajo el control del atacante. Aquí es útil tener un algoritmo de muestreo eficiente para asegurar que el espacio se muestree correctamente. La cantidad de trazas a recopilar varía según la protección de la implementación

El conjunto de evaluación es diferente. Su objetivo es representar un objetivo real. Como resultado, las claves utilizadas en este conjunto deben generarse completamente al azar, ya que el atacante no tiene control sobre cómo los usuarios eligen sus claves. Además, este conjunto debe generarse en un chip diferente para asegurar que el ataque se generalice entre chips, ya que puede haber ligeras variaciones de comportamiento debido a la tolerancia de fabricación del chip.

Fase 3. Ataque:

La siguiente pregunta que necesitamos responder es: ¿qué valores debería predecir el ataque? La respuesta obvia sería el valor de los bytes de la clave. Sin embargo, en la práctica, para la mayoría de las implementaciones, esto no funcionará a menos que, por error, se haya incluido la carga de la clave en la memoria en la captura. Se llama a esto un error porque, en la mayoría de los escenarios del mundo real la clave está integrada en el dispositivo y solo se carga una vez, no en cada encriptación. Por lo tanto, esperar observarla con cada encriptación es un modelo de ataque poco razonable y conduce a resultados falsos positivos.

En cambio, el ataque intenta predecir los valores de lo que se llaman puntos de ataque, también conocidos como variables (de fuga) sensibles en la comunidad de criptoanálisis. En pocas palabras, un punto de ataque es un punto en el algoritmo donde la computación causó algún cambio en la memoria (cambiar un valor de registro, establecer un valor, etc.) que tiene alguna relación con el valor que intentamos recuperar (por ejemplo, XOR). Cambiar los valores de la memoria desencadena un cambio en el consumo de energía, lo que significa que este cambio será observable en la traza de potencia.

Diagrama

Descripción generada automáticamente

Ilustración 20: Ejemplo Gráfico del Comportamiento de AES Attack Points

Como se puede ver en el diagrama anterior, que representa las operaciones de AES a lo largo del tiempo, AES tiene muchos puntos de ataque a lo largo de sus 10 rondas (los puntos de ataque están representados por los puntos rojos/amarillos). Sin embargo, en la práctica, la mayoría de ellos no son invertibles; no podemos inferir el valor del byte clave a partir del valor supuesto. Como resultado, nos enfocamos solo en aquellos marcados con puntos rojos, que son directamente invertibles. Los tres principales, todos ubicados en la primera ronda y representados en el lado derecho del diagrama, son:

* Clave (key): Intentar inferir la clave ya que el valor del byte se almacena en la memoria.
* Sub\_bytes\_in: El valor del byte objetivo después de que la clave se almacena con el texto plano.
* Sub\_bytes\_out: El valor del byte después de ser sustituido por otro valor utilizando la caja AES.

En este punto, para recuperar las claves después del análisis, recopilación de datos y evaluación de dichos resultados, recurriremos a ChipWhisperer y a nuestro modelo ya previamente diseñado para la extracción de la clave que necesitamos.

Diagrama

Descripción generada automáticamente

Ilustración 21: Modelo de suma para predecir que byte es correcto

Como se mencionó anteriormente, una de las principales ventajas de utilizar el aprendizaje profundo para SCA es que facilita la realización de un ataque probabilístico que escala según el número de trazas utilizadas. Simplemente necesitas acumular las predicciones de los modelos, como se muestra en el diagrama anterior, para decidir cuál es el valor de byte más probable. Esta suma directa significa que, cuantas más trazas combines durante el ataque, más evidencia se acumulará, y mejor será tu tasa de éxito, siempre y cuando tu modelo haya convergido (es decir, predijo de manera más precisa que al azar).

Esta formulación del ataque también tiene la ventaja de producir naturalmente una lista completamente ordenada de suposiciones potenciales, lo que facilita la adición de un componente de fuerza bruta para probar las combinaciones más probables en los 16 bytes clave. Por último, pero no menos importante, nos permite saber cuán lejos está la suposición correcta de la parte superior, lo que nos permite medir mejoras en el ataque y comparar modelos de ataque de manera más detallada.

**Contramedidas Contra los *Power Analysis Attacks*:**

Fw: Generar ruido activando otros bloques aleatoriamente; ¿jugar con timing para adelantar o retrasar picos de consumo?; ¿precifrado, por ejemplo una xor antes del AES?

Hw: atenuar picos

### Análisis de tiempo (*Clock Glitching* o *Timing Analysis*)

**Contramedidas Contra los *Clock Glitching attacks Attacks*:**

Fw: varias trap instructions en el código para detectar si son saltadas; usar timer con oscilador interno y ver drift.

Hw: uso de oscilador interno; medida del ancho de pulsos (esto es una idea que tengo para investigar en el futuro)

### Otros

* Análisis de Emisión Electromagnética (EMA):
* Análisis de sonido (*Acoustic Analysis*):
* Análisis de Radiación Térmica (*Thermal Analysis*):
* Ataques de Canal Lateral en Criptografía (*Side Channel Attacks in Cryptography*):

# Trabajo práctico

## Ataques simples de Power analysis

## Ataques simples de Clock glitching

## Implementación y prueba de contramedidas para ataques de power analysis

## Implementación y prueba de contramedidas para ataques de clock glitching

# Conclusiones

# Bibliografía

1. Wikipedia contributors. Side-channel attack. Wikipedia, The Free Encyclopedia. January 10, 2024, 11:14 UTC. Available at: <https://en.wikipedia.org/w/index.php?title=Side-channel_attack&oldid=1194717603>. Accessed January 27, 2024.
2. Colaboradores de Wikipedia, "Ataque de canal lateral," Wikipedia, La enciclopedia libre, <https://es.wikipedia.org/w/index.php?title=Ataque_de_canal_lateral&oldid=155734766> (descargado 1 de diciembre de 2023).
3. Jake Hert. Understanding Side Channel Attacks Basics. All about circuits, <https://www.allaboutcircuits.com/technical-articles/understanding-side-channel-attack-basics/>. Accedido el 27 de enero de 2024.
4. Elie Bursztein, Hacker’s guide to deep-learning side channel attacks: the theory. Elie publications, <https://elie.net/blog/security/hacker-guide-to-deep-learning-side-channel-attacks-the-theory/>. Accedido el 27 de enero de 2024.
5. Wikipedia contributors. Power analysis. Wikipedia, The Free Encyclopedia. August 8, 2022, 01:31 UTC. Available at: <https://en.wikipedia.org/w/index.php?title=Power_analysis&oldid=1103018064>. Accessed January 27, 2024.
6. <https://media.newae.com/datasheets/NAE-CW1173_datasheet.pdf>
7. <https://github.com/newaetech/chipwhisperer>
8. <https://rtfm.newae.com/Targets/>

# Anexos

## Código Fuente X

## Código Fuente Y