Instituto de Microelectrónica de Sevilla
(Universidad de Sevilla y CNM-CSIC)

Vulnerabilidad y análisis diferencial mediante inserción de fallos de cifradores Trivium en FPGA y ASIC

Memoria presentada por
Francisco Eugenio Potestad Ordóñez
para optar al título de Doctor por la Universidad de Sevilla

Directores de Tesis
Dr. Manuel Valencia Barrero
Dr. Carlos Jesús Jiménez Fernández

Lugar y fecha
Sevilla, 2019

Instituto de Microelectrónica de Sevilla
(Universidad de Sevilla y CNM-CSIC)

Vulnerabilidad y análisis diferencial mediante inserción de fallos de cifradores Trivium en FPGA y ASIC

Memoria presentada por:

Francisco Eugenio Potestad Ordóñez
para optar al título de Doctor por la Universidad de Sevilla

Directores:

Manuel Valencia Barrero

Carlos Jesús Jiménez Fernández

Sevilla, 2019

A mis padres,

Agradecimientos

Me gustaría agradecer a mis tutores Carlos Jesús Jiménez y Manuel Valencia toda su ayuda y tiempo dedicado a lo largo de la realización de esta Tesis, así como la oportunidad ofrecida y confianza depositada en mí para llevar a cabo este trabajo.

A Antonio Acosta por darme la oportunidad de poder trabajar en su grupo de investigación.

A José Miguel Mora por su ayuda y numerosos consejos a lo largo de estos años.

A Ricardo Chaves por darme la oportunidad de realizar mi estancia en Lisboa y poder colaborar con su grupo de investigación.

A mis compañeros y amigos del IMSE.

A mi familia.

A los siguientes proyectos y entidades por su soporte y financiación:

- CITIES: Circuitos Integrados para Transmisión de Información Especialmente Segura (TEC2010-16870), Ministerio de Ciencias e Innovación.
- CESAR: Circuitos Microelectrónicos Seguros Frente a Ataques Laterales (TEC2013-45523-R), Ministerio de Economía y competitividad.
- INTERVALO: Integración y validación en laboratorio de contramedidas frente a ataques laterales en criptocircuitos microelectrónicos (TEC2016-80549-R), Ministerio de Economía y Competitividad.
- MISAL: Microelectrónica segura frente a ataques laterales (201450E034), Consejo Superior de Investigaciones Científicas (CSIC).
- LACRE: LAboratorio de Criptoanálisis HardwarE (201550E039), Consejo Superior de Investigaciones Científicas (CSIC).
- CRYPTACUS: ICT COST ACTION IC1403: Cryptanalysis in ubiquitous computing systems.

Resumen

Las comunicaciones entre dispositivos aumenta día a día y un gran ejemplo de ello es el crecimiento del Internet de las cosas, en inglés *Internet of things* (IoT). De entre todas las comunicaciones que se producen, parte de ella está compuesta por información sensible susceptible de ser interceptada por terceras partes con fines malintencionados. Con el fin de evitar este gran problema, la comunidad científica se ha centrado en la constante búsqueda y desarrollo de algoritmos criptográficos o criptosistemas, algoritmos orientados tanto a software como a hardware, que permitan asegurar unas comunicaciones donde los canales de transmisión son potencialmente inseguros.

A la hora de poder establecer nuevos estándares de seguridad, es necesario estudiar la seguridad ofrecida por los nuevos algoritmos desde el punto de vista de su vulnerabilidad con el objetivo de reducirla. Estas vulnerabilidades de los llamados criptosistemas es posible estudiarlas tomando el rol de una tercera parte que trata de obtener la información secreta del dispositivo y con ello conocer dónde se encuentran sus puntos débiles. Es aquí donde se enmarca la presente Tesis Doctoral. A lo largo de este texto, se realiza un estudio del estado del arte de la criptografía, así como las técnicas más importantes para comprometer la seguridad de los criptosistemas actuales, siendo objeto de estudio el cifrador de flujo Trivium, tanto el diseño original presentado en el portfolio del proyecto eSTREAM, como diferentes variantes de éste.

Para poder estudiar la vulnerabilidad de estos criptosistemas y poder recuperar su información secreta, se han diseñado diferentes sistemas de inserción de fallos tanto en tecnología FPGA como en ASIC. Estos sistemas de ataque se han implementado para poder atacar al cifrador mediante la manipulación de su señal de reloj y sus señales de control. Gracias a estos sistemas de ataque experimentales, es posible determinar los puntos débiles de estos criptosistemas y mediante el uso de análisis diferenciales recuperar su información secreta, clave y vector de inicialización. Este estudio, por tanto, presenta la primera rotura de este cifrador de forma experimental, consiguiendo en el 100 % de los casos la recuperación de su clave secreta y probando que este criptosistema es vulnerable a los ataques por inserción de fallos.

Índice general

1. Introducción

1.1. Presentación

Hoy en día, debido al desarrollo constante de los dispositivos electrónicos y de los sistemas de comunicación, en nuestra sociedad es posible establecer comunicaciones en cualquier lugar y en cualquier momento entre dispositivos al alcance de la mano de cualquier usuario. En muchas de las comunicaciones establecidas en cualquiera de los múltiples entornos, por ejemplo, mediante el uso de un teléfono móvil o mediante una tarjeta bancaria, los mensajes que se transmiten contienen información sensible de los propios usuarios. La necesidad de dotar de la debida seguridad a los miles de millones de comunicaciones que hay en el mundo supone un auténtico reto social, científico y tecnológico. Se trata de un problema de carácter trasversal que afecta a múltiples sectores sociales y que posee una enorme diversidad de la que, a mero título de ejemplo, podemos indicar que abarca desde el acceso a documentos del más alto secreto en un entorno que cuenta con sofisticados recursos, hasta las transacciones más habituales en nuestro día a día.

Ante este reto, la comunidad científica, al igual que otros sectores, ha volcado su atención en el desarrollo de sistemas que permitan establecer comunicaciones seguras incluso en medios de comunicación potencialmente inseguros en los que pueden aparecer terceras partes que quieran acceder a la información sensible a veces con intenciones maliciosas e indeseadas. Para dotar de la necesaria seguridad se han desarrollado múltiples algoritmos que permiten cifrar (encriptar) la información, así como se han implementado sistemas que operan con esos algoritmos, a los que nos referiremos como sistemas criptográficos o criptosistemas. Los criptosistemas se encuentran en constante desarrollo, siendo objeto de un exhaustivo estudio para establecer nuevos estándares y poder así mejorar su seguridad.

Paralelamente al desarrollo de los criptosistemas se produce un desarrollo de los ataques contra ellos con objetivos que pueden ser malintencionados como son, entre otros, conocer la información que se transmite o cambiarla sin que lo conozca el remitente de la misma. Estos ataques se realizan tanto contra el algoritmo de cifrado como contra el sistema criptográfico. Uno de los objetivos más buscados por los atacantes es averiguar las claves que sustentan el cifrado y que se suponen son un secreto solo compartido por los usuarios. Por su parte, la comunidad científica busca desarrollar los algoritmos y criptosistemas para minimizar sus vulnerabilidades.

En la actualidad y debido a la apreciable longitud de las claves utilizadas en los algoritmos, los ataques matemáticos a estos algoritmos ya no son prácticos por el gran consumo de recursos y de tiempo que se necesitarían para comprometer su seguridad. Sin embargo, los ataques a las implementaciones de dichos algoritmos están demostrando ser mecanismos bastante efectivos. Por tanto, para que los sistemas criptográficos sean robustos, no sólo deben ser resistentes los algoritmos, sino también dispositivos físicos que los implementan.

En esta Tesis estamos interesados en las implementaciones hardware de los algoritmos y, en particular, en las implementaciones electrónicas con tecnologías de alta densidad de integración, tanto ASIC (Applied System Integrated Circuit) como FPGA (Field-Programmable Gate Array). Los ataques a las implementaciones físicas de los criptosistemas obtienen información del circuito durante su funcionamiento. Esta información se puede obtener de forma pasiva (midiendo el consumo de potencia, radiación electromagnética, etc), de forma activa (provocando errores en el funcionamiento del circuito) o de forma invasiva (accediendo a puntos internos del circuito). Dado que con la información obtenida podría ponerse en peligro la seguridad del dispositivo, con el fin de prevenir los ataques surge la necesidad de estudiar las vulnerabilidades de los dispositivos criptográficos implementados en diferentes tecnologías. Para ello es necesario tomar el rol de un posible atacante y crear sistemas que permitan atacar las implementaciones para conocer sus vulnerabilidades y, así, poder corregirlas en diseños futuros.

En cuanto al sistema de cifrado, este trabajo de Tesis se ha centrado en cifradores de flujo, concretamente, en el denominado Trivium. De esta forma, la Tesis tiene como principal objetivo atacar implementaciones FPGA y ASIC de cifradores de flujo Trivium con el fin último de llegar a conocer las claves secretas de manera experimental.

Debe indicarse, por una parte, que en la literatura especializada no existen reportados trabajos que cubran el proceso completo de ataque activo y recuperación de clave de forma experimental para el cifrador Trivium, ni para otros cifradores exceptuando el cifrador de bloques AES. Existen diferentes trabajos que proponen mecanismos para llevar a cabo el proceso, pero en ellos se obtienen los resultados vía simulación o de forma teórica, sin llegar a romper el criptosistema de forma real. Por otra parte, para alcanzar ese objetivo principal es necesario conseguir otros hitos entre los que podemos señalar los de disponer de implementaciones de cifradores de flujo Trivium en FPGA y en ASIC, diseñar e implementar sistemas de ataques sobre ambas tecnologías, estudiar su vulnerabilidad frente a ataques, realizar un análisis diferencial a partir de datos experimentales y recuperar la clave secreta.

Ciñéndonos al cifrador Trivium, estudios teóricos sobre el algoritmo de cifrado presentados en la literatura constatan vulnerabilidades del algoritmo si se es capaz de introducir un solo error en el registro de estados del cifrador. En nuestro trabajo de Tesis pretendemos demostrar la viabilidad de forma experimental de dichos estudios, llevando a la práctica las suposiciones teóricas. Para ello nos planteamos desarrollar sistemas de ataques del tipo activo no invasivo con los que se procurará la inyección de fallos en el funcionamiento del cifrador Trivium mediante la manipulación de señales temporales, bien de la propia señal de reloj, bien de la temporización de las señales de control. Tras demostrar la vulnerabilidad del cifrador estaremos en condiciones de aplicar a los datos experimentales obtenidos un análisis diferencial que permitirá la obtención de la clave secreta y, de aquí, la ruptura de la seguridad del cifrador. Es nuestro propósito llevar a cabo estos estudios tanto al cifrador Trivium estándar, como también a versiones multi-bit y de baja potencia de dicho cifrador, con la finalidad de probar la eficacia de los sistemas diseñados y probar que es posible comprometer la seguridad de los cifradores Trivium en todas sus versiones. Este trabajo de Tesis, por tanto, se encuentra dentro del muy pequeño grupo de trabajos experimentales que intentan probar en la práctica las suposiciones teóricas de recuperación de clave de un criptosistema mediante el uso de ataques por inserción de fallos de forma experimental.

1.2. Objetivos

El principal objetivo de este trabajo de investigación es el estudio del comportamiento de implementaciones hardware de los cifradores de flujo Trivium, en sus diferentes versiones, frente a ataques activos no invasivos por inserción de fallos, el análisis de su vulnerabilidad y la ruptura del cifrador mediante la realización de un análisis diferencial de fallos. A continuación se desglosa en objetivos más concretos la consecución de este objetivo principal:

1. Actualizar el estado de conocimientos del campo del criptoanálisis y, en particular, lo relacionado con el cifrador de flujo Trivium. Se trata de analizar los diferentes sistemas de cifrado y las técnicas reportadas en la literatura para atacar las implementaciones físicas de los algoritmos criptográficos. A su vez, estudiar las técnicas de análisis que permiten aprovechar los ataques experimentales para poder recuperar la información secreta a partir de los fallos generados. Esto debe permitir clasificar las referencias bibliográficas sobre recuperación de información secreta, enfatizando en la diferencia entre los trabajos teóricos y los experimentales. Estos estudios deben finalizar con un análisis bibliográfico específico de los trabajos reportados sobre el cifrador Trivium.

2. Trabajar con cifradores de flujo Trivium reales. De las distintas opciones de cifrado, se trabajará con cifradores de flujo, en concreto, con el cifrador de flujo Trivium, del que se explorarán diferentes versiones. Se analizará en profundidad la estructura del cifrador Trivium y su funcionamiento, se afrontarán sus diseños bajo diferentes consideraciones (p. ej., la forma serie o paralela de introducir la clave y el vector de inicialización, si el Trivium es mono-bit o multi-bit, etc.) y se implementarán para tecnologías FPGA y ASIC.

3. Determinar las formas de introducir experimentalmente fallos en cifradores Trivium. Dado que los estudios reportados por la comunidad internacional sobre la vulnerabilidad de estos cifradores frente a la inserción de fallos, como presentaremos en el próximo capítulo, son sólo teóricos, un objetivo parcial es analizar las estrategias para introducir fallos de forma experimental. Entre éstas, la Tesis se centrará en los ataques activos no invasivos con el fin de llevarlos a cabo tanto sobre implementaciones FPGA del cifrador Trivium como sobre implementaciones ASIC.

4. Implementar sistemas de ataques. Una vez establecidas las posibles formas de introducir errores en la operación del cifrador, se diseñarán e implementarán sistemas que permitan llevar a cabo estos ataques de forma experimental. Dichos sistemas, además de inducir el fallo, posibilitarán analizar el comportamiento de los cifradores de flujo frente a ataques.
5. Analizar la vulnerabilidad de las implementaciones del cifrador Trivium. Se procurará establecer los puntos débiles de este tipo de cifradores y la posibilidad de explotarlos a la hora de generar trazas de datos útiles para los análisis diferenciales que permitan recuperar su información secreta. La vulnerabilidad se explorará mediante experimentos de inserción de fallos tanto en FPGA como ASIC.
6. Implementar un sistema experimental de recuperación de clave. Una vez estudiadas las vulnerabilidades de los cifradores, se llevará a cabo el montaje de un sistema completo que permita el ataque experimental al cifrador y el análisis diferencial de las trazas generadas a partir de los fallos introducidos experimentalmente con el fin último de recuperar la clave secreta del cifrador.
7. Obtener conclusiones. Se tratará tanto de exponer los principales resultados obtenidos del trabajo como de extraer de ellos las principales conclusiones. Por su especial relevancia, adelantamos que en este trabajo de Tesis se muestra por primera vez que es posible romper el sistema de cifrado Trivium de forma experimental.

1.3. Contenido

La memoria de esta Tesis se ha dividido en seis capítulos, tal como se muestra en el siguiente resumen. Se debe señalar que su organización obedece a los pasos seguidos en la metodología de trabajo.

- Capítulo 1: Introducción.

  Es el presente capítulo donde, además de motivar la realización de la presente Tesis y puntualizar sus objetivos, presentamos la estructura de esta Memoria.

- Capítulo 2: Criptografía, cifradores y criptoanálisis.

  Se realiza un análisis en profundidad del estado del problema que se pretende abordar. Se estudia el marco teórico, así como el contexto en el que se encuadra la problemática a abordar en la Tesis. Además, en este capítulo se presenta el sistema criptográfico utilizado, haciendo un análisis de los estudios teóricos y un repaso a los diferentes tipos de ataques posibles a la implementación de los sistemas criptográficos. Tras esto se presenta la técnica de análisis, el tipo de ataque y las propuestas de ataque elegidas para llevar a cabo el desarrollo de esta Tesis.

- Capítulo 3: Inserción de fallos en implementaciones FPGA de cifradores Trivium.

  En este capítulo se presenta la implementación del sistema de ataque en FPGA. El sistema de ataque utilizado ha consistido en la manipulación de la señal de reloj del cifrador. Para ello se han utilizado las familias de FPGA Spartan 3E y Spartan 6, las placas de desarrollo Nexys 3 y los diseños del cifrador Trivium estándar, estándar con carga en serie, estándar con carga en paralelo y de baja potencia. Finalmente se presentan las conclusiones obtenidas a partir de las pruebas realizadas tras aplicar el sistema de ataque diseñado a las diferentes versiones del cifrador.

- Capítulo 4: Inserción de fallos en implementaciones ASIC de cifradores Trivium.

  En este capítulo se lleva a cabo la implementación de dos sistemas de ataques sobre implementaciones ASIC del cifrador Trivium. Los sistemas de ataques utilizados han consistido por un lado en la manipulación de la señal de reloj del cifrador y por otro lado la manipulación de las señales de control. Para ello se ha utilizado el equipo de test Agilent 93000. En cuanto a los diseños del cifrador Trivium utilizados, se han considerado tanto las versiones estándar como las de baja potencia y salidas multi-bit. En este capítulo se presentan los datos más relevantes obtenidos en los ataques. Los resultados completos se recogen en el Anexo de esta memoria. Por último, en este capítulo se presentan las conclusiones obtenidas tras atacar a las diferentes implementaciones del cifrador.

- Capítulo 5: Criptoanálisis en cifradores Trivium con datos experimentales.

  En este capítulo se realiza la descripción, montaje y utilización de un sistema de criptoanálisis, el cual utiliza análisis diferenciales de los datos obtenidos para recuperar la información secreta (registro de estados interno) de forma experimental de cifradores Trivium estándar. Además, se ha realizado el diseño de un cifrador de flujo Trivium inverso mediante el cual es posible obtener la clave secreta y vector de inicialización a partir de un registro de estados interno conocido. Por último en este capítulo se describe el proceso para recuperar la clave secreta de forma experimental a partir de ataques realizados sobre implementaciones ASIC del cifrador Trivium estándar.

- Capítulo 6: Conclusiones y líneas futuras.

  En este capítulo se resumen los principales resultados aportados por este trabajo de Tesis, se analiza el cumplimiento de los objetivos y se presentan las principales conclusiones obtenidas tras el trabajo de investigación realizado. Además, se plantean posibles líneas de investigación futuras surgidas a partir de los resultados obtenidos a lo largo de la realización de la presente Tesis.

Esta Memoria también contiene una sección con la bibliografía utilizada para el desarrollo de esta Tesis y un Anexo que incluye de forma más exhaustiva los resultados obtenidos tras la inserción de fallos sobre las implementaciones ASIC, de todos los diseños del cifrador Trivium. Estos resultados, no presentados en el Capítulo 4, son para las versiones Trivium x2 estándar, Trivium x2 Low Power, Trivium x8 estándar, Trivium x8 Low Power, Trivium x16 estándar y Trivium x16 Low Power.

2. Criptografía, cifradores y criptoanálisis

En este capítulo, se expone una visión general del estado del arte de la las implementaciones hardware de circuitos criptográficos, centrándose en los algoritmos de clave privada, concretamente en el cifrador de flujo Trivium. Además en este capítulo se presentan las ideas principales del criptoanálisis, así como los diferentes tipos de análisis teóricos que permiten llevarlo a cabo y los tipos de ataques laterales a los que está expuesto cualquier dispositivo que incorpore capacidades criptográficas. De entre todos los métodos de análisis teóricos, en el caso del cifrador Trivium y en concreto para esta Tesis Doctoral, el más importante es el denominado Análisis Diferencial de Fallos o Differential Fault Analysis (DFA) en inglés. Este método ha sido ampliamente utilizado en una gran cantidad de dispositivos criptográficos y en especial con el cifrador Trivium, donde gracias a su aplicación es posible recuperar de forma teórica la información secreta del dispositivo criptográfico. Una vez introducida la idea general de los ataques a los sistemas criptográficos, se presentan las propuestas para llevar a cabo ataques laterales de forma experimental.

2.1. La criptografía

La criptografía como medio para proteger la información escrita es un arte tan antiguo como la propia escritura. Durante siglos permaneció estrechamente vinculada a entornos militares y diplomáticos, puesto que eran los únicos que tenían auténtica necesidad de ella. En la actualidad la situación ha cambiado de forma radical: vivimos inmersos en un mundo de comunicaciones electrónicas, teniendo al alcance un número ilimitado de dispositivos que transmiten y almacenan información. Gran parte de las actividades cotidianas se traducen

en intercambio de datos entre personas, dispositivos o instituciones, además del emergente campo de Internet de las cosas (IoT) [1] , [2]. Así pues, es necesario proteger este intercambio continuo de información y es entonces cuando la criptografía pasa de ser una exigencia de campos muy concretos a convertirse en una necesidad real del hombre de la calle, ya que una falta de protección en su información privada supondría una grave amenaza para su intimidad y su integridad.

El proceso general de un procedimiento criptográfico de cifrado y descifrado, o criptosistema, puede resumirse tal y como se muestra en la Figura 2.1. En este ejemplo puede observarse de forma esquemática la transmisión de información de un emisor a un receptor. La importancia de la criptografía es que en cualquier momento una tercera parte, denominado atacante, puede acceder al mensaje intercambiado y tratar de obtener la información sensible con diferentes fines.

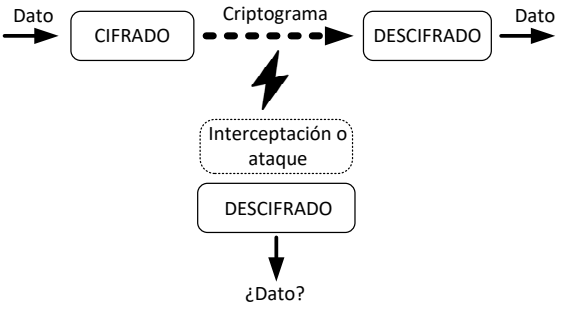

Figura 2.1.: Proceso de un criptosistema frente a un posible ataque.

El atacante, una vez capturado el mensaje cifrado, lleva a cabo un procedimiento de desencriptado[1], es decir, intenta a partir del dato cifrado, y sin conocimiento de la clave, recuperar el mensaje original. El escenario en el que se desarrolla este trabajo, se sitúa en el peor de los casos. El atacante tiene acceso al mensaje cifrado y conoce el algoritmo de cifrado, esto es, tiene información del dispositivo que cifra la información y por tanto sólo tiene como barrera la clave que está utilizando dicho dispositivo.

[1]Según DRAE en castellano no existe “encriptar" sino "cifrar". En casi toda la Tesis se usa reiteradamente el anglicismo encriptar y sus derivados. A lo largo del presente texto se van a utilizar ambas palabras indistintamente por su importancia en este campo científico.

Es aquí donde entra en juego el desarrollo de sistemas que permitan un elevado nivel de seguridad, es decir, sistemas que proporcionen la transmisión segura de información, donde se reduzca al máximo la vulnerabilidad de las comunicaciones.

Tanto las comunicaciones inalámbricas como las cableadas presentan vulnerabilidades frente a los ataques. Por un lado la tecnología inalámbrica, por su propia naturaleza, utiliza un medio inseguro. Mientras que, por otro lado, una comunicación cableada puede ser más segura, pero en un escenario donde un atacante tiene acceso al cable de transmisión, también presenta una grave vulnerabilidad. Estas consideraciones hacen que la seguridad sea un factor significativo que debe ser desarrollado.

Teniendo en cuenta lo descrito se puede afirmar que no existe una comunicación segura al cien por cien. Es por esto que a lo largo de la historia se han ido desarrollando y optimizando los sistemas que permitan minimizar los riesgos. En concreto, a través de la criptografía es posible dar solución a diferentes ámbitos de aplicación, garantizando que la información transmitida posea unas cualidades fundamentales, siendo estas confidencialidad, integridad, autenticidad, disponibilidad y no repudio [3].

En este sentido la confidencialidad consiste en lograr que la información permanezca secreta y solo pueda ser conocida por aquellos que tienen autorización para ello. La integridad abarca la necesidad de que la información haya sido manipulada o alterada desde su origen hasta su destino. Mientras que la autenticidad obliga a que tanto el origen como la información transmitida sean auténticos, asegurando que no se produzcan suplantaciones. En el caso de la disponibilidad se trata de la capacidad de un sistema a ser accesible y utilizable por los usuarios autorizados cuando estos lo requieran. Por último, el no repudio consiste en una autenticación que con un alto nivel de seguridad puede ser garantizado como genuino.

Considerando esto, existen dos grandes grupos de algoritmos para cifrar información: los basados en claves públicas y los basados en clave privada [4]. A este último grupo es al que pertenecen los que utilizan los cifradores de flujo para garantizar una comunicación segura. En la siguiente Figura 2.2 puede apreciarse una clasificación general de los criptosistemas actuales.

Dado que el objetivo del presente trabajo está enfocado a un dispositivo basado en un algoritmo de clave privada, se explicarán brevemente los algoritmos

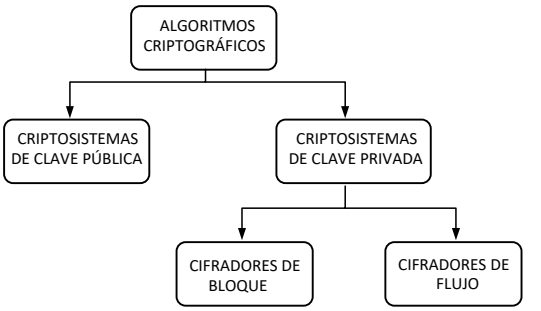

Figura 2.2.: Clasificación de los grandes grupos de criptosistemas.

basados en clave pública y se realizará una presentación más exhaustiva de los sistemas basados en clave privada.

2.1.1. Criptografía de clave pública

Cualquier receptor en un criptosistema de clave pública (o criptosistema asimétrico [5]) dispone de dos claves: una privada que debe mantener secreta y una pública que debe ser conocida por todas las restantes entidades que van a comunicar con ella.

Para el envío de un mensaje cifrado el emisor utiliza la clave pública, k_c, para cifrar el mensaje. Este mensaje solo puede ser descifrado con la clave privada, k_p, que posee el receptor. Se trata por tanto de procedimientos que no necesitan intercambio secreto de clave entre ambos comunicantes.

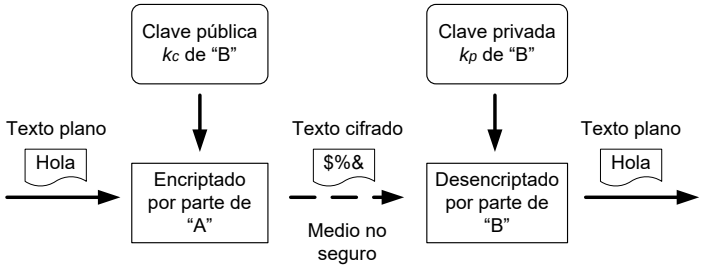

Figura 2.3.: Esquema de funcionamiento de la criptografía asimétrica o de clave pública.

El mecanismo para la transmisión de información cifrada utilizando este tipo de criptosistemas se muestra de forma esquemática en la Figura 2.3. Cuando un emisor "A" quiere enviar un mensaje cifrado a un receptor "B" lo primero que debe hacer es acceder a la clave pública k_c de "B". A continuación, "A" cifrará el mensaje a enviar con dicha clave pública y se lo enviará a "B", quien lo descifrará utilizando su clave privada k_p.

Desde el punto de vista teórico, los criptosistemas asimétricos se basan en la existencia de una función unidireccional que permita transformar un mensaje cualquiera en un mensaje cifrado mediante una operación computacionalmente fácil, pero que la operación inversa sea computacionalmente impracticable, excepto para quien posea una información secreta (de nuevo, la clave secreta).

Desde el punto de vista práctico, la criptografía asimétrica presenta las siguientes características:

1. Son criptosistemas considerablemente más complejos que los de cifrado simétrico, requiriendo muchos más recursos computacionales para su implementación (software o hardware).
2. La criptografía asimétrica podría ser vulnerable a ataques de texto plano elegido o fácilmente deducible. Esto quiere decir que si se sabe que el mensaje a transmitir pertenece a un conjunto de n posibles mensajes, al ser la clave del receptor pública, una tercera parte podría encriptar los n posibles y comparar resultados para deducir el mensaje enviado.
3. No se utilizan para encriptar mensajes, sino que se implementa para encriptar claves, siendo el uso más común de este tipo de criptosistemas el intercambio de claves secretas entre las distintas partes.

2.1.2. Criptografía de clave privada

En la criptografía de clave privada la clave de cifrado y de descifrado es la misma, es decir, se trata de una clave secreta que comparten el emisor y el receptor del mensaje. Debido a esta característica son denominados también criptosistemas simétricos. En este caso hay que suponer que tanto emisor como receptor se han puesto de acuerdo previamente en la determinación de la clave, o bien que existe un centro de distribución de claves que por un canal seguro se la ha hecho llegar a ambas partes. En cualquier caso se trata de procedimientos con clave compartida entre "A" y "B". En el esquema de la Figura 2.4 se

puede observar la forma de operar de la criptografía simétrica, donde "A" pretende enviar un mensaje a "B". Básicamente se trata de lo siguiente: estas partes acordarán una clave de manera secreta. Luego la parte "A" encriptará el mensaje (utilizando el algoritmo y la clave acordados) y lo enviará a la parte "B" a través del canal que se ha asumido inseguro. La parte "B", valiéndose del mismo algoritmo y clave, obtendrá el mensaje original, realizando el proceso de desencriptado.

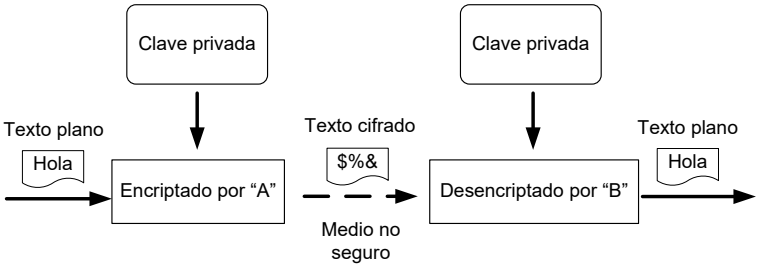

Figura 2.4.: Cifrado basado en sistema de clave privada.

Este tipo de criptografía presenta una serie de ventajas e inconvenientes. A continuación se enumeran algunos de los puntos más importantes:

1. Como principales ventajas se encuentran, en primer lugar, la velocidad de procesamiento, la cual es mucho más alta que la de los criptosistemas asimétricos debido al uso de operaciones lógicas y algebraicas más sencillas. Se trata de un tipo de cifrado muy fácil de usar y, además, es muy útil a la hora de cifrar datos personales debido a que sólo se utiliza una clave para encriptar y desencriptar. Por otro lado, la longitud de las claves es menor que la de los criptosistemas asimétricos, siendo la longitud mínima establecida por el *National Institute of Standards and* Technology (NIST) de 112 bits. Por lo tanto, su principal utilización, al contrario que el de la criptografía asimétrica, se centra en el intercambio de mensajes encriptados, pero no el de claves.

2. Entre sus principales desventajas se encuentran la necesidad de compartir la clave secreta, lo cual impone un problema de difícil solución entre las partes si no se dispone de un medio seguro para hacerlo. Además, existe el peligro de que muchas personas deban conocer una misma clave y la dificultad de almacenar y proteger muchas claves diferentes.

En los criptosistemas de clave privada se encuentran dos tipos básicos de cifradores, diferenciados en cuanto a cómo dividir el texto plano para abordar la tarea de cifrado. Estos criptosistemas son los denominados cifradores de bloques y cifradores de flujo. A continuación se describen ambos.

2.1.2.1. Cifradores de bloque

Los cifradores de bloques cifran los mensajes con grupos o bloques de n bits. Los tres sistemas que han sido normalizados hasta la fecha son el *Data Encryption Standard* (DES) [6], ya retirado, que cifra bloques de 64 bits con claves de 56 bits; el *Triple Data Encryption Standard* (TDES) [7], que cifra bloques de 64 bits con claves de 56, 112 y 128 bits y el *Advanced Encryption Standard* (AES) [8], que cifra bloques de 128 bits con claves de 128, 192 y 256 bits.

En la Figura 2.5 puede observarse un esquema de un cifrado por bloques donde los datos son cifrados y descifrados en grupos de n bits. La característica principal de este tipo de cifradores consiste en que cada bloque se cifra de igual forma, independientemente del lugar que ocupe en el mensaje, de manera que todos los bits del bloque se cifran conjuntamente, participando en operaciones que tratan de oscurecer las posibles relaciones que tuviesen en el mensaje original.

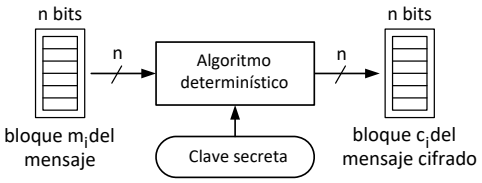

Figura 2.5.: Esquema simplificado de un cifrado por bloques.

Todos los cifradores de bloque tienen las siguientes propiedades:

- Dependencia entre bits: en cada bloque, cada bit del texto cifrado es una función compleja de todos los bits de la clave y todos los bits del bloque de texto plano.
- Cambio de los bits de entrada: un cambio de un bit en un bloque del mensaje original produce el cambio de aproximadamente el 50 % de los bits del bloque del mensaje cifrado.

- Cambio de los bits de clave: un cambio en un bit de la clave produce, aproximadamente, el cambio de la mitad de los bits del mensaje cifrado.
- Un error en la transmisión de un texto cifrado se propaga a todo el bloque del que forma parte, produciendo un conjunto de errores después del descifrado del 50 % de los bits del bloque afectado, lo que equivale a que todo el bloque sea inteligible.

Debe tenerse en cuenta que un cambio total de un mensaje es cualquier otro mensaje cuyos bits coincidan con el original en un 50 % y, consecuentemente, difieran en otro 50 %, por lo que la correlación cruzada entre ambos mensajes valdría 0. En cambio, dos mensajes que difieran en todos sus bits, es decir, que fueran complementarios, presentan una correlación de valor -1. Así pues, como operación de cifrado resultaría inútil, pues, aunque a primera vista disimulase el mensaje, bastaría con invertir los valores de todos los bits del segundo mensaje para obtener el primero.

En cuanto a la arquitectura del cifrado en bloque, todos se componen de cuatro elementos:

- Una transformación inicial.
- Una función criptográficamente débil iterada r veces.
- Una transformación final.
- Un algoritmo de expansión de clave.

La transformación inicial en algunos sistemas carece de significación criptográfica y su función consiste simplemente en aleatorizar los datos de entrada (para ocultar bloques de datos), pero en otros algoritmos puede tener significación criptográfica para entorpecer ataques por análisis lineal o diferencial.

Las vueltas intermedias consisten en una función no lineal, complicada, de los datos y de la clave, que puede ser unidireccional o no. La función no lineal puede estar formada por una sola operación muy compleja o por la sucesión de varias transformaciones simples. Estas vueltas no han de tener estructura de grupo algebraico, para que el conjunto de varias pasadas sucesivas con sus subclaves correspondientes no sean equivalentes a una pasada única con una subclave diferente. La transformación final puede tener o no tener significación criptográfica y su función consiste en invertir la transformación inicial.

Por último el algoritmo de expansión de clave tiene por objeto convertir la clave de usuario, normalmente con una longitud comprendida entre 56 y 256 bits,

en un conjunto de subclaves que pueden estar constituidas por varios cientos de bits en total. Conviene que sea unidireccional y que el conocimiento de una o varias subclaves intermedias no permita deducir las subclaves anteriores o siguientes.

En cuanto a las propuestas de cifradores actuales, existen múltiples algoritmos de cifradores de bloque aplicables al ámbito de la criptografía de bajo consumo o *lightweight.* Entre estas propuestas es posible encontrar cifradores de bloques como el cifrador LED [9], SIMON, SPECK [10] y PRESENT [11] entre otros.

2.1.2.2. Cifradores de flujo

Antecentes

La base del cifrado de flujo se encuentra en un procedimiento de cifrado por sustitución denominado cifrado Vernam [12], patentado en el año 1919 por G.S. Vernam.

Durante mucho tiempo, en ambientes criptográficos se creyó que el cifrado Vernam era un procedimiento seguro, aunque no existía una demostración rigurosa de ello. A día de hoy, este hecho puede ser corroborado. En el cifrado de Vernam se cumplen las siguientes condiciones:

1. La clave es una secuencia binaria perfectamente aleatoria.
2. La clave se utiliza solamente una vez, pues para cada nuevo proceso de cifrado/descifrado es necesario usar una nueva secuencia aleatoria, distinta de las anteriores, que sería la clave del nuevo proceso.
3. El proceso de cifrado/descifrado se reduce a una operación modular elemento a elemento del mensaje.

Dentro del panorama criptográfico actual, el cifrado Vernam es el único procedimiento incondicionalmente seguro o con una seguridad matemáticamente demostrable.

Aunque este método ofrece máximas garantías de seguridad criptográfica, presenta sin embargo un inconveniente muy evidente y es que requiere un dígito de clave secreta por cada dígito de texto plano que hay que cifrar o descifrar. Además hay que añadir que esa clave sólo se utiliza una vez, por tanto para cada mensaje hay que hacer llegar a ambos comunicantes por un canal seguro

tanta cantidad de secuencia aleatoria como bits tenga el mensaje original. Por tanto se trata de un procedimiento de cifrado incondicionalmente seguro, pero inviable para su implementación práctica.

A partir de aquí, el procedimiento fue modificado para convertirlo en un procedimiento viable. Para ello, en lugar de una secuencia aleatoria, se utiliza una secuencia pseudoaleatoria obtenida a partir de un generador pseudoaleatorio que, a partir de una clave corta conocida por el cifrador y descifrador, genere simultáneamente en emisión y recepción un secuencia pseudoaleatoria de la longitud deseada. Esta modificación del cifrado Vernam es lo que se conoce como cifrador de flujo.

Aspectos generales

Según lo descrito, el cifrado de flujo se presenta como una alternativa viable del cifrado Vernam, ya que goza de un buen nivel de seguridad. En la Figura 2.6 se puede apreciar una representación esquemática del cifrado de flujo. Como puede observarse, realizando una operación XOR del texto plano con el flujo cifrado, es posible obtener el texto cifrado.

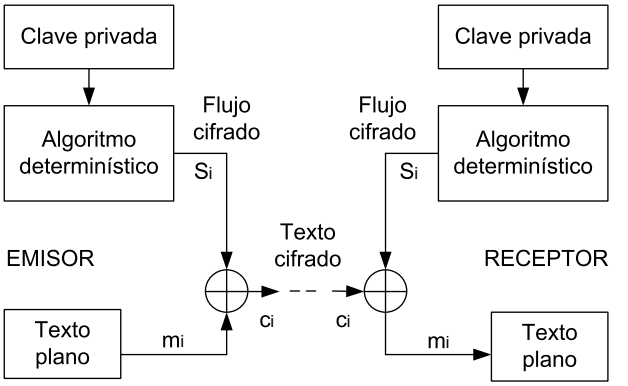

Figura 2.6.: Representación de un cifrado de flujo.

Se trata de un sistema cuyo emisor y receptor, poseen una clave corta (secreta) y un algoritmo público, que genera una secuencia pseudoaleatoria. El sistema es más seguro cuanto más se aproxime la secuencia a una secuencia aleatoria. Para generar la secuencia, la única información secreta que tienen que compartir emisor y receptor es la clave.

La idea fundamental del cifrado de flujo consiste en generar una secuencia larga e imprevisible de dígitos binarios a partir de una clave corta elegida de forma aleatoria. Se trata de un sistema cómodo, sencillo y rápido.

Características

Es difícil evaluar cuándo una secuencia binaria es lo suficientemente segura para su utilización, ya que no existen pautas que ayuden a garantizarlo. Sin embargo se pueden establecer ciertas características que ayudan a verificar su correcta aplicación [13]:

- **Período:** La longitud de la secuencia cifrante debe ser al menos como la longitud de la secuencia que hay que cifrar. En la práctica es fácil generar secuencias con períodos del orden de 10^{38} bits o incluso superiores.
- **Distribución de bits:** En [14], Golomb formula tres postulados donde establece qué debe satisfacer una secuencia binaria finita para poder ser denominada secuencia pseudoaleatoria. Estos son "Propiedad del balance", "Propiedad de funcionamiento" y "Propiedad de doble nivel de correlación", denominados R-1, R-2 y R-3 respectivamente.
- **Imprevisibilidad:** La secuencia debe poseer esta propiedad, lo que quiere decir que dada una secuencia de cualquier longitud, una tercera persona no debería poder predecir el siguiente dígito con una probabilidad superior a 1/2. Una medida de imprevisibilidad de una secuencia es su complejidad lineal, donde esta debe ser lo más grande posible, siendo al menos la mitad de la longitud del período.
- **Test de aleatoriedad:** Generalmente el punto de partida de un ataque criptográfico sobre estos cifradores se basa en la posibilidad de distinguir la secuencia cifrante de una secuencia perfectamente aleatoria. Para ello se aplica una serie de test estadísticos que eliminan las secuencias que no poseen las propiedades estadísticas que deben poseer las secuencias aleatorias.

Los generadores más conocidos para la obtención de secuencias pseudoaleatorias son los que se basan su estructura en registros de desplazamiento realimentados no linealmente.

Los registros de desplazamiento realimentados son estructuras básicas para la generación de secuencias binarias, constituidas por:

- L celdas de memoria conectadas de forma serie.
- Una función de realimentación F que permite expresar cada elemento a_i de la secuencia con $i > L$ en función de los L elementos anteriores.

Si la función F es lineal se habla de un registro de desplazamiento con realimentación lineal (LFSR, *Linear Feedback Shift Register*), si no es lineal, se habla de un registro de desplazamiento con realimentación no lineal (NLFSR, *Non Linear Feedback Shift Register*).

Por otro lado se denomina estado del registro interno al contenido binario de los registros en cada ciclo de reloj. El estado inicial del registro es el contenido en el momento de empezar el proceso de generación, el cual debe ser iniciado con una semilla.

2.1.3. Proyecto eSTREAM

El Proyecto eSTREAM comenzó en noviembre de 2004, siendo una iniciativa de la Comunidad Europea, dentro de la segunda Red Europea de excelencia en Criptografía (ECRYPT II) [15, 16]. Su objetivo era seleccionar un algoritmo de cifrado de flujo que pudiera considerarse como estándar de cifrado de flujo.

Tras 6 meses de la convocatoria, se presentaron 34 algoritmos de cifrado de flujo, siendo éstos evaluados por una comisión de reconocido prestigio internacional. Tras sucesivas rondas eliminatorias, 16 algoritmos llegaron a la fase final, de los cuales 8 estaban orientados a su implementación software, Perfil 1 (SW), y otros 8 estaban orientados a implementaciones hardware, Perfil 2 (HW). La Tabla 2.1 recoge los nombres de estos 16 algoritmos.

Más tarde en 2008, la comisión al cargo de la evaluación de los algoritmos candidatos presentó el resultado de sus deliberaciones, cuyo resultado fue un portfolio de 8 algoritmos, la mitad de ellos software y la otra hardware. Tras esto, el proyecto ha ido siendo sometido a revisiones, la última en enero de 2012. En esta ocasión el resultado fue de 4 algoritmos software y 3 hardware. La Tabla 2.2 recoge los nombres de estos finalistas.

De estos finalistas, los importantes para esta Tesis son los de perfil hardware, en concreto, el cifrador de flujo Trivium que es sobre el que se han realizado los ataques. A continuación se realiza una breve descripción de los tres finalistas hardware.

Tabla 2.1.: Listado de los algoritmos en fase final del proyecto eSTREAM.

Perfil 1 (SW)	Perfil 2 (HW)
CryptMT (v.3)	DECIM (v.2)
Dragon	Edon80
HC (HC-128)	F-FCSR (F-FCSR-H v2)
LEX (LEX-128)	Grain (Grain-128)
NLS (NLS v.2)	MICKEY (MICKEY 2.0)
Rabbit	Moustique
Salsa20	Pomaranch (v.3)
Sosemanuk	Trivium

Tabla 2.2.: Finalistas del proyecto eSTREAM (portfolio 2012).

Perfil 1 (SW)	Perfil 2 (HW)
HC (HC-128)	Grain v1
Rabbit	MICKEY (MICKEY 2.0)
Salsa20/12	Trivium
Sosemanuk	–

Grain v1

El cifrador Grain se puede describir como una familia de cifradores de flujo síncronos con notable eficiencia hardware [17]. La versión inicial utilizaba una clave de 80 bits y un vector de inicialización de 64 bits, pero las primeras pruebas del proyecto eSTREAM pusieron en duda su seguridad. Tras esto se revisaron las especificaciones y se presentó Grain v1, que son en realidad dos propuestas de cifradores de flujo, uno que utiliza una clave de 80 bits (con un vector de inicialización IV de 64 bits) y otro que utiliza una clave de 128 bits (con un IV de 80 bits). Elegante y simple, consta de dos registros de desplazamiento, uno con realimentación lineal y otro con realimentación no lineal. Estos registros y los bits de salida están unidos mediante funciones booleanas muy sencillas. Como los demás cifradores de flujo, se necesita una fase de inicialización antes de poder generar una secuencia válida de bits.

MICKEY 2.0

Su nombre es un acrónimo de "*Mutual Irregular Clocking KEY stream generator*" el cual describe el comportamiento esencial del cifrador [18]. Formado por dos registros de desplazamiento realimentados de 100 bits, uno lineal y otro no lineal, cada uno de ellos está temporizado irregularmente bajo el control del otro. Los mecanismos específicos de temporización contribuyen a la fuerza criptográfica del cifrador al tiempo que ofrece garantías de período y pseudo-aleatoriedad. La especificación de su estado consiste en que cada clave puede utilizar hasta 2^{40} vectores de inicialización (IV) del mismo tamaño, pudiendo producir con ello 2^{40} bits de flujo por cada par de clave e IV.

Este cifrador puede ser implementado con pocos recursos hardware, haciéndolo un buen candidato para aplicaciones de baja potencia. Sin embargo, el mecanismo de temporización irregular significa que no podría ser paralelizado fácilmente.

Trivium

El cifrador Trivium, cuyos autores son C. De Cannière y B. Preneel [19], es un generador de secuencia síncrono orientado a su implementación hardware, diseñado para encontrar un equilibrio entre simplicidad, seguridad y velocidad. Posee una estructura simple y elegante, considerándose un ejemplo de cómo el diseño de un cifrador puede ser simplificado sin sacrificar su seguridad, velocidad o flexibilidad. En la sección siguiente se estudia con más detenimiento este cifrador de flujo, siendo descritas algunas variantes de diseño en el Capítulo 3, las cuales han sido utilizadas durante la realización de la presente Tesis en las implementaciones sobre tecnologías FPGA y ASIC.

2.2. Cifrador de flujo Trivium

Se trata de un cifrador de flujo síncrono diseñado para generar un flujo cifrado de hasta 2^{64} bits a partir de una clave de 80 bits y un IV de 80 bits. A diferencia de otros generadores que posteriormente han ampliado en sucesivas versiones la longitud de sus parámetros, el cifrador Trivium no ha modificado lo más mínimo su diseño inicial y mantiene intactas las especificaciones requeridas. Su funcionamiento consta de dos fases bien diferenciadas: fase de inicialización y fase de generación de flujo cifrado. El estado interno del generador de flujo, compuesto por 288 bits, se inicia utilizando la clave secreta y el IV, junto a otros 125 ceros y 3 unos en posiciones preestablecidas, dejando pasar los primeros 1152 ciclos de reloj. A partir de ese ciclo de reloj comienza la fase de generación, produciendo como salida un bit de cifrado en cada ciclo de reloj.

En concreto, el procedimiento puede describirse tal y como se muestra a continuación en el siguiente Algoritmo 1. El algoritmo se inicia introduciendo los 80 bits de la clave en las posiciones 0 a 79 y los 80 bits del vector de inicialización en las posiciones 93 a 172, completando el resto de las posiciones con ceros, excepto los bits s_{285} ,s_{286} y s_{287} que se ponen a 1. Luego se rota el registro de estados 4 veces completamente (1152 ciclos de reloj en total) antes de poder generar una secuencia válida de bits de cifrado.

Algoritmo 1 Trivium

1: $(S_0, S_1, ..., S_{92}) \leftarrow (k_0, k_1, ..., k_{79}, 0, ..., 0)$
2: $(S_{93}, S_{94}, ..., S_{176}) \leftarrow (iv_0, iv_1, ..., iv_{79}, 0, ..., 0)$
3: $(S_{177}, S_{178}, ..., S_{287}) \leftarrow (0, ..., 0, 1, 1, 1)$
4: **for** $i = \{1\}$ *to* n **do**
5: $t_1 \leftarrow S_{65} \oplus S_{90}S_{91} \oplus S_{92} \oplus S_{170}$
6: $t_2 \leftarrow S_{161} \oplus S_{174}S_{175} \oplus S_{176} \oplus S_{263}$
7: $t_3 \leftarrow S_{242} \oplus S_{285}S_{286} \oplus S_{287} \oplus S_{68}$
8: $(S_0, S_1, ..., S_{92}) \leftarrow (t_3, S_1, ..., S_{92})$
9: $(S_{94}, S_{95}, ..., S_{177}) \leftarrow (t_1, S_{94}, ..., S_{176})$
10: $(S_{178}, S_{179}, ..., S_{288}) \leftarrow (t_2, S_{178}, ..., S_{287})$
11: **end for**

En la Figura 2.7, se muestra la representación esquemática del cifrador Trivium con sus tres registros de desplazamiento y sus realimentaciones. Estos registros son registros de desplazamiento con realimentaciones no lineales (NLFSR).

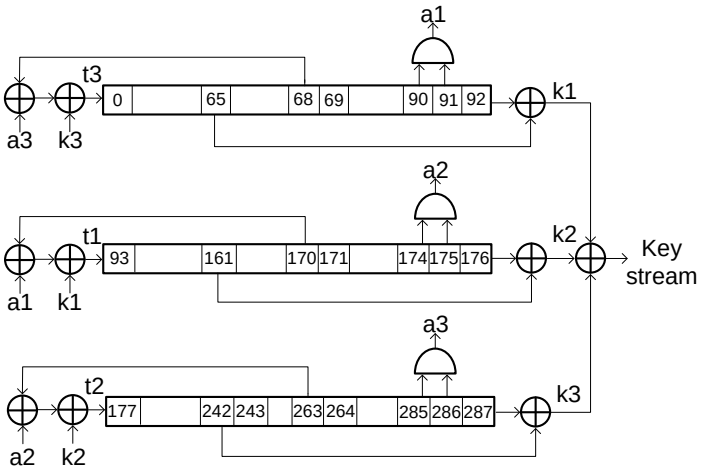

Figura 2.7.: Esquema del cifrador Trivium con sus tres registros internos.

Por último, en cuanto al aspecto de implementación y seguridad, se puede afirmar que el cifrador Trivium tiene un algoritmo orientado al hardware que pretende ser compacto en entornos con restricciones sobre el número de puertas lógicas, eficiente sobre arquitecturas con limitaciones en cuanto a recursos computacionales y rápido en aplicaciones que requieran una alta velocidad de cifrado. Este algoritmo reúne todas esas características y, hasta el momento, se ha mostrado como el más rápido de todos los presentados en eSTREAM [15].

Respecto a su seguridad, puede decirse que, hoy en día, se considera inmune a los ataques del tipo correlación, *guess and determine* y ataques algebraicos conocidos. Esto tiene todavía mayor mérito teniendo en cuenta que, dada su simplicidad y velocidad, la vulnerabilidad del cifrador Trivium ha estado y sigue estando en el punto de mira de la comunidad criptográfica internacional.

2.3. Criptoanálisis

El criptoanálisis es la rama de la criptografía que estudia cómo puede ser puesta en peligro la integridad de los algoritmos y los circuitos criptográficos. Se trata de un aspecto fundamental en el desarrollo de algoritmos criptográficos

seguros, ya que para diseñar un sistema de cifrado robusto y fiable, es necesario que los desarrolladores de dicho sistema sean capaces de entender y definir dónde se encuentran las debilidades potenciales del mismo con el fin de eliminar dichas vulnerabilidades [20].

En la Figura 2.8 se presenta un esquema general del ámbito del criptoanálisis sobre los criptosistemas de clave simétrica. Como puede verse, existen dos grande ramas, por un lado los análisis de vulnerabilidades a nivel de algoritmo y, por otro lado, los análisis de vulnerabilidad a nivel de las implementaciones físicas del dispositivo criptográfico. En la primera de las ramas se analiza la seguridad del propio algoritmo matemático que implementa el sistema criptográfico. Se entiende por nivel de seguridad a la habilidad que posee un algoritmo para proteger la información privada. Cada sistema tendrá su propio nivel de seguridad de acuerdo tanto a su formulación matemática para llevar a cabo la encriptación, como a la longitud de su clave. Cuantos más bits tiene su clave, más seguro es el dispositivo criptográfico. Teniendo en cuenta los recursos computacionales actuales, una longitud de clave menor a 56 bits implicaría que el sistema no fuese seguro dado que podría ser atacado por fuerza bruta con un coste asumible. En el caso de la criptografía *lightweight*, el NIST recomienda una longitud mínima de clave de 112 bits con el objetivo de mantener un equilibrio entre una complejidad hardware y un nivel de seguridad razonable [21].

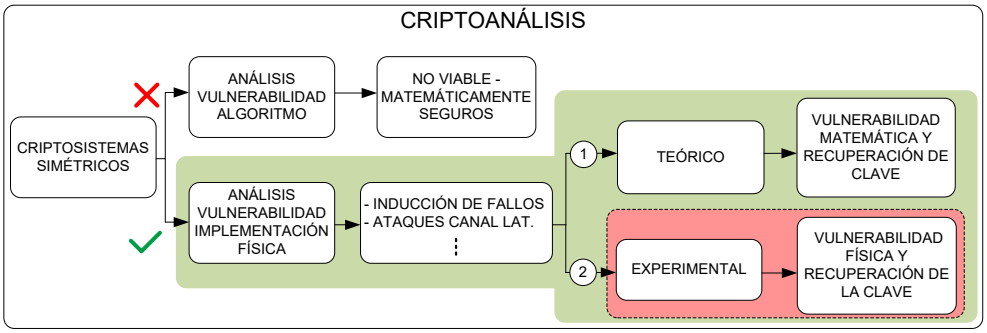

Figura 2.8.: Criptoanálisis en sistemas criptográficos simétricos.

Teniendo en cuenta esto, los análisis o ataques sobre los propios algoritmos que implementan los sistemas criptográficos actuales son escasamente viables, ya que matemáticamente las funciones que implementan los nuevos dispositivos criptográficos y la longitud mínima de clave establecida los hacen seguros

frente a este tipo de prácticas. Sin embargo, los que sí han demostrado ser eficientes son los análisis sobre las implementaciones físicas. En el esquema de la Figura 2.8 este tipo de análisis se sitúa en la segunda rama del criptoanálisis de los criptosistemas simétricos.

Son numerosas las técnicas para llevar a cabo un criptoanálisis sobre una implementación física, desde los ataques invasivos hasta los conocidos como ataques laterales. Dada la gran cantidad de trabajos y métodos en función del tipo de dispositivo criptgráfico (ya sean cifradores de bloque o flujo por ejemplo), hemos distinguido dos grupos de trabajos. El primer grupo incluye aquellos trabajos que utilizan un análisis de vulnerabilidad teórico fundamentado en un modelo del dispositivo (Figura 2.8, 1). El segundo grupo de trabajos basa su análisis de vulnerabilidad en implementaciones reales donde se prueba experimentalmente la posibilidad de conseguir los supuestos asumidos de forma teórica (Figura 2.8, 2).

La gran mayoría de los trabajos reportados en la bibliografía se encuentran en el primero de estos dos grupos, asumiendo que los supuestos teóricos son aplicables a los circuitos mediante la utilización de alguna de las técnicas de ataque existentes. Sin embargo, los trabajos que pueden ser clasificados en el segundo de los grupos (marcado en la Figura 2.8, 2 en rojo) son escasos.

La implementación de un análisis práctico a veces no es factible debido a los requisitos del sistema desarrollado y a la cantidad de variables que hay que tener en cuenta a la hora de llevar un análisis teórico a la práctica. Son numerosos los ejemplos de la dificultad o controversia existente a la hora de llevar a la práctica los análisis teóricos. Por ejemplo, en uno de los primeros trabajos en este campo [22], los autores afirman que “El uso de modelos de fallos transitorios es considerado irrealista, pero nosotros creemos que un ataque basado en fallos inducidos por radiación es posible.”. Otro ejemplo más actual se puede encontrar en [23], donde los autores afirman que un ataque mediante el uso de fallos experimentales es difícil de implementar y que por tanto su trabajo es una primera aproximación de forma teórica.

Es debido a esto que el número de trabajos llevados a la práctica de forma satisfactoria es despreciable frente a los desarrollados asumiendo de forma teórica el escenario del análisis y del ataque. Por lo tanto, para poder tener una visión general tanto de los ataques experimentales como de los análisis teóricos, en la siguientes subsecciones se analizan las principales técnicas que permiten llevar a cabo de forma práctica el criptoanálisis de un criptosistema

y las técnicas de análisis teóricas empleadas para estudiar la vulnerabilidad de un criptosistema.

2.3.1. Descripción de las técnicas de ataque a los circuitos

Los ataques a la implementación son aquellos que tratan de aprovechar las características físicas de tales implementaciones para extraer información que permita obtener las claves de los algoritmos utilizados, rompiendo así su seguridad. Los riesgos para el sistema de cifrado se basan fundamentalmente en la existencia de ciertos canales a través de los cuales puede salir información sensible de la zona considerada segura del dispositivo, o en la posibilidad de generar fallos en el circuito electrónico para, mediante el análisis de los efectos que esos fallos tienen en el comportamiento del sistema, deducir información sobre las claves.

Las distintas técnicas para realizar estos ataques difieren significativamente en términos de coste, tiempo, equipamiento y experiencia necesitados. Hay diferentes formas de organizar los tipos de ataques físicos, distinguiendo un primer criterio donde se encuentran los ataques pasivos y los ataques activos. A su vez, otro criterio es el que los divide entre ataques invasivos, ataques semi-invasivos y ataques no invasivos. En la Tabla 2.3 se muestra una clasificación de los ataques físicos más comunes. Cabe destacar que los ataques físicos son menos generales que los análisis teóricos clásicos, puesto que son específicos para cada tipo de implementación, pero a menudo son mucho más potentes.

Según [24], los ataques a implementaciones físicas pueden ser clasificados como:

1. Ataques activos: El ataque activo es aquel donde las entradas y/o el entorno del dispositivo criptográfico es manipulado (alterado) con el objetivo de obtener un funcionamiento anormal. La obtención de la clave se lleva a cabo explotando dicho comportamiento anormal del dispositivo.
2. Ataques pasivos: El ataque pasivo es el que observa el comportamiento durante el procesado del algoritmo, sin modificarlo, dentro de sus especificaciones. La clave se obtiene observando alguna de las propiedades físicas del dispositivo, como por ejemplo el tiempo de ejecución o el consumo de potencia.

Tabla 2.3.: Clasificación de ataques físicos.

Ataques	Activos	Pasivos
Invasivos	Modificación física	– –
	del dispositivo (FIB)	– –
Semi-invasivos	Fallos inducidos por:	Ataques por sonda
	láser o campos magnéticos	o medida de fotones
No invasivos	Fallos inducidos por:	Análisis de:
	Alimentación	Tiempo de ejecución
	Señal de reloj	Potencia consumida
	Temperatura	Campo EM

Una segunda forma de clasificar los ataques atiende a si se altera o no el propio dispositivo:

1. Ataques invasivos: Este tipo de ataques son los más fuertes que pueden montarse sobre un sistema criptográfico. Para este tipo de ataques no existen límites respecto a lo que se puede hacer contra el dispositivo para recuperar su clave. Se basan en la manipulación del dispositivo, teniendo acceso directo a partes internas del circuito, pero en cuanto a coste de equipamiento, son los que más recursos necesitan. Un ejemplo de estos ataques son [25–27].
2. Ataques semi-invasivos: En estos ataques el dispositivo es también manipulado. Pero al contrario que los ataques invasivos, no existe contacto eléctrico directo con el chip, permaneciendo las capas del circuito intactas. El objetivo de estos ataques es inducir fallos en el dispositivo, produciendo estos mediante rayos-X, campos electromagnéticos, o luz pulsada. Estos ataques no requieren un equipamiento costoso normalmente. Sin embargo el esfuerzo total para conseguir un ataque de este tipo suele ser relativamente alto. En particular, el proceso para determinar la posición correcta para un ataque en la superficie del chip requiere mucho tiempo y experiencia. Los trabajos más importantes en este campo son [27–29].
3. Ataques no invasivos: En este tipo de ataques el dispositivo criptográfico es atacado de forma directa solo a través de los canales disponibles. El dispositivo no es alterado permanentemente y por tanto no existe evi-

dencia de que se haya realizado un ataque. Estos ataques pueden llevarse a cabo con relativa poca experiencia y equipamiento. Por lo tanto, estos ataques suponen una amenaza práctica seria contra la seguridad de los sistemas criptográficos.

Los ataques no invasivos han recibido gran atención durante los últimos años. Por un lado, se encuentran los *pasivos no invasivos*, también conocidos como ataques de canal lateral (*side channel attacks* en inglés). Los ataques más importantes que se encuentran este grupo son los ataques temporales [30], ataques de análisis de potencia [31–33] y los ataques electromagnéticos [34–36]. La idea básica de estos ataques es determinar la clave secreta del dispositivo criptográfico mediante la medida del tiempo de ejecución, el consumo de potencia y su campo electromagnético. Por otro lado, se encuentran los ataques *activos no invasivos*. El objetivo de estos ataques es insertar fallos en el dispositivo sin alterarlo físicamente. Los fallos pueden ser inducidos por ejemplo mediante la alteración de la señal de reloj, la tensión de alimentación o cambiando la temperatura ambiente.

En el caso de la presente Tesis, los ataques más interesantes para ser utilizados en un sistema de ataque experimental son los pertenecientes al último grupo de los ataques descritos, es decir, los ataques activos no invasivos o por inserción de fallos. Por lo tanto, en la siguiente subsección se presentan las diferentes técnicas para esta clase de ataques con el objetivo de poder tener una visión general, mediante la cual podamos elegir cuál de ellas es la más apropiada para diseñar un sistema de ataques.

2.3.2. Técnicas de inserción de fallos

Una de las referencias bibliográficas más conocidas y utilizadas es la titulada *The Sorcerer's Apprentice Guide to Fault Attack* [37]. En este trabajo se hace una revisión de una gran variedad de métodos que pueden ser utilizados para introducir fallos en circuitos integrados, explicando ejemplos de ataques y presentando una serie de contramedidas para tratar de prevenir estos ataques. También en [38] y [39] se lleva a cabo una revisión de las técnicas de inserción de fallos y de contramedidas para evitar dichos fallos.

Como se ha descrito anteriormente en la clasificación de estos ataques, el objetivo es manipular el dispositivo para alterar su funcionamiento normal.

Una vez manipulado, la salida del dispositivo debe ser diferente a la correcta. Utilizando una o varias salidas erróneas y la salida correcta y haciendo uso de ecuaciones que relacionan las salidas erróneas con la clave, un atacante puede obtener la clave secreta.

Para poder llevar a cabo estos ataques se requiere un conocimiento amplio del dispositivo y del sistema criptográfico. Además, la eficacia del ataque depende del modelo de fallo utilizado. El modelo de fallo es una representación de cómo el fallo influye en el comportamiento del dispositivo, haciendo una hipótesis de cuántos bits han sido modificados y cuál ha sido el impacto de la modificación sobre ellos. A continuación se describen algunas de las técnicas de inducción de fallos.

- **Calor/Radiación infrarroja:** Si se hace funcionar un circuito fuera de los márgenes de temperatura entre los que se garantiza un correcto funcionamiento, las características temporales y de tensión pueden verse alteradas hasta el punto de provocar fallos en el funcionamiento normal del circuito. Estos fallos pueden afectar a una zona localizada del circuito, pero también pueden provocar fallos generalizados. Los trabajos presentados en [40] y [29] son dos ejemplos de estos ataques.
- **Picos de tensión :** Un exceso o una disminución del voltaje suministrado al chip por encima del nivel de tolerancia de los dispositivos (típicamente un 10 %), puede provocar fallos en las operaciones combinacionales o en los bits almacenados en los biestables. De nuevo estos fallos pueden afectar a una parte del circuito o provocar fallos generalizados. Un ejemplo de estos ataques se presenta en el trabajo [41].
- **Variaciones en la frecuencia de reloj:** En circuitos síncronos es posible introducir fallos reduciendo el tiempo entre dos flancos activos de la señal de reloj. Uno o varios biestables almacenan un valor antes de que éste esté estable en su entrada. Esta técnica tiene la característica de que si se controla bien el tiempo entre dos flancos activos, entonces no se introducen fallos generalizados en el circuito, sino únicamente en aquellos biestables cuyas entradas pasan por los caminos con mayor retraso. Esta técnica fue utilizada para obtener resultados experimentales en el trabajo presentado por Street y Lafayette [42], donde se aplican estos ataques a dispositivos RSA y DES. Otros trabajos donde se llevan a cabo estas técnicas son por ejemplo [43], [44] y [45].

- **Ataques combinados:** Estos ataques engloban, como su nombre indica, las combinaciones de varios de los ataques descritos anteriormente. Es posible realizar ataques físicos alterando de forma simultánea la señal de reloj y la tensión de alimentación. De esta forma se pueden conseguir situaciones más favorables para la introducción de fallos. También se pueden combinar dos o más ataques por canal lateral, utilizando por ejemplo el consumo de potencia y emanaciones electromagnéticas simultáneamente [46, 47].

De estas técnicas, la más interesante para nuestros propósitos es aquella que manipula la señal de reloj del sistema criptográfico para insertar fallos en su interior. Esta técnica permite controlar de forma precisa el ciclo en el que se intenta insertar el fallo y además requiere únicamente de equipos habituales en un laboratorio de electrónica. También es posible utilizarla con diferentes tecnologías como pueden ser FPGA o ASIC. Por otro lado, es posible combinarla con diferentes condiciones de funcionamiento como por ejemplo la tensión de alimentación o la temperatura del circuito.

2.3.3. Análisis por inyección de fallos en implementaciones físicas

A la hora de llevar a cabo ataques sobre las implementaciones físicas es necesario, además de las técnicas de inserción de fallos descritas en las subsecciones anteriores, realizar un análisis teórico que permita explotar la información obtenida del circuito a partir de los fallos generados. Debido a esto, es muy importante aclarar la imposibilidad de separar el análisis teórico del algoritmo criptográfico de los ataques físicos realizados sobre el criptosistema, ya sean estos ataques realizados de forma teórica mediante suposiciones o realizados experimentalmente en una implementación física. Este análisis teórico no debe ser confundido con los análisis de vulnerabilidad matemática de los propios algoritmos que implementan el sistema criptográfico anteriormente descritos. Este concepto es importante debido a que, por un lado, un ataque de inserción de fallos a un criptosistema no sirve de nada sin un estudio matemático del propio algoritmo y, por otro lado, un estudio matemático por sí solo no sirve de nada ya que los criptosistemas son matemáticamente seguros. Por lo tanto, ambos conceptos van de la mano y es necesario entender que un ataque a una implementación, ya sea teórico o experimental, implica un estudio del algoritmo.

Entre las posibles técnicas de análisis teóricos de un criptosistema como pueden ser los análisis lineales o los análisis algebraicos, entre otros, la técnica más interesante para el desarrollo de esta Tesis, en relación al cifrador Trivium, es el denominado análisis diferencial de fallos, en inglés *Differential Fault Analysis* (*DFA*)[2] . Debido a esto, a continuación solo se describe dicha técnica, pero para mayor información sobre las demás, consultar [20, 48].

La técnica del DFA es una de las técnicas más influyentes dentro del campo del criptoanálisis. Ésta consiste en recuperar la clave secreta mediante un análisis matemático donde se compara la relación que existe entre las diferencias de una salida generada por el cifrador en su modo de funcionamiento correcto y erróneo. Esta técnica surgió a partir de la técnica *Differential Cryptanalysis* presentada originalmente por los autores Biham y Shamir [49], donde se realizaba un breve análisis teórico sobre el cifrador *Data Encryption Standard* (DES). Tras este trabajo, estos mismos autores continuaron aplicando el concepto del differential cryptanalysis como una nueva técnica para el estudio de vulnerabilidad sobre diferentes cifradores, en este caso sobre el cifrador orientado a software Feal[50, 51]. Pero no fue hasta la presentación de los trabajos [52, 53], donde se llevaron a cabo una serie de mejoras y fue posible realizar los primeros estudios completos sobre el cifrador DES, demostrando la importancia de esta técnica.

Esta novedosa técnica atrajo el interés de la comunidad científica y no pasó demasiado tiempo hasta que en septiembre de 1996, Boneh, Demillo y Lipton presentaron un nuevo tipo de criptoanálisis teórico, basado en el differential cryptanalysis, el cual explotaba errores producidos en criptosistemas para poder recuperar su clave sobre implementaciones físicas [54]. Esta nueva técnica fue presentada como Differential Fault Analysis. Mediante este trabajo probaron que ciertos algoritmos de cifrado son vulnerables contra los ataques basados en propiedades algebraicas de aritmética modular y, por tanto, eran aplicables solo a sistemas criptográficos de clave pública como el *Rivest-Shamir-Adleman* (RSA) [55], pero no los de clave privada como el DES. Este nuevo enfoque del differential cryptanalysis abrió un amplio campo de estudio donde era posible determinar la vulnerabilidad de los sistemas, no desde

[2]El nombre y abreviatura de esta técnica es habitualmente confundida con Differential Fault Attack, que también coincide con DFA. Originalmente en muchos trabajos dicha técnica fue referenciada como Differential Fault Analysis Attack, lo que conduce a confusión. Su definición original se forma a partir de Differential Cryptanalysis que derivó en Differential Fault Analysis, término correcto para referirse a dicha técnica.

el punto de vista de su complejidad matemática como hasta ese momento se realizaba, sino por las características de su implementación física.

Desde entonces esta técnica fue tomada como referente en la bibliografía a la hora de estudiar los sistemas de cifrado, pero no fue hasta el trabajo presentado de nuevo por Biham y Shamir [22] donde ésta fue aplicada para poder analizar criptosistemas de clave privada como el DES. Concretamente, estos autores consiguieron obtener la clave secreta del sistema criptográfico DES mediante el uso de entre 50 y 200 textos cifrados, probando con ello que la seguridad de los sistemas criptográficos puede ser puesta en riesgo mediante la combinación de un modelo de fallo teórico y un análisis diferencial.

Desde estos trabajos esta técnica ha sido utilizada satisfactoriamente en numerosos algoritmos criptográficos, incluyendo sistemas de clave privada como cifradores de bloque y de flujo. Anderson y Kuhn [56] presentaron una revisión sobre ataques de bajo coste basados en el análisis de fallos para diferentes sistemas criptográficos como el RSA, DES y otros cifradores de bloques, así como para smart cards. Aplicando el DFA estos autores pudieron recuperar la clave secreta de un cifrador DES utilizando entre uno y diez textos cifrados. Gracias a la información obtenida mediante el estudio de los efectos que producía un fallo sobre la salida de los sistemas, estos autores pudieron mostrar la efectividad de estas técnicas.

En [57] se presentaron algunas técnicas generales para atacar cifradores como RC4 [58], LILI-128 [59] y SOBER-t32 [60]. Por otro lado, el DFA también es aplicado sobre cifradores de bloque, como por ejemplo el cifrador AES, en [61–63] se muestra cómo este cifrador es vulnerable frente al análisis de fallos. Concretamente en [61] el ataque es realizado de forma teórica sobre un ordenador personal que implementa este cifrador.

En el caso de los cifradores de bajo consumo de recursos o *lightweight,* como por ejemplo el cifrador PRESENT [11], en los trabajos presentados [64, 65] se demuestra de forma teórica cómo inyectando fallos en estados intermedios o en las rondas de clave es posible recuperar la clave secreta. Otros ejemplos de esta misma aplicación serían los trabajos [66–68], donde se llevan a cabo los análisis diferenciales de fallos sobre los cifradores de bloques LED [9], SIMON y SPECK [10].

Otro de los ejemplos que demuestran que esta técnica se trata de una de las más influyentes es el continuo desarrollo de ésta. Son diferentes los trabajos presentados donde el DFA es mejorado combinándolo con otras técnicas como

el *Differential Power Analysis* (DPA) [32, 69] y pasa a ser llamado *Differential Fault Intensity Analysis* (DFIA). En los trabajos [46, 47] se llevan a cabo unos análisis diferenciales donde se combinan los supuestos de inyecciones de fallos con el análisis diferencial de potencia para poder recuperar la clave de los sistemas bajo prueba. Por otro lado, es posible encontrar que la técnica del DFA también es aplicada a las funciones hash para comprometer su seguridad como el que se presenta en [70]. Estos son claros ejemplos del amplio uso y versatilidad de este método para poder estudiar la vulnerabilidad de diferentes tipos de sistemas criptográficos.

En el caso de los cifradores de flujo, esta técnica es ampliamente utilizada para estudiar su vulnerabilidad [71]. En concreto para los mencionados anteriormente del proyecto eSTREAM (Mickey, Grain y Trivium), la técnica del DFA también ha sido empleada sobre ellos. En [72, 73] se proponen una serie de ecuaciones lineales con el objetivo de, tras definir de forma teórica el modelo de inyección de fallos, permitir el acceso al estado interno del cifrador Mickey y por tanto romper el sistema obteniendo su clave secreta. En cuanto al cifrador de flujo Grain, en [74] se presentó un análisis teórico con el objetivo de determinar una serie de ecuaciones lineales que permitiesen recuperar el estado interno de este cifrador y con ello su clave. Otro buen ejemplo del uso del DFA sobre este cifrador es el trabajo presentado en [75], donde se describe un sistema teórico más real en cuanto al número de inyecciones de fallos sin restricciones de tiempo y localización en el interior de los registros del cifrador. Más recientemente se presentó en [76] un nuevo análisis diferencial para los cifradores de flujo basados en registros de desplazamientos no lineales, como son los finalistas del proyecto eSTREAM, aplicándolo al cifrador Grain.

Finalmente, para el cifrador de flujo Trivium se han presentado diversos trabajos teóricos [77–82]. En estos trabajos los autores presentan la posibilidad de recuperar la clave de este cifrador siempre que se inyecte un único fallo en su registro de estados interno. Estos trabajos se discuten en mayor detalle más adelante en la Sección 2.4. A continuación se realiza un estudio y clasificación de los tipos de trabajos más significativos reportados en la literatura donde se hace uso de ataques experimentales a circuitos que, a su vez, son combinados con análisis diferenciales, con el objetivo de poder tener una visión general acerca del uso de este tipo de técnicas de forma concreta.

2.3.4. Aplicaciones concretas de ataques sobre circuitos

Una vez presentadas las técnicas de ataque para inyectar errores en circuitos físicos y presentada la técnica de análisis diferencial basada en escenarios de ataque teórico, el siguiente paso es analizar los trabajos donde se presentan ataques prácticos sobre dispositivos criptográficos.

Los trabajos presentados en la subsección anterior tienen en común el empleo de escenarios de ataque teóricos haciendo uso de suposiciones sobre cómo deben ser los fallos insertados. En ninguno de estos estudios se utilizan ataques reales sobre implementaciones físicas para comprobar la viabilidad real de poder conseguir insertar los fallos propuestos. Para comprender mejor esta situación hay que considerar la Figura 2.9. En esta figura pueden observarse dos grandes grupos de trabajos, los llevados a cabo de forma teórica y los que son realizados de forma experimental. La gran mayoría de los trabajos reportados en la bibliografía sobre análisis de fallos, como los referenciados hasta ahora, suelen llegar hasta la determinación de la vulnerabilidad del sistema criptográfico de forma teórica, dejando en manos de terceros la comprobación de los ataques de forma experimental (Figura 2.9, Grupo A). En un segundo grupo se encuentran un pequeño conjunto de trabajos que llevan a la práctica las suposiciones teóricas presentadas en algunos trabajos teóricos, pero no llegan a montar el setup experimental completo y dan por hecho que el paso final de recuperación de la clave es inmediato. (Figura 2.9, Grupo B). Sin embargo, son muy escasos los trabajos que desarrollan el flujo completo, es decir, aquellos que implementan un setup de medida, aplican los ataques requeridos por las suposiciones teóricas, capturan experimentalmente las trazas y llevan a cabo el análisis diferencial obteniendo la clave (Figura 2.9, Grupo C).

Hay que resaltar que aunque algunos autores puedan considerarlo inmediato, hay una diferencia muy grande entre los trabajos que se encuadran en el grupo B y en el grupo C. Como ejemplo, en el caso del cifrador Trivium, un supuesto teórico indispensable para el DFA es el inyectar un solo fallo en su registro interno en un ciclo determinado. Pero además esta inyección debe ser en diferentes posiciones para el mismo ciclo de reloj en cada ataque. Esta suposición llevada a la práctica puede considerarse viable desde el momento en que el diseñador es capaz de inyectar un solo fallo en un instante determinado. Pero, esto no implica que pueda ser capaz de inyectarlo o replicarlo en sucesivos ataques sobre el número de posiciones diferentes que el análisis posterior requiere. Esto podría implicar la imposibilidad de llevar a cabo el supuesto teórico y

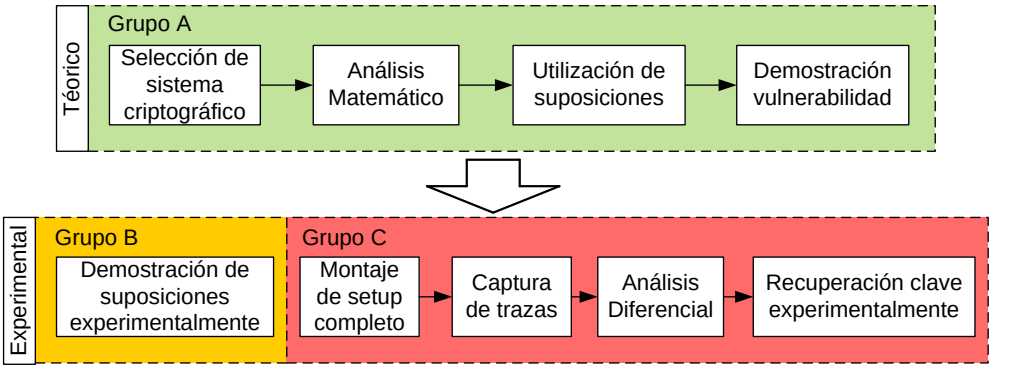

Figura 2.9.: División por grupos de los trabajos reportados, tanto teóricos como experimentales.

obligaría al diseñador a desarrollar nuevas estrategias de inserción de fallos que le permitiesen conseguir el objetivo[3]. Es por esto que, para demostrar la vulnerabilidad de la implementación de un cifrador, es necesario realizar pruebas experimentales y llevar estas pruebas hasta el punto de obtener la clave secreta a partir de los resultados experimentales. Sólo así, además de demostrar la vulnerabilidad del cifrador, podrán conocerse los puntos débiles para el ataque DFA y proponer contramedidas eficaces para este tipo de ataques.

La mayor parte de los trabajos que presentan el proceso completo de recuperación de clave mediante el uso de un setup experimental lo hacen sobre el cifrador AES. Esto es debido a que se trata de un cifrador estandarizado y ampliamente utilizado, el cual es posible encontrar implementado de serie en muchas tecnologías y su algoritmo es de sobra conocido y accesible. A continuación se describen los trabajos más representativos en cuanto a aplicación de ataques.

DFA on AES

El primero de los trabajos de criptoanálisis sobre el cifrador AES-128 es el titulado *DFA on AES* [62] presentado por Christophe Giraud. En este trabajo

[3]Este ejemplo es un adelanto del estudio realizado en el Capítulo 5 de esta Tesis, donde se estudian las variables a tener en cuenta para llevar a cabo un ataque real sobre el cifrador Trivium aplicable a un DFA.

se realizan dos modelos de fallos para hacer el análisis diferencial, mediante los cuales es posible recuperar la clave secreta del cifrador. Por un lado se encuentra un modelo a partir del cual es posible realizar el DFA siempre que se introduzca un único bit erróneo en un resultado intermedio del proceso de cifrado. Mediante este modelo pueden recuperar la clave con una media de 50 textos cifrados erróneos. Por otro lado, el segundo modelo planteado es capaz de recuperar la clave a partir de la inyección un byte erróneo en un resultado intermedio. Este ataque lo consideran más realista desde el punto de vista práctico y es capaz de recuperar la clave con una media de 250 textos cifrados erróneos.

Para llevar a cabo la inserción de fallos de forma práctica utilizan una cámara con flash modificada y un ordenador personal. El cifrador AES se encuentra implementado en una smart card, pero el código del AES implementado es conocido para para poder facilitar el ataque. El ataque lo realizan disparando el flash de la cámara sobre la superficie del chip. Al llevar a cabo el experimento, el autor encuentra el problema de no obtener "buenos" textos cifrados erróneos (no puede obtener las suposiciones teóricas de fallos fácilmente) y por tanto el experimento debe repetirse mas de 1000 veces. El autor considera que si tuviese al alcance mejores herramientas posiblemente la eficacia de estas inyecciones sería mayor.

Este trabajo representa un claro ejemplo de lo expuesto anteriormente. Se trata de un trabajo con buena base teórica, donde se asumen modelos de fallos teóricos, que una vez que se trasladan a la práctica encuentran la dificultad de la realización física. Esta dificultad lleva a la imposibilidad de conseguir el escenario planteado teóricamente en el primer modelo y muy difícilmente el del segundo. Si tenemos en cuenta el esquema anteriormente descrito Figura 2.9, este trabajo se enmarcaría en el Grupo B.

Practical Fault Attack on a Cryptographic LSI with ISO/IEC 18033-3 block ciphers

El segundo de los trabajos recopilados es el presentado por Fukunaga y Takahashi [43]. Este trabajo lleva a cabo de forma experimental ataques sobre los cifradores AES, DES, Camellia, CAST-128, SEED y MISTY1 [83–86]. Concretamente la inserción de fallos se lleva a cabo mediante la inyección de pequeños pulsos en la señal de reloj mediante un sistema de ataque. De los cifradores

atacados consiguen recuperar la clave del cifrador AES mediante un modelo teórico de DFA presentado previamente por Piret [87].

Para llevar a cabo el experimento de forma práctica utilizan una placa SASEBO-R, controlando la FPGA de control mediante un ordenador. La parte práctica de este trabajo describe cómo los autores son capaces de introducir un error en un estado intermedio de los cifradores y por tanto asumen que es posible aplicarlo a un ataque teórico. En este trabajo realmente se utiliza una parte práctica para probar la suposición de una inyección de fallo pero tras ello hacen un análisis por fuerza bruta basándose en el trabajo de Piret, concluyendo que sería altamente posible llevar a cabo el ataque completo experimentalmente.

Este trabajo, al igual que el anterior, se encuentra en el Grupo B de este tipo de estudios. Presentan un sistema de inyección de fallos y prueban que es posible conseguir dicha inserción, pero más tarde no completan el proceso y se centran en la parte teórica asumiendo el escenario del ataque teórico. Realizan por fuerza bruta el análisis basado en supuestos y no llegan a implementar el ataque completo, realizando las inyecciones de fallos necesarias para determinar la probabilidad de conseguir un texto cifrado erróneo o el número de intentos necesarios que permitirían obtener toda la información.

When clocks fail: On critical paths and clock faults

Otro de los trabajos más interesantes es el presentado por Michel Agoyan y otros [44]. Como su propio título indica, este trabajo estudia el efecto de las inyecciones de fallos mediante la manipulación de la señal de reloj, haciendo un estudio completo de los efectos que provocan dichas inyecciones. Para ello, montan un setup donde utilizan una placa FPGA como generador de fallos a través del reloj y otra donde se implementa el cifrador AES, todo ello controlado por un PC. Con este trabajo tratan de probar la parte práctica del primero de los trabajos estudiados en esta sección [62].

El análisis que llevan a cabo se basa en la posibilidad de conseguir la inyección de un solo bit erróneo en una ronda intermedia. En [62], el autor no podía conseguirlo. Por lo tanto, en este trabajo se presenta la posibilidad de llevar a cabo de forma experimental las suposiciones teóricas de un estudio anterior. Este ataque práctico concluye que es posible inyectar un solo bit erróneo pero no es sencillo y que debe ajustarse muy bien el ataque para poder conseguirlo. A pesar de esto, este trabajo constituye uno de los mejores ejemplos de

recuperación de clave desde un análisis teórico hasta el montaje de un setup experimental, perteneciendo al grupo C.

DFA of the advanced encryption standard using a single fault

En este trabajo realizado por Tunstall, Mukhopadhyay y Ali [88] se presenta un modelo teórico de la inserción de fallos y análisis diferencial. Estos autores llevan a cabo el experimento sobre una FPGA y sobre un microcontrolador, utilizando un programa en C para realizar el DFA. La inyección de fallos se realiza nuevamente mediante la manipulación de la señal de reloj y en el microprocesador utilizando una bajada en la tensión de alimentación.

En las pruebas experimentales lo que tratan de probar es la posibilidad de conseguir experimentalmente sus suposiciones teóricas del modelo de fallo. Para ello realizan experimentalmente los ataques sobre las implementaciones físicas y prueban que pueden cambiar el texto cifrado. Como comentario a los resultados experimentales, estos autores añaden que es posible cumplir su suposición teórica de inserción de fallos para uno de sus modelos, pero que la probabilidad de éxito para el segundo de sus modelos es muy baja y que por tanto sería complicado conseguirlo. Concluyen con que estos ataques son probables en unas circunstancias muy específicas, pero reconocen que no pueden saber cómo de efectivos serían estos ataques en otros microcontroladores.

En este trabajo la implementación física se lleva a cabo para probar las suposiciones teóricas pero una vez determinada su posibilidad no continúan su estudio completo. Este es otro ejemplo de cómo los autores en este campo prueban las suposiciones pero no completan el proceso dando una probabilidad estadística de inserciones o de rotura de un sistema implementado físicamente, por lo que pertenecería al grupo B.

Transient-Steady Effect Attack on Block Ciphers

Como último trabajo descrito en esta sección se encuentra el presentado por Yanting Ren y otros [45], donde se presenta un nuevo ataque denominado *Transient-Steady Effect attack* (TSE). Este ataque se basa en el fenómeno descrito por el que la salida de un circuito combinacional tiene un valor transitorio antes de cambiar a su valor correcto. Los autores generan un modelo de

fallo para este fenómeno, de forma que manipulando la señal de reloj y capturando los valores transitorios son capaces de recuperar la clave. Sin embargo, para poder aplicar este ataque, es necesario que el cifrador (en este caso el AES) transmita a la salida estos valores transitorios de las señales.

Para llevar a cabo experimentalmente el ataque utilizan una FPGA y un generador de funciones. Utilizando un PLL generan las señales de reloj necesarias para producir los fallos. El proceso de inyección de fallos lo realizan internamente, porque por los pads de entrada del dispositivo no se pueden introducir señales con una frecuencia muy alta. Una vez conseguido internamente, prueban a hacerlo de forma externa bajando la tensión de alimentación de la placa, consiguiendo con ello reducir las frecuencias a las que se introducen fallos. Con este sistema demuestran la posibilidad de implementar su modelo de fallo basado en este nuevo fenómeno y son capaces de recuperar la clave experimentalmente.

Este trabajo representa uno de los mejores ejemplos de cómo llevar a la práctica un ataque experimental considerando todas las variables hasta recuperar la clave. Este caso se encontraría en el Grupo C, según la clasificación realizada anteriormente.

Como puede observarse, el número de trabajos que implementan el setup de medida completo y recuperan la clave de un cifrador son extremandamente escasos. Además hay que tener en cuenta que todos estos trabajos son llevados a cabo sobre el cifrador AES y no sobre cifradores lightweight diseñados para ser mucho más eficientes, rápidos e implementados con unas grandes restricciones de recursos. Esto hace más difícil encontrar este tipo de estudios sobre cifradores lightweight, por lo que en este campo hay mucho trabajo por realizar. En concreto, sobre el cifrador Trivium solo se encuentran estudios teóricos de DFA, donde todos los modelos de fallos están basados en suposiciones, pero en ningún caso se ha probado su viabilidad. A continuación se estudian los principales trabajos relacionados con este cifrador.

2.4. Revisión bibliográfica del DFA sobre el cifrador Trivium

En esta sección se analizan las propuestas existentes en la literatura que realizan ataques diferenciales sobre cifradores de flujo, centrándose únicamente

en los ataques propuestos sobre el cifrador Trivium. Los artículos recopilados describen tanto el escenario de ataque como los supuestos de partida. Pero, hay que destacar que se trata de propuestas de ataques teóricos y de desarrollo matemático. En nuestro análisis no es necesario incluir los desarrollos matemáticos que los justifican, ya que el principal objetivo de esta Tesis es introducir de forma experimental fallos en implementaciones FPGA y ASIC del cifrador Trivium y utilizarlos más adelante en el sistema DFA presentado en [77], del cual por simplicidad no se mostrarán sus formulaciones matemáticas. A continuación se describen los aspectos de más interés de cada una de las propuestas. Más información se puede encontrar directamente en las referencias bibliográficas de cada uno de los artículos.

2.4.1. Differential Fault Analysis of Trivium

Se trata del primero de los artículos presentados por Hojsík y Rudolf [77], presentando el ataque DFA sobre el cifrador Trivium. En él, se introduce una técnica de ataque activo no invasivo en la que se parte del supuesto en el cual el atacante es capaz de insertar un fallo en el estado interno del cifrador, es decir, cambiar un único bit de dicho estado interno. El artículo presenta dos formulaciones diferentes para realizar el ataque, basados en la misma idea principal, siendo el segundo método el más eficiente.

Uno de los principales supuestos que se asumen en este estudio es que el atacante puede obtener una parte del flujo cifrado de salida del Trivium, denominado $z_{i(i=1)}^{\infty}$, producido por un estado interno Is_{t0} arbitrario pero fijo y el producido por un estado interno $Is_{t0}^{'}$ denominado $z_{i(i=1)}^{'\infty}$. El número de bits necesitados depende de ciertos requisitos, pero gira en torno a los 280 bits.

Por otro lado el atacante tiene que poder inyectar el fallo en repetidas ocasiones para un mismo estado interno Is_{t0} donde el valor de t_0 es fijo pero arbitrario. En el escenario contemplado el cifrador de flujo Trivium tiene que funcionar en repetidas ocasiones con el mismo vector de incialización, en inglés *Initialization Vector* (IV), y clave (K), siendo por tanto la inyección del fallo siempre sobre el mismo estado interno Is_{t0}.

Teniendo en cuenta que tanto el dispositivo cifrante como el descifrador utilizan el mismo par de IV y clave, el proceso sería aplicable a cualquiera de estos dos dispositivos.

Tabla 2.4.: Número de inyecciones de fallos necesitados (m) para obtener un número de bits del estado interno (T).

T	20	40	60	80	100	120	140	180	220	240	280	288
m	10.1	20.3	28.4	35.4	39.8	41.9	42.4	42.7	42.9	43.0	43.1	43.1

El resultado del ataque es la determinación del estado interno, el cual puede ser utilizado para obtener la clave y el IV. Esto es posible debido a la reversibilidad del algoritmo del cifrador Trivium y al hecho de que, al inicializar el cifrador, gran parte de los bits del estado interno son puestos a valores fijos y conocidos. Tras determinar Is_{t0}, se podría hacer funcionar el cifrador Trivium de forma inversa.

Para resumir lo expuesto anteriormente, se puede decir que se asumen los siguientes requisitos para poder llevar a cabo los ataques:

Siendo t_0 arbitrario pero fijo entero positivo e Is_{t0} el estado interno de Trivium arbitrario pero fijo en el instante t_0. Se considera que un atacante:

- Es capaz de obtener los n primeros bits consecutivos del flujo cifrado z_i producidos por el estado interno Is_{t0}.
- Es capaz de inyectar un fallo en una posición aleatoria del estado interno Is_{t0}.
- Es capaz de repetir la inyección del fallo en una posición aleatoria de Is_{t0}, m veces.
- Es capaz de obtener los n primeros bits consecutivos del flujo cifrado $z_i^{'}$ producidos por el estado interno $Is_{t0}^{'}$ para todas las inyecciones de fallo.

El número de bits del flujo correcto y erróneo necesitados, n, difiere para las dos formulaciones del ataque. Para la primera de de ellas es de $n = 160$ y para la segunda $n = 280$. En la Tabla 2.4 se muestra el número de bits del estado interno recuperados (T) en función del número de inyecciones de fallos (m). Este número fue calculado por los autores como la media de 1000 ataques.

Se puede observar que para una media de 43.1 inyecciones de fallos se puede recuperar el estado interno por completo, rompiendo así el algoritmo del cifrador Trivium.

2.4.2. Floating Fault Analysis of Trivium

Este artículo, presentado también por Hojsík y Rudolf [78], introduce una mejora esencial en el ataque DFA de su primer artículo, obteniendo unos mejores resultados en el número de inyecciones necesitadas para romper el cifrador Trivium. En el primer artículo, el DFA necesitaba una media de 43.1 inyecciones de fallos para obtener todos los bits del estado interno, mientras que en este artículo la modificación de la formulación matemática reduce el número de inserciones de fallos necesarias para obtener todos los bits del estado interno a una media de 3.2 (este dato está obtenido tras repetir el experimento 10000 veces). En el mejor de los casos se puede obtener la clave con solo 2 inyecciones.

La idea principal del ataque es una simple transformación de las ecuaciones polinómicas, obtenidas durante el ataque, en ecuaciones lineales. Este procedimiento se basa en lo que denominan *"floating description of Trivium"* y en algunas de las propiedades de propagación de fallos en la evolución del estado interno.

Por otro lado, se siguen asumiendo los prerrequisitos del DFA exceptuando el número de bits de flujo cifrado correcto y erróneo necesitados, siendo en esta ocasión $n = 800$. Con este número de bits los resultados obtenidos tras 10000 experimentos son de un 2 % de probabilidad de acierto para $m = 2$ (donde m son las inyecciones de fallos), 78.5 % para $m = 3{,}99$, 8 % para $m = 4$ y para $m = 5$ el ataque siempre revela la clave.

2.4.3. Fault Analysis of Trivium

En este artículo de Yupu, Juntao, Qing y Yiwei [79] parten del artículo de Hojsík y Rudolf [78] anteriormente descrito y del que mantienen algunos de sus supuestos, pero presentan algunas novedades. Para empezar tienen en cuenta los siguientes supuestos establecidos por Hojsík y Rudolf:

- La necesidad de inyectar un fallo en el estado interno en un tiempo fijado.
- Tras cada inyección de fallo, puede cambiar únicamente un 1 bit del estado interno.

Sin embargo para este trabajo, los autores consideran que para cualquier cifrador de flujo, el estado interno se actualiza muy deprisa, haciendo difícil que

un atacante tome el estado en un tiempo fijado. Por otro lado, asumen que los ataques se deberían realizar por fuerza bruta y que cuando cambia un bit es difícil no dañar los vecinos de este bit. A partir de aquí se asumen los siguientes supuestos:

1. Pueden realizar una inyección en el estado interno en un instante de tiempo aleatorio.
2. Tras cada inyección, la posición del fallo es aleatoria dentro uno de los tres registros y se encuentra en un área aleatoria dentro de un conjunto de 8 bits.

En un instante aleatorio M durante el proceso del generación de flujo cifrado, el atacante inyecta fallos en algunos bits del estado interno (esto es, en el instante M cambia algunos bits del estado interno). El conjunto de bits erróneos se denota por A. Tras la inyección, el atacante no conoce ninguno de los dos, es decir, M y A, sólo puede conocer que:

- El conjunto de fallos A pertenece a alguno de los tres registros, es decir, se encuentra dentro de uno de los tres registros pero no puede saber su posición.
- El bit erróneo se encontrará en A y puede ser un bit dentro de un conjunto de 8 bits.

Por otro lado, el escenario contemplado se basa en un atacante que consigue un sistema de encriptación equipado con un cifrador Trivium. Este lo hace funcionar y obtiene las trazas originales del flujo cifrado, reinicia el sistema y realiza la inyección repetidamente bajo los supuestos 1 y 2. Así puede obtener la traza de flujo cifrado con error inyectado y por tanto establecer el análisis diferencial entre flujo correcto y erróneo. El proceso podría repetirse cuantas veces fuese necesario y de esta forma romper el cifrador Trivium con la información obtenida. Por último, consideran que el IV siempre será el mismo y por tanto deciden obviarlo en sus cálculos. Como resultado se presenta que por cada inyección de fallo se necesita un promedio de 195 bits de flujo cifrado para realizar el análisis diferencial y con ello poder romper el cifrador Trivium.

2.4.4. Otras aportaciones de análisis diferenciales

Además de los trabajos descritos anteriormente, en la literatura se encuentran otros trabajos, entre los cuales se puede destacar [80] donde se presenta un

Improved Differential Fault Analysis of Trivium, mediante el cual los autores presentan una mejora del trabajo de Hojsík y Rudolf [77]. En este artículo no se especifican los requisitos del ataque pero los datos necesitados para recuperar el registro de estados en un instante $t = 0$ son los siguientes:

1. Conocer la posición de los fallos insertados m veces.
2. Tener acceso al flujo cifrado generado sin fallos inyectados.
3. Tener acceso al flujo cifrado generado tras inyectar un fallo.

Los resultados presentados en este artículo ofrecen la posibilidad de obtener la clave y el IV del cifrador Trivium con dos inserciones de fallo, utilizando 420 bits del flujo cifrado correcto y erróneo capturados.

Por otro lado, en [82] se presenta un nuevo tipo de análisis denominado *Mutant Differential Fault Analysis* (MDFA). Este trabajo, que también tiene como punto de partida el trabajo [77], presenta una mejora sobre el original, puesto que sus autores son capaces de obtener la clave e IV con una sola inserción de fallo. Esta inserción debe producirse en una posición aleatoria y desconocida del estado interno en la fase de funcionamiento. Para insertar fallos realizan muchos reinicios del sistema y en la fase *offline* se analiza la información mediante diferentes métodos aprovechando las ventajas de cada uno de los métodos utilizados para el MDFA. Este proceso permite obtener la clave con una sola inserción.

Estos autores distinguen dos instantes de tiempo $t = 0$ y t_0:

1. $t = 0$: instante incial de carga de vector y clave, no ha habido desplazamientos aún.
2. t_0: instante en el que el cifrador comienza a generar el flujo cifrado.

La información que el atacante obtiene de la fase de funcionamiento tras las inserciones es:

1. El flujo cifrado correcto.
2. El flujo cifrado incorrecto.
3. Índices de inserciones de fallo k obtenidos del registro interno en t_0.

Mediante la resolución del sistema de ecuaciones se obtiene el estado interno en el instante $t \geq t_0$ y a partir de aquí volviendo hacia atrás se obtiene la clave. Utilizan para resolver el sistema *SAT solver Minisat2*. Consideran que su

ataque podría aplicarse además a los otros cifradores del proyecto eSTREAM, MICKEY y Grain.

Por último, otra aportación significativa se encuentra en la referencia [81], el cual es titulado *Improved Multi-Bit Differential Fault Analysis of Trivium*. En este trabajo se mejoran las restricciones del sistema presentado en [79]. En este artículo se establece que para cuatro inserciones siempre se obtiene la clave. El método que presentan permite atacar el sistema mediante diferentes modelos de fallo multi-bit, con fallos en cualquier ciclo aleatorio y desconocido de la generación del flujo cifrado. Para resolver los sistemas de ecuaciones utilizan *SAT solver Cryptominisat 2.9.6*. Por último, eliminan la restricción en la que el fallo no debe inducirse en diferentes registros.

2.4.5. Resumen de requisitos para realizar ataques teóricos

Con todo lo expuesto anteriormente se van a recopilar las condiciones que deben cumplir los ataques desde el punto de vista teórico con la idea de poder establecer los cimientos de uno o varios ataques que permitan estudiar la vulnerabilidad del cifrador Trivium.

Los ataques que se presentan se producen sobre el estado interno del cifrador, es decir Is, el cual está formado por 288 bits. Esto quiere decir que no se producen ataques hasta que no se ha generado Is. Por otro lado, los ataques siempre se producen en un mismo instante inicial t_0 (tras obtener Is, teniendo por tanto Is_{t0}), siendo este establecido por el atacante en el momento oportuno. Los prerrequisitos para los ataques son los siguientes:

- Se es capaz de obtener los n primeros bits consecutivos del flujo cifrado z_i producidos por el estado interno Is_{t0}.
- Se es capaz de inyectar un fallo en una posición aleatoria del estado interno Is_{t0}.
- Se puede obtener la inyección del fallo en posiciones diferentes de Is_{t0} m veces.
- Se es capaz de obtener los n primeros bits consecutivos del flujo cifrado $z_i^{'}$ producidos por el estado interno $Is_{t0}^{'}$ para cada una de las inyecciones de fallo.

Por lo tanto, el proceso de ataque descrito consiste en:

1. Inyectar el fallo en Is en un instante determinado y obtener los bits necesarios.
2. Determinar la posición del fallo mediante los sistemas de ecuaciones.
3. Repetir el proceso las veces que sean necesarias, siempre con el mismo Is, es decir, mismo IV y clave y en el mismo instante de tiempo.
4. Analizar la diferencia entre el flujo cifrado original y el erróneo.
5. Obtener Is en el instante inicial y con éste la clave secreta y el IV.

Con todo lo explicado anteriormente, el principal problema para realizar este ataque de forma experimental es encontrar un mecanismo de inyección de fallos que inserte un único fallo en el estado interno del cifrador y, que al repetir la inyección de fallos, sea capaz de producir los fallos en posiciones diferentes del estado interno del cifrador en un mismo instante de tiempo.

2.5. Técnicas para insertar fallos experimentalmente sobre el cifrador Trivium

Una vez analizados la estructura y el funcionamiento del cifrador Trivium, así como los ataques activos no invasivos y los estudios teóricos sobre ataques a este sistema, se encuentra la necesidad de diseñar un sistema para la inserción y análisis de fallos en implementaciones de este cifrador tanto en FPGAs como en ASIC. Este sistema debe permitir inyectar los fallos necesarios para poder llevar a cabo un DFA y así probar su vulnerabilidad. En este trabajo de Tesis se han diseñado dos mecanismos para la inyección de fallos, uno de ellos consiste en un sistema basado en la manipulación de la señal de reloj mediante inserción de pequeños pulsos y el otro en la manipulación de la señal de control del sistema.

2.5.1. Inserción de fallos mediante pulsos de reloj

El primer mecanismo para la inserción de fallos en el cifrador consiste en la introducción de pequeños pulsos en la señal de reloj. Esta técnica ha sido descrita anteriormente, por ejemplo en [43–45] donde se realizan los DFA y

se muestran resultados prácticos sobre el cifrador de bloques AES. Teniendo en cuenta los modelos teóricos de vulnerabilidad del cifrador Trivium frente a ataques, el mecanismo de inserción de fallos desarrollado debe de ser capaz de inyectar un solo fallo efectivo en el registro interno del cifrador, en un ciclo determinado.

La introducción de pulsos suficientemente pequeños en la señal de reloj provoca violaciones de los tiempos de setup o de hold de los biestables, lo que podrá producir un fallo en el bit que debe almacenar. Esto será más probable cuanto mayor sea el retraso del camino combinacional en su entrada de excitación. La consecuencia del fallo es que se puede traducir en un error en el dato cargado en el estado interno del cifrador.

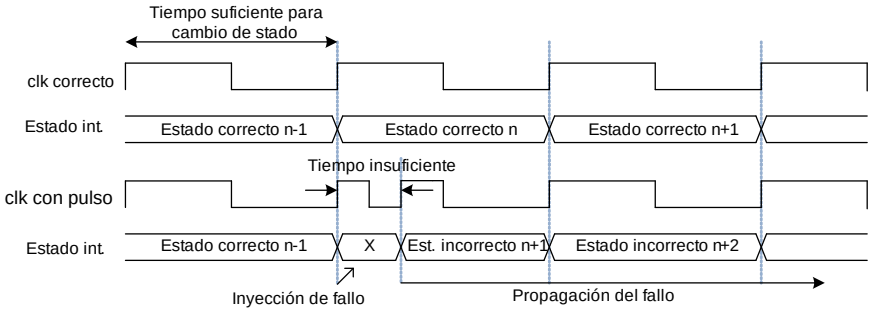

Figura 2.10.: Esquema de inducción de un pulso de reloj y error.

De forma gráfica, en la Figura 2.10 se muestra la forma en la que debe alterarse la señal de reloj y el efecto que tiene en el funcionamiento del circuito. Cuando el tiempo entre dos flancos activos de reloj es el adecuado, el estado interno del cifrador se va actualizando correctamente, pero cuando el tiempo se reduce lo suficiente, el estado interno no se actualiza de forma correcta. Este estado incorrecto se sigue transmitiendo en los sucesivos ciclos de reloj produciendo un flujo cifrado incorrecto.

2.5.2. Inserción de fallos mediante manipulación de la señal de control

El segundo de los mecanismos que se va a utilizar para la inyección de fallos en el cifrador consiste en el desplazamiento de las señales de control del ci-

frador respecto al flanco activo del reloj. Este mecanismo no ha sido descrito anteriormente debido a que no es un mecanismo habitual para la inserción de fallos. Ha sido propuesto en los trabajos de esta Tesis como un mecanismo específico para el cifrador Trivium.

Como puede apreciarse en la Figura 2.11 la inyección del fallo se produce cuando el cambio de la señal de control se aproxima lo suficiente al flanco activo del reloj. En esta situación se producirán violaciones en los tiempos de setup de los biestables que forman el registro interno.

Este mecanismo aprovecha el hecho de que las señales de control de este cifrador pueden no considerarse críticas y, en consecuencia, el diseño no balancea sus retrasos ni las distribuye por líneas especiales, como ocurre en el caso de la señal de reloj del sistema. Por lo tanto, tras activar la señal de control para que se produzca una parada del cifrador y tras volver a activar dicha señal para que continúe el funcionamiento del cifrador, puede forzarse la posibilidad de que haya biestables que, debido al cambio tan próximo de la señal de control al flanco activo de reloj, no vean el cambio antes del flanco y otros, sí. Por lo tanto, puede producirse un almacenamiento erróneo de los datos. Esto provocaría que en el nuevo ciclo de reloj se encuentren con un valor previo y por tanto se haya insertado un fallo.

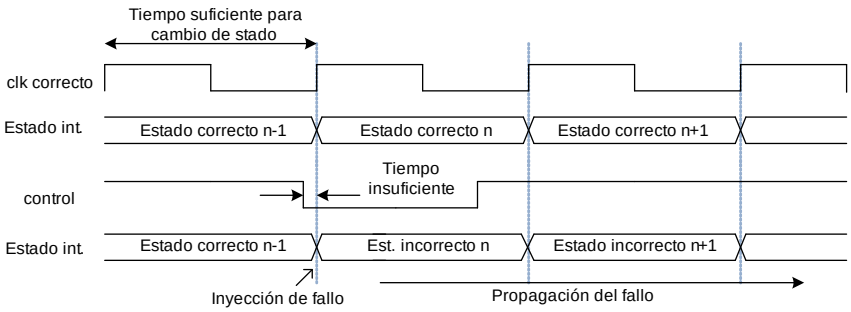

Figura 2.11.: Esquema de inducción de error mediante manipulación de la señal de control.

2.6. Conclusiones

En este capítulo se ha llevado a cabo una descripción detallada del estado del arte de la criptografía, los cifradores y el criptoanálisis. Para ello se ha realizado una presentación general de la criptografía como marco para poder introducir el ámbito principal de esta Tesis Doctoral, es decir, el análisis de vulnerabilidad de los cifradores de flujo. En concreto, se ha descrito el principal sistema criptográfico objeto de estudio de esta Tesis, el cifrador de flujo Trivium.

Una vez descrita la elección del sistema criptográfico, se ha podido ver que, debido al desarrollo matemático de los algoritmos y la longitud de las claves secretas, atacar un dispositivo criptográfico por su vulnerabilidad matemática es un proceso obsoleto. Este hecho ha llevado a estudiar las implementaciones físicas de los propios algoritmos matemáticos para poder determinar sus vulnerabilidades. Para poder estudiar estas vulnerabilidades, se ha considerado una de las técnicas más influyentes en este ámbito, es decir, el análisis diferencial de fallos o DFA.

Como se ha podido ver a lo largo del capítulo, los trabajos presentados en la literatura sobre el DFA son en su gran mayoría estudios teóricos basados en modelos de fallos teóricos (incluidos los relativos al cifrador Trivium), siendo muy escasos aquellos que realizan un ataque completo de recuperación de clave experimentalmente. Hasta donde sabemos, en el caso del cifrador Trivium no existen trabajos en la literatura que realicen ataques experimentales a su implementación.

Es por esto que se encuentra la necesidad de estudiar las técnicas prácticas de inserción de fallos para poder diseñar un sistema de ataque que permita llevar a la práctica un escenario de ataque teórico sobre el cifrador Trivium. Con ello es posible demostrar la viabilidad de los estudios previos y determinar la vulnerabilidad de este tipo de cifrador. Para ello, tras estudiar las técnicas de ataques más comunes sobre circuitos para conseguir la inserción de fallos, se ha determinado que el conjunto de técnicas más interesantes para nuestro objetivo son las que engloban los denominados ataques activos no invasivos. Esta selección se lleva a cabo debido a que son las técnicas más eficientes a la hora de poder inyectar fallos en los cifradores, son válidas para implementar un sistema de ataque válido tanto para FPGA como ASIC y su coste en recursos es muy inferior en comparación con los demás métodos.

A partir de aquí, se han estudiado los trabajos reportados donde se aplica el DFA sobre el cifrador Trivium y se han podido determinar las principales características de los escenarios de ataque teóricos con el fin de poderlos trasladar a la práctica. De estas características, la más importante es la de inyectar un solo error efectivo en el interior del cifrador. Esta inserción debe realizarse repetidamente para un mismo instante de tiempo en pruebas sucesivas, teniendo que capturar el flujo cifrado correcto y erróneo a la salida del sistema.

Considerando todo el estudio realizado, para poder conseguir el objetivo de inserciones de fallos en el cifrador Trivium, se han elegido dos técnicas de inyección de fallos. Por un lado se encuentra la técnica mediante la cual se manipula la señal de reloj del sistema con pequeños pulsos de reloj, y por otro lado, se encuentra la técnica mediante la cual es posible manipular la señal de control del sistema para provocar un funcionamiento anómalo.

3. Inserción de fallos en implementaciones FPGA de cifradores Trivium

3.1. Introducción

En este capítulo se presenta el análisis de vulnerabilidad del cifrador Trivium implementado en FPGA frente a ataques activos no invasivos mediante la manipulación de la señal de reloj. A lo largo de este capítulo se realizará el estudio de la vulnerabilidad de varios diseños del cifrador Trivium frente a ataques: el cifrador Trivium con carga de clave e IV en paralelo, con carga en serie y un modelo del cifrador de consumo de baja potencia. De estos diseños, el de carga en serie ha sido diseñado a lo largo de la realización de la presente Tesis Doctoral, así como sus variantes. Por otro lado, para poder estudiar las vulnerabilidades se presenta un sistema de ataque donde se realizan inducciones de pulsos cortos en la señal de reloj del sistema en diferentes ciclos de reloj y para diferentes configuraciones.

Gracias a este análisis de vulnerabilidades es posible establecer los puntos débiles del cifrador Trivium implementado en FPGA, los cuales pueden servir también para el estudio de vulnerabilidad sobre tecnología ASIC y por consiguiente para el diseño de contramedidas frente a este tipo de ataques. Para llevar a cabo estos análisis se han utilizado dos familias de FPGA y así poder obtener conclusiones generales acerca del comportamiento del sistema. Además, para poder ver si el rutado y colocación de los componentes afecta en la vulnerabilidad, en la FPGA se incluyen dos cifradores que funcionan en paralelo y que son sometidos a las mismas pruebas bajo las mismas condiciones. Así mismo, el trabajo descrito en este capítulo se trata de una continuación

de los Trabajos Final de Grado y de Máster [89, 90], donde se presentaban resultados preliminares de análisis de vulnerabilidad de los cifradores y diseños preliminares de los sistemas de ataque. Siendo a lo largo del desarrollo de la presente Tesis Doctoral donde se ha mejorado y profundizado en el desarrollo de los mismos para obtener los resultados presentados.

El contenido de este capítulo no se muestra porque se encuentra publicado en C.V. 1.

3.2. Conclusiones

A largo de este capítulo se ha presentado el estudio de las características de los diferentes diseños del cifrador Trivium implementado en FPGA, así como un estudio de los métodos que permiten implementar un ataque activo no invasivo que hace uso de la manipulación de la señal de reloj. Tras ello, se ha presentado el diseño de un sistema de ataque a través del cual es posible inyectar fallos en los registros internos de los cifradores y el cual es posible utilizar en diferentes familias de FPGA.

Los resultados obtenidos tras realizar la implementación del sistema de ataque lateral activo al cifrador Trivium, tanto con el diseño de carga en paralelo, como los de carga serie y baja potencia, permiten concluir que se produce una diferencia en los comportamientos de un diseño frente al otro, tanto en la misma, como en diferentes familias de FPGA, cuando se llevan a cabo los ataques con el sistema diseñado.

El conjunto de resultados no sólo demuestra que este cifrador en sus diferentes versiones es vulnerable frente a ataques por modificación de la señal de reloj, sino que sus vulnerabilidades dependen de sus puntos más críticos, los cuales son los biestables realimentados o utilizados para las operaciones lógicas, así como las celdas vecinas.

El sistema de ataque diseñado ha permitido la inyección de errores efectivos en todos los cifradores en diferentes ciclos de inserción, para todas las diferentes claves secretas. Esto demuestra la independencia del sistema de ataque de la selección de clave y vector de inicialización, puesto que la capacidad y eficiencia del sistema han demostrado ser elevadas en todos los casos.

En el caso del cifrador de baja potencia, se ha podido ver que la capacidad y eficiencia del sistema es menor debido a la menor capacidad de inserción

de fallos efectivos. Igualmente se ha conseguido la inyección de fallos efectivos en sus registros y por tanto se ha demostrado su vulnerabilidad al igual que ocurre con los otros diseños.

Por último, se ha podido demostrar que con unas herramientas básicas y con un coste prácticamente nulo, se puede comprometer la seguridad del cifrador Trivium en sus diferentes versiones. Mientras que hasta donde se ha podido ver, la versión del cifrador con carga serie y con contador externo presenta una mayor resistencia a los ataques, ya que se trata del cifrador cuya frecuencia máxima de operación es la mayor entre los diferentes diseños del cifrador Trivium.

4. Inserción de fallos en implementaciones ASIC de cifradores Trivium

En este capítulo se presenta el estudio y desarrollo de los ataques realizados sobre diferentes cifradores de flujo Trivium implementados en tecnología ASIC. Al igual que en las implementaciones FPGA, en ASIC, los cifradores Trivium también operan a frecuencias muy elevadas y por tanto comparten los problemas a la hora de generar los pulsos de reloj que permiten la inducción de fallos. Por otra parte, al contrario que en FPGA, en ASIC no es posible montar el sistema de ataques de forma interna en el propio chip donde se encuentran los cifradores Trivium y debido a esto es necesario montar un sistema de ataque externo a la implementación. El sistema de ataque, pues, deberá generar señales muy rápidas que se puedan inyectar en el ASIC y hacerlo de forma muy precisa para que pueda inyectarse un fallo en el cifrador Trivium. Esta Tesis se ha desarrollado en el Instituto de Microelectrónica de Sevilla, donde se cuenta con el equipo de test *Agilent 93000*, que aunque no está diseñado para el propósito de realizar ataques, se llevaron a cabo diferentes configuraciones y test que permitieron su utilización para tal efecto y con el cual se ha desarrollado nuestro sistema de ataque sobre ASIC.

En este capítulo se presenta la descripción del ASIC donde se encuentran los diferentes cifradores de flujo Trivium, la configuración del equipo de test, así como la creación de un detallado plan de test para llevar a cabo los ataques. Por último se presentan los resultados obtenidos tras la aplicación del sistema de ataque por inducción de pulsos en la señal de reloj y manipulación de la señal de control, cerrando el capítulo con las conclusiones obtenidas a partir de dichos resultados.

4.1. Descripción de la implementación ASIC

A la hora de llevar a cabo las pruebas de inserción de fallos sobre los cifradores Trivium se utilizó un circuito en tecnología ASIC diseñado en el marco del proyecto CITIES[1]anterior al inicio de este trabajo de investigación. Este ASIC contiene internamente tres grupos de circuitos, uno de los cuales es el utilizado para la realización de ataques por inserción de fallos.

El grupo de circuitos que concierne a la realización de los ataques por inserción de fallos consta de las siguientes caracerísticas:

- Contiene ocho diseños diferentes del cifrador Trivium. Estos diseños son la versión estándar del cifrador y las versiones de generación de flujo cifrado multi-bit de dos, ocho y dieciséis bits y versiones de baja potencia también para Triviums con salidas de uno, dos, ocho y dieciséis bits. A diferencia de la versión estándar, los modelos multi-bit son versiones del cifrador capaces de generar un flujo cifrado de dos, ocho y dieciséis bits en cada ciclo de reloj.

- El mecanismo de carga de clave y vector de inicialización sobre el ASIC se realiza de forma serie. Los datos transferidos desde el exterior en modo serie se almacenan en dos registros de 80 bits, uno para la clave y otro para el vector de inicialización. Posteriormente el contenido de estos registros es cargado en paralelo en el registro de estado de todos los Triviums.

- Implementación de dos cifradores idénticos por cada modelo. Con el objetivo de poder estudiar el comportamiento de los cifradores y su dependencia del rutado frente a un mismo ataque y condiciones de ataque, se encuentran implementados dos cifradores del mismo modelo por cada diseño del cifrador Trivium enumerado anteriormente.

- Señales de control compartidas. Las señales de control utilizadas son comunes a todos los cifradores implementados en el circuito, es decir,

[1]Este proyecto es anterior a la realización del presente trabajo de investigación y por tanto queda fuera del ámbito de esta memoria. Aún así, la utilización del circuito para la obtención de resultados hace necesaria una breve descripción de su implementación y características.

cuando se inicializan todos, se inicializan a la vez, cuando funcionan todos, funcionan a la vez y cuando se paran, todos los cifradores se paran.

- Multiplexado de las salidas. Dado que todos los cifradores funcionan con las mismas señales de entrada, para poder ver las señales de cada cifrador a la salida del circuito, se utiliza una multiplexación que permite seleccionar entre las salidas de un cifrador u otro.

- Muestreo del estado interno de cada cifrador. Para poder comprobar los resultados de los ataques de inserción de fallos sobre los cifradores, se tiene acceso al registro de estados de cada uno de los cifradores. Este acceso se consigue muestreando el registro interno de un cifrador Trivium en un registro intermedio de 288 bits y sacando de forma serie el contenido del registro intermedio por una de las salidas del ASIC. Para poder muestrear los registros de estados internos se debe de tener en cuenta lo siguiente:

  1. Se debe seleccionar la versión deseada del cifrador a muestrear.
  2. El muestreo de las dos copias en paralelo del modelo del cifrador se realiza con un reloj "clk2" independiente del general.
  3. La salida de datos de los dos registros anteriores se realiza de forma serie por dos pines de salida.

En la Figura 4.1 puede observarse una representación de una parte del circuito ASIC que contiene a dos cifradores iguales. Los cifradores funcionan en paralelo con las señales generales de entrada de reloj y control, mientras que la salida serie de los registros intermedios lo hacen con la señal clk2. Como puede verse, el registro de estados de cada cifrador se copia en dos registros intermedios comunes para todos los cifradores y éstos son muestreados con el reloj clk2. Este diseño permite por tanto tener acceso al registro de estado del cifrador que se desee mediante el uso de una señal de selección.

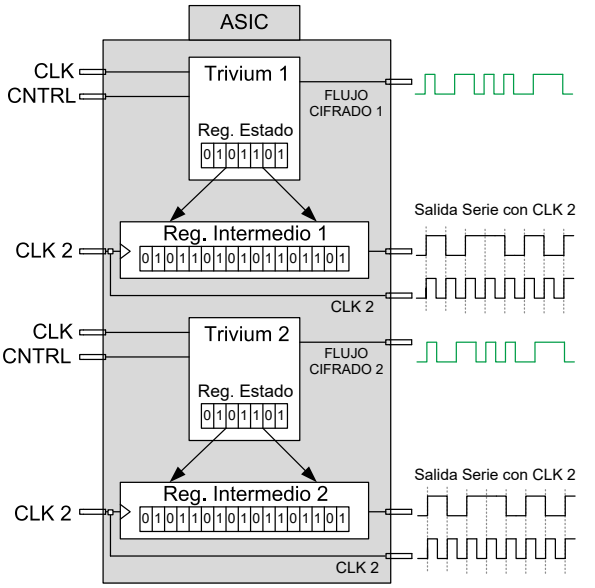

Figura 4.1.: Representación de los cifradores en paralelo y sus salidas del circuito ASIC.

4.2. Equipo de test Agilent 93000

Para llevar a cabo los test y pruebas sobre la implementación ASIC se ha utilizado el sistema de test automático de señal mixta *Agilent 93000 SOC C200e*. Este equipo de test abarca un amplio abanico de aplicaciones y permite realizar las pruebas necesarias para el testado de diferentes circuitos digitales, analógicos y mixtos. En la Figura 4.2 puede apreciarse una imagen de dicho equipo de test, el cual se ha utilizado para la realización de las pruebas y ataques sobre las implementaciones. Este equipo está formado por diferentes componentes, los cuales son: una cabeza de test, un manipulador, un armario de soporte de equipo interno, una unidad de refrigeración y la estación de trabajo de HP. En el caso de la cabeza de test, se trata del componente necesario para conectar la placa de prueba y contiene los módulos digitales/analógicos de test que utilizan las entradas/salidas y alimentaciones. Por otro lado, el manipulador permite seleccionar la posición de la cabeza de test, mientras que el armario de soporte contiene la instalación de alimentaciones y conexiones de refrigeración

de agua. A su vez, las unidades de refrigeración proporcionan, como su propio nombre indica, la refrigeración al equipo de test por circulación de agua. Por último, se encuentra la estación de trabajo que contiene el software de control del equipo denominado *SmarTest*, el cual funciona bajo el sistema operativo HP-UX.

Figura 4.2.: Equipo de test Agilent 93000.

Aunque este equipo de test no está diseñado específicamente para la inserción de fallos en circuitos, su uso es posible para generar unos patrones que hagan fallar al circuito y para automatizar las pruebas a realizar. Además, otro punto a tener en cuenta es que el equipo ofrece una gran precisión en cuanto a la generación de señales, lo cual es fundamental a la hora de insertar fallos, como por ejemplo inducir pulsos en las señales de reloj o desplazar flancos de funcionamiento. Esta posibilidad de generación de señales con gran precisión permite determinar rangos de funcionamiento. Esta característica es la que nos resulta más útil en la inyección de fallos dado que, como se pudo ver anteriormente, los cifradores son capaces de funcionar a altas frecuencias, por lo que es necesario tener un gran control para evitar incurrir en errores de medida y falsos positivos.

Para llevar a cabo las pruebas en este equipo es necesaria la utilización de una placa específica, previamente diseñada y fabricada, denominada *Device Interface Board* (DIB) que sirve de interfaz entre el equipo y el circuito ASIC que contiene los cifradores.

Características que ofrece el equipo

Este equipo de test posee diferentes características relacionadas con la frecuencia máxima de operación, configuración de canales digitales para diferentes propósitos, determinación de vectores de test, generación automática de vectores, temporización de señales, almacenamiento de resultados en memoria y posibilidad de generación y visualización de resultados esperados y recibidos. De las posibilidades que ofrece este equipo, las características más relevantes para los test que se van a realizar se encuentras las siguientes:

- **Temporización de pines de forma independiente.** Se puede configurar la temporización de cada uno de los pines de entrada del sistema de forma independiente. De aquí que se pueden definir señales cuyos retrasos y flancos son completamente independientes unos de otros. Incluso, como veremos más adelante, permite cambiar la temporización de un mismo pin en diferentes posiciones del vector de test.
- **Posibilidad de cambiar los niveles de tensión.** Mediante la configuración del sistema es posible establecer diferentes niveles de tensión a la hora de realizar las pruebas, permitiendo incluso la variación de dichas tensiones de forma dinámica durante un proceso de test.
- **Memoria de datos de hasta 7 megabytes por canal.** El equipo posee una memoria de 7 MB para la definición de los vectores de test de cada pin o canal digital y una memoria por pin para instrucciones y configuración de 2 MB.
- **Visualización de forma de onda.** Con ayuda del software SmarTest es posible visualizar las diferentes formas de ondas generadas así como las obtenidas tras un test.
- **Automatización.** Es posible realizar test de forma automática tanto desde el software del equipo como utilizando un programa en lenguaje C que permita realizar configuraciones, test y captura de datos de forma automática.
- **Incluye una serie de test ya definidos como:** Test funcionales, test paso/fallo, Golden device, Smooh plot o test de frecuencias entre otros.
- **Inclusión de patrones esperados para test de paso/fallo.** El equipo permite utilizar patrones generados de forma externa con herramientas de simulación donde se establecen los valores esperados en las salidas

del circuito y que son utilizados para realizar pruebas de paso/fallo. Estos test permiten determinar si el comportamiento del circuito se corresponde con el diseñado o no.

De las características que posee este equipo de test, una de las más significativas para llevar a cabo inserciones de fallos es la frecuencia máxima de reloj, la cual está establecida en 200 MHz. Aunque esta es la máxima frecuencia del equipo, es posible generar pulsos con una separación inferior a 50 ns, los cuales serán utilizados para la inducción de errores. La definición de estas formas de onda se comenta más adelante a lo largo de este capítulo.

En cuanto a la configuración de canales, se pueden aplicar y capturar señales digitales sobre el dispositivo, sincronizar entre módulos digitales y analógicos y realizar test funcionales entre otros. Por otro lado, hay que definir los vectores de datos de test y la temporización de las señales, los cuales determinan la forma de onda de la señal digital de test. En este equipo la forma temporal de una señal se genera mediante la definición de las llamadas ventanas de temporización. En estas ventanas se definen los instantes en los que la señal puede cambiar, permitiéndose 8 flancos para la definición de valores y 6 flancos para la captura de valores lógicos. Mediante la definición de ventanas de temporización con cambios en tiempos diferentes puede conseguirse la introducción de pulsos pequeños en la señal de reloj. Este proceso de definición de señal mediante los diferentes flancos disponibles es de vital importancia para la generación de pulsos de reloj y se explica en mayor detalle más adelante. Por otro lado, los patrones de test a los que se aplica esta temporización pueden ser generados automáticamente con cierta periodicidad o ser importados desde herramientas de diseño externas mediante la ayuda del propio equipo.

Respecto a la modificación de los niveles de tensión de polarización del circuito, es posible hacerlo mediante las configuraciones del test. El equipo permite definir separadamente los valores de tensión y realizar los test necesarios para diferentes valores de tensión del core y del anillo de pads. Esto permite alimentar al anillo de pads del circuito a la tensión recomendada para el correcto funcionamiento del sistema y a su vez modificar la tensión del core haciendo bajar o subir su valor con el objetivo de llevarlo a los límites de funcionamiento asegurado por el fabricante. Esta modificación de valores se puede realizar mediante una configuración previa o de forma dinámica, es decir, se puede lanzar un proceso de test en el cual la tensión de alimentación se modifique en función de los test que se estén realizando. Esta característica nos ha permitido

obtener los resultados relacionados con variaciones de tensión y comprobar el comportamiento del circuito frente a ataques por señal de reloj en combinación con la variación de tensión.

En relación al proceso de test, las pruebas realizadas se han basado en test de paso/fallo. Este tipo de test permite realizar pruebas donde, a partir de una serie de patrones previamente establecidos, se comprueba si tras la realización de la prueba los resultados se corresponden con el patrón esperado o no. En caso de corresponder el resultado con los patrones esperados, el resultado será de paso, mientras que en el caso contrario se tendrá un resultado de fallo. Por último, el equipo permite el almacenamiento de los resultados y su visualización y comparación mediante los datos esperados y recibidos. Esta última opción es muy útil en el caso de la inserción de fallos dado que permite visualizar cómodamente si durante la prueba realizada se han inyectado errores o si por el contrario la prueba ha sido insatisfactoria y debe repetirse.

Configuración del equipo de test

Para poder llevar a cabo las pruebas, es necesario configurar del equipo, manipular datos, realizar test y volcar a fichero de resultados entre otras operaciones. Todas estas operaciones son realizadas mediante el entorno software de configuración modular denominado *SmarTest*, el cual es propio del equipo Agilent y puede ser completado por scripts de código C, C++ o funciones predefinidas del equipo. En la Figura 4.3 puede apreciarse una imagen de la ventana de inicio del software SmarTest del equipo. Con ayuda de este software es posible configurar los test necesarios, realizarlos y visualizar los resultados entre otras opciones.

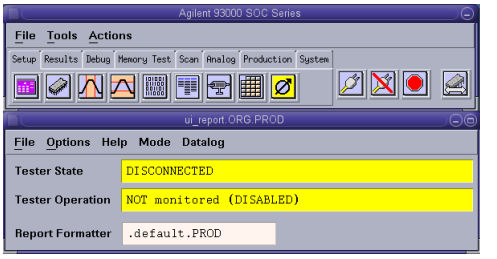

Figura 4.3.: Ventana de entorno del software HPSmarTest.

En cuanto a la configuración del equipo, consiste en el establecimiento de los pines, niveles de tensión (de entrada, salida y de polarizaciones), temporización y definición de patrones de test. La configuración de pines asocia cada pin físico a un nombre lógico, pudiendo ser estos de entrada, de salida, bidireccional o de alimentación, individuales o por grupos. Por otro lado, se encuentran las configuraciones de niveles de tensión de polarización y niveles lógicos, siendo estos cero, uno y el nivel lógico de comparación para las señales de salida. Otra de las configuraciones básicas es la temporización, donde se definen las posibles formas que pueden tener las señales de entrada de cada uno de los periodos de los test y los instantes de muestreo de las señales de salida. Por último se encuentra la configuración de los patrones de vectores de test para cada pin de entrada o salida, los cuales vienen definidos por los valores lógicos anteriormente configurados. En la Figura 4.4 puede apreciarse la ventana de configuración, denominada Setup.

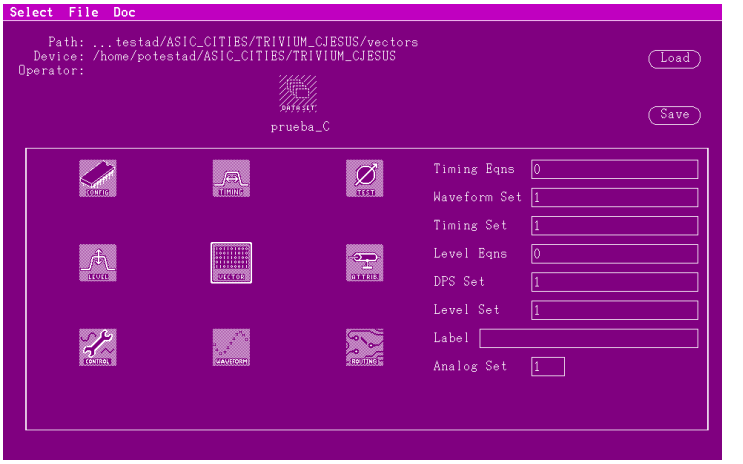

Figura 4.4.: Ventana Setup del equipo de test.

Entre todas las opciones disponibles, analizaremos las más importantes para nuestro objetivo. Como primera función necesaria se encuentra la configuración de los pines de entrada y salida, donde el programa permite definir los nombres de las entradas y salidas, asociándolos a los pines físicos del circuito. Con esta función quedan perfectamente definidos cada uno de los pines para posteriormente identificarlos a la hora de introducir patrones o leer datos de salida tras las pruebas.

En un segundo momento hay que crear y definir las formas de onda de entrada, donde la más importante para el objetivo de inserción de errores es la señal de reloj. Con ayuda de la ventana de temporización pueden definirse varias ventanas temporales, cada una de las cuales generará un comportamiento temporal diferente. Estas definiciones pueden observarse en la Figura 4.5, donde pueden verse las diferentes ventanas de definición de formas de ondas necesarias para realizar las pruebas. Cada una de estas ventanas tendrá una etiqueta para poder asociarla a señales y a ciclos concretos de un patrón de test. Así para el caso concreto de la señal de reloj se han generado tres ventanas temporales diferentes para CLK_FT y una para CLK2_FT. Como puede verse en la Figura 4.5, la señal denominada CLK_FT tiene definido el valor 3 para la columna "DevCyc". Esto significa que la señal con la etiqueta CLK_FT tendrá tres formas de onda diferentes y que en función del valor seleccionado en el test, se utilizará una forma de onda u otra. En el caso mostrado en la imagen, esta señal cuenta con siete flancos, identificados por su número. En la posición de cada uno de ellos (menos en el caso del 2) se ha configurado para que se produzca una transición de 0 a 1 o de 1 a 0. Esta configuración será utilizada solo en aquellos ciclos en los que se quiera provocar un fallo. Además las posiciones de las transiciones se pueden parametrizar y así ajustar dinámicamente los valores a los necesarios para introducir los errores. La segunda señal denominada CLK2_FT, la cual posee una sola definición de señal, se ha configurado para que tenga una transición de valor en el flanco 2 de 0 a 1. Esta es la utilizada para el funcionamiento normal del reloj en la salida de datos de los registros intermedios.

Por otro lado se encuentran las configuraciones de las tensiones. Estas configuraciones permiten definir las tensiones de entrada del anillo de pads y del core del circuito, así como los valores de tensión para los niveles lógicos cero y uno, tanto de los pines de entrada como los de salida del circuito.

Por último, para la carga de patrones se pueden utilizar dos métodos, por un lado la carga de patrones obtenidos desde simulación mediante otra herramienta software, o por el contrario, haciendo uso de la función *Golden Device.* En el primer caso los patrones son generados de forma externa y son cargados en la herramienta SmarTest. En el segundo caso los patrones se generan a partir de un test propio de la herramienta SmarTest que permite realizar una generación de patrones esperados ideales a partir de unos patrones de entrada. Con este último solo es posible obtener las salidas esperadas, puesto que las entradas deben de proporcionarse al sistema para poder realizar este test.

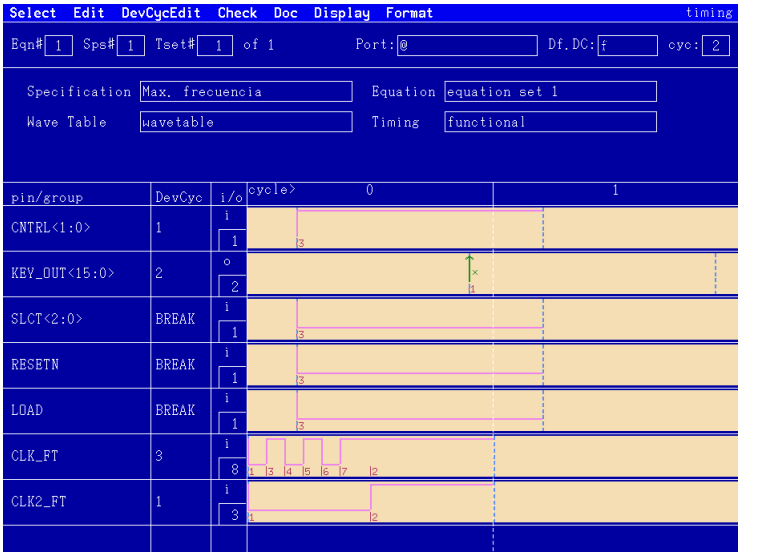

Figura 4.5.: Ventana definición señales de temporización.

Aplicaciones utilizadas para la obtención de resultados

Para comprender la metodología de inserción de fallos en las implementaciones mediante el uso de este equipo, es necesario conocer las funciones que ofrece este. Dado que las posibilidades que permite este equipo son muy amplias, por simplicidad, en este texto se hace referencia exclusivamente a las opciones utilizadas, presentándose por tanto en mayor detalle aquellas que son útiles para la obtención de los resultados.

Una vez configurado el sistema y cargados unos vectores de test para las entradas se realiza un test denominado *Golden Device.* Este test, cuya ventana puede apreciarse en la Figura 4.6, permite pasar los datos capturados por el equipo a datos esperados. Por ello, este test debe realizarse en unas condiciones que garanticen que el circuito funcione correctamente. Para el test *Golden Device* los parámetros de entrada utilizados son los mismos que en los casos donde se ataca el circuito, salvo que no está alterada la señal de reloj y permitiendo por tanto obtener el comportamiento ideal de los cifradores a lo largo del test completo. Los resultados de este test, son los utilizados para realizar la comparación con los resultados tras la realización de los ataques.

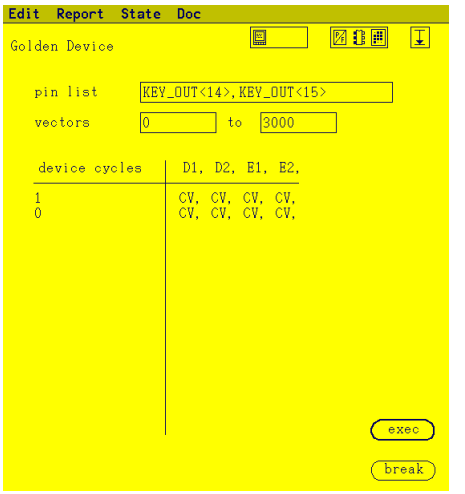

Figura 4.6.: Ventana de test Golden Device del equipo.

Otra aplicación básica en este equipo es el test funcional. De entre todos los posibles test que ofrece la herramienta, el test funcional es el más importante de todos, puesto que es el que va a permitir obtener todos los resultados de los ataques. Este tipo de test funcional consiste en una prueba donde se aplican los patrones de entrada al circuito y se comparan los datos recibidos con los datos esperados, siendo los patrones de entrada y salida esperados los obtenidos a partir del test *Golden Device*. Mediante dicha comparación se determina si el test ha pasado o si por el contrario ha fallado. Con el test funcional es posible determinar si se han producido fallos en el interior del circuito y a partir de este resultado volcar los datos de salida a fichero para su posterior tratamiento.

Una tercera aplicación muy utilizada de este equipo es la función denominada resultados, que de forma gráfica muestra en una pantalla un diagrama temporal de las entradas y las salidas. También muestra las diferencias con las salidas esperadas. En la Figura 4.7, puede verse representado el resultado obtenido y esperado de las pruebas. Esta función es muy útil debido a que, en nuestro caso, permitirá determinar de forma interactiva si han sido introducidos errores y por tanto se ha producido un ataque satisfactorio o si por el contrario no se han inducido errores, si ha habido un error de muestreo o si el pulso en la señal de reloj ha sido filtrado.

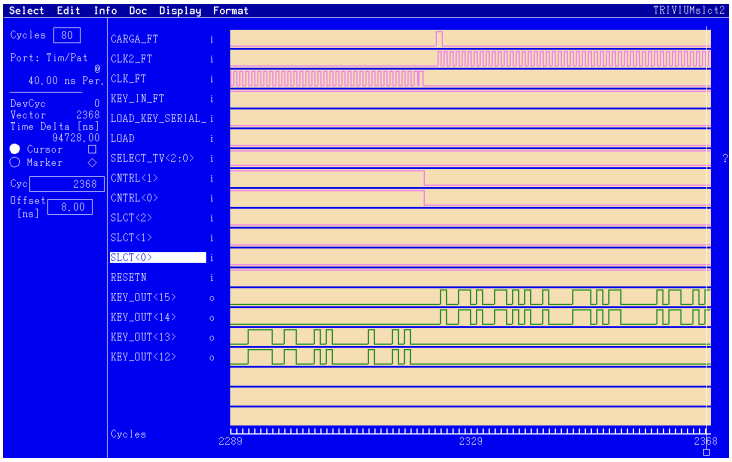

Figura 4.7.: Ventana de resultados de test.

Por último, se encuentra la función de representación de resultados en formato tabla mediante una lista de patrones y resultados. En la Figura 4.8 puede observarse una captura de la ventana de esta función. Esta aplicación permite visualizar en formato tabla los resultados y los patrones de entrada, permitiendo además un volcado de datos a un fichero para su posterior tratamiento para determinar el número de fallos introducidos y posiciones en los registros de estados internos de los cifradores.

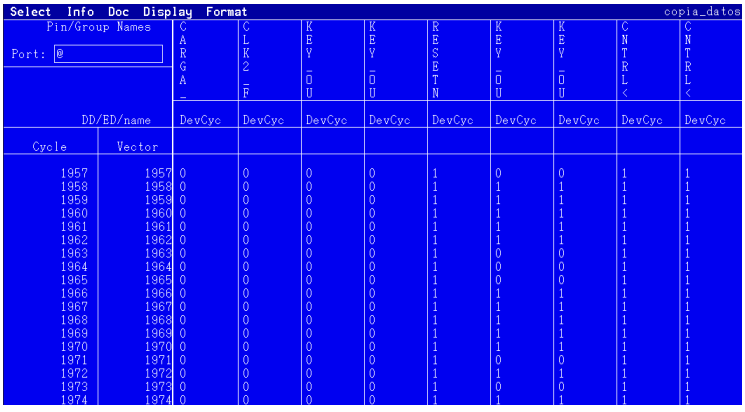

Figura 4.8.: Ventana de lista de patrones y resultados.

4.3. Creación del plan de test

Siguiendo la línea de trabajo de la presente tesis, se ha utilizado la manipulación de la señal de reloj para insertar fallos en los cifradores Trivium implementados en la tecnología ASIC. En esta ocasión, aprovechando las características que ofrece el equipo de test, se ha combinado la inserción de pequeños pulsos de reloj con la variación de la tensión de alimentación y temperatura. Además de estas combinaciones, se ha llevado a cabo otro tipo de ataque haciendo uso de las señales de control de los cifradores junto con la señal de reloj del sistema. Teniendo en cuenta el amplio abanico de variables para cada test, es necesario plantear de forma muy clara los pasos a seguir para poder estructurar todo el conjunto de pruebas. El flujo de trabajo se presenta en la Figura 4.9. Como puede verse, en un primer paso hay que definir las señales o formas de onda utilizadas para las pruebas. Después de esta primera definición, se encuentra la realización de test funcionales (con sus correspondientes generaciones de patrones) para corroborar el correcto funcionamiento, así como la determinación de las frecuencias óptimas para la inyección de errores. Tras esto, el siguiente punto consiste en la toma de datos y análisis de los mismos.

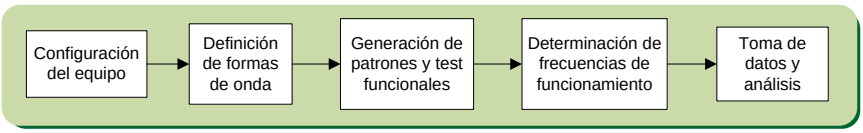

Figura 4.9.: Flujo principal de trabajo para la realización de las pruebas.

4.3.1. Definición de formas de onda

A la hora de realizar las pruebas de inserciones de fallos sobre los cifradores, es necesario definir una serie de señales o formas de onda de las señales de entrada. Dado que los ataques se realizan mediante la manipulación de la señal de reloj, al igual que ocurría en FPGA, es muy importante tener una gran precisión a la hora de generar las señales de reloj. Esta precisión engloba, además, a las señales de captura de datos y a las señales que se generan para inducir errores en los cifradores.

El equipo de test Agilent 93000 permite la definición de formas de onda que pueden ser replicadas el número de veces que sea necesario, permitiendo de

este modo generar por ejemplo una señal de reloj mediante el uso repetido de un único patrón. Para mostrar esto de forma gráfica se presenta la Figura 4.10. Como puede observarse en esta figura, es posible definir una señal mediante la utilización de flancos denominados desde d0 a d7 (en este caso solo se utilizan d0 y d1). En función de los valores que se le den a estos flancos, quedará definida la forma de onda de la señal.

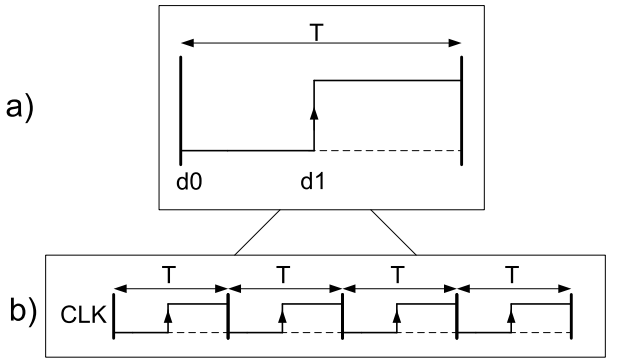

Figura 4.10.: Definición y utilización de forma de onda para la generación de una señal de reloj.

En la Figura 4.10 a), se ha presentado a modo de ejemplo la definición de una forma de onda mediante dos flancos, d0 a valor bajo y d1 a valor alto. La selección de esta misma forma de onda de modo secuencial es lo que permite la generación de la señal de reloj en este equipo de test, Figura 4.10 b). Además es posible la utilización de variables que modifiquen de forma dinámica los flancos, por ejemplo, es posible la utilización de una variable denominada "T_x" que multiplique los valores de tiempo en los que se producen los flancos, cambiando así de forma dinámica las formas de onda.

Teniendo en cuenta lo expuesto hasta ahora, se han definido diferentes formas de onda para la generación de las señales de reloj, así como dos formas de onda especiales para la inducción de pulsos cortos en dicha señal. En el caso de la señal de reloj normal se ha utilizado la forma de onda mostrada en la Figura 4.10 a), pero en el caso de la inducción del pulso corto se ha llevado a cabo el diseño de unas formas de onda especiales. Centrémonos en la definición de estas formas de ondas especiales de las que trataremos dos casos, uno con solamente un pulso corto (Figura 4.11) y otro con dos pulsos cortos (Figura 4.12). En el primer caso encontramos una forma de onda que presenta

un pulso corto en función de la variable "T_2" y en el segundo caso otra forma de onda que presenta dos pulsos cortos en función de esta misma variable. La necesidad de este último caso se debe al hecho ya mencionado de que, al filtrar los pads del chip pulsos de muy alta frecuencia, es preciso generar dos pulsos consecutivos para que llegue al interior del cifrador al menos uno en condiciones de inducir un fallo.

En el primer caso, Figura 4.11, se tiene una forma de onda definida por los flancos d3, d4, d5 y d6, cuyos valores están multiplicados por la variable "T_2". Esta variable permite la modificación de la posición de los flancos, haciendo que estos se encuentren más cercanos unos a otros y por tanto aumentar o disminuir el tiempo de los pulsos. Así puede obtenerse un pulso de reloj formado por los cuatro flancos. Por otro lado, la señal completa está definida por una variable denominada "T", global a todas las formas de onda, la cual establece el ancho completo de todas las señales de patrones del sistema. Esta forma de onda viene definida por los valores de los flancos dados en 4.1. Esta señal será utilizada en los casos en los cuales la frecuencia de operación de los cifradores no sea muy elevada.

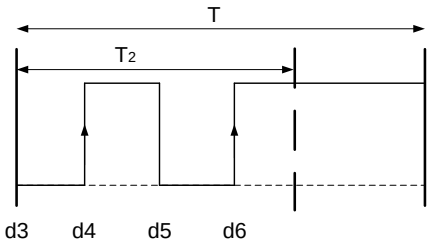

Figura 4.11.: Definición de la forma de onda de la señal con un sólo pulso.

$$d_3 = 0; \quad d_4 = 0,25T_2; \quad d_5 = 0{,}5T_2; \quad d_6 = 0{,}75T_2 \tag{4.1}$$

En el segundo caso, Figura 4.12, se tiene una forma de onda definida por los flancos d0, d3, d4, d5, d6 y d7, siendo sus valores temporales una fracción de la variable "T_2". En este caso, la señal presenta dos pulsos cortos que permiten, como se explicó anteriormente, que los pads de entrada del circuito toleren uno de estos dos pulsos. Al contrario que el caso anterior, esta señal será

utilizada en los casos de mayor frecuencia de operación donde un pulso puede ser filtrado. Esta señal viene definida por los valores de los flancos dados en 4.2.

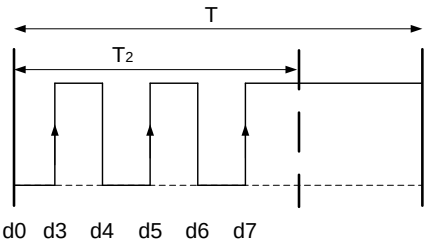

Figura 4.12.: Ventana de definición de la señal con dos pulsos.

$$d_0 = 0;\ \ d_3 = 0,15T_2;\ \ d_4 = 0,3T_2;\ \ d_5 = 0{,}45T_2;\ \ d_6 = 0{,}6T_2;\ \ d_7 = 0,75T_2 \tag{4.2}$$

Con esto quedan definidas las formas de onda que se van a utilizar para generar la señal de reloj. La utilización de una u otra dependerá del caso en que se encuentre el sistema bajo prueba, ya sean pruebas que utilicen una frecuencia muy elevada o de pruebas donde la frecuencia no es llevada al límite de poder filtrar los pulsos inducidos.

Caso especial: manipulación de la señal de control

Este caso se puede considerar especial debido a que la señal de control no se manipula para generar un pulso pequeño, sino para adelantar o retrasar el instante de cambio de dicha señal. Este planteamiento sigue la misma metodología que los anteriores, en cuanto que utiliza una forma de onda donde se modifica una variable denominada "T_3" (Figura 4.13). El objetivo de esta variable es acercar o alejar los flancos de la señal de control al flanco de subida de la señal de reloj. Esto permite actuar sobre el instante de tiempo que hay entre el flanco de reloj y el cambio de las señales de entrada de los biestables, lo que se puede traducir en violaciones de tiempo de setup. De esta forma,

la manipulación de los flancos de la señal de control puede provocar que al detener y activar el cifrador se inyecten errores en los registros internos, dado que la señal de control no se encuentra balanceada como lo está la señal de reloj que utiliza lineas especiales de propagación.

En la Figura 4.13 puede observarse la definición de la forma de onda que permite la manipulación de la señal de control en función del valor de la variable "T_3" que modifica el flanco d2 desde 0 hasta T.

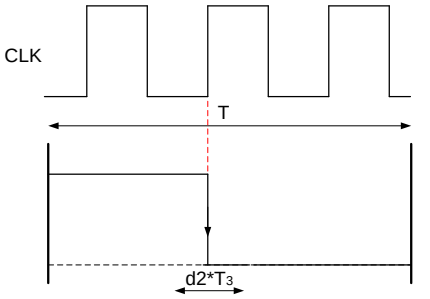

Figura 4.13.: Ventana de definición de la señal de control manipulable bajo señal de reloj.

4.3.2. Características del plan de test

Una vez realizada la configuración y definidas las formas de ondas, el siguiente paso consiste en la generación de patrones y realización de test funcionales que permitan verificar el correcto funcionamiento de los diferentes circuitos de cifradores incluidos en el ASIC. En la Figura 4.14 se ha representado el desglose de la etapa de generación de patrones y test funcionales para las pruebas de inserciones de errores. Esta etapa tiene como primer proceso el planteamiento de test y determinación de secuenciación de las señales de entrada que modelan cada una de los test a realizar, como son los ciclos de carga de clave secreta e IV, la secuenciación de procesos de carga, el funcionamiento y muestreo, etc. Tras esto, se establece qué claves y vectores de inicialización se utilizarán en las pruebas, así como los ciclos donde se inyectarán los pulsos de reloj que permitirán la inducción de errores, el muestreo de resultados, etc. Llegados a este punto es posible generar los patrones de funcionamiento que permitan la realización de cada uno de los test a realizar. En el siguiente paso se aplicarán estos patrones de test para llevar a cabo los tests.

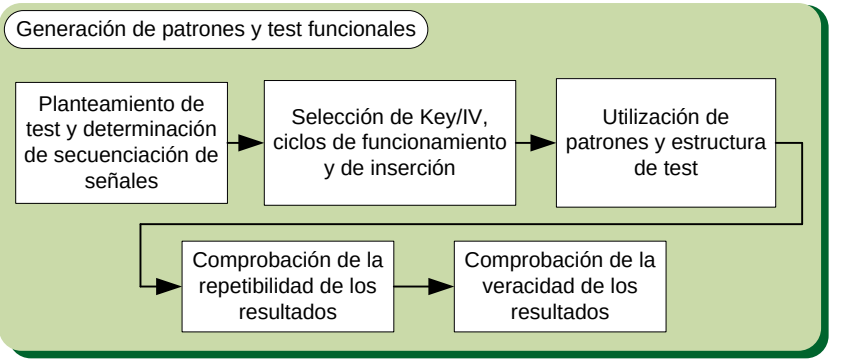

Figura 4.14.: Desglose de la etapa de generación de patrones y test funcionales.

Los siguientes pasos están destinados a la comprobación de la repetibilidad de resultados así como a la comprobación de la veracidad de los datos capturados. Es decir, se comprueba funcionalmente que los fallos producidos y detectados en los cifradores de flujo son reales. Esta comprobación se basa en el hecho de que si un error introducido en un instante determinado no genera el mismo registro interno cientos de ciclos después que el que puede generar una simulación funcional, este error será un falso positivo o un error de captura. Con este proceso se asegura la veracidad de las inserciones y del funcionamiento del sistema de inserción de fallos ya que un error real debe ser propagado a lo largo del tiempo en el interior del cifrador.

Los test que se han realizado sobre los circuitos tienen las siguientes características.

1. Realizar pruebas funcionales. Aunque los chips ya han sido probados, se vuelve a comprobar su correcto funcionamiento a frecuencias bajas con una tensión nominal, mínima y máxima.
2. Funcionar un número de ciclos determinado y capturar datos. El test debe permitir funcionar al sistema una serie de ciclos establecidos y tras la inserción del pulso, detener el reloj y capturar los registros para su posterior procesamiento. Esta característica permite observar si siempre se repite el mismo fallo bajo unas mismas condiciones. Este proceso se realiza para cada situación y debe obtener tanto el número de fallos como la posición de los fallos. Estos dos análisis se tendrán que hacer mediante procesamiento de datos posteriores.

3. Reiniciar y repetir el test. Se debe repetir la prueba bajo las mismas condiciones el número de veces que se estime oportuno. Esta repetición consistirá en volver a realizar la carga de los valores de clave y vector de inicialización y hacerlo funcionar el número de ciclos establecido. El reinicio se realiza para poder ver la repetibilidad de los fallos inducidos en los registros internos.
4. Cambiar ciclo de inserción. Para un mismo cifrador y una misma frecuencia de fallo, se debe poder cambiar el ciclo en el que inyecta el pulso en la señal de reloj. Esto se realiza con el objetivo de observar si la vulnerabilidad de ciertos bits se presentan de igual forma cuando el pulso se induce en un ciclo donde tienen transición de valor o no. Es conveniente probar la inserción en ciclos diferentes para tener resultados suficientes que permitan generalizar las conclusiones.
5. Cambiar cifrador y frecuencia utilizados. Esta característica se debe implementar para obtener resultados en todos los cifradores disponibles. Deben ser independientes las pruebas con respecto al cifrador utilizado. Debe estar perfectamente diferenciado e identificando en cada caso qué cifrador se prueba y en que ciclo se ataca.

En cuanto al proceso de test en sí, en general tiene las siguientes características:

1. Debe ser lo más automático posible. Esta característica es importante ya que son muchas las pruebas a realizar. Teniendo en cuenta que las condiciones de las pruebas van a ser alteradas, es conveniente que los test se realicen de la forma más automática posible para que estos cambios no deriven en un gasto de tiempo considerable. Por otro lado, esta característica permitirá obtener un mayor número de resultados y por tanto un mayor banco de pruebas con el que poder obtener conclusiones.
2. Procesamiento de datos. Los resultados deben ser exportados en modo texto con el objetivo de poder analizarlos de forma automática mediante un procesador de datos.
3. Análisis de fallos frente a la anchura del pulso. Hacer análisis del número de fallos introducidos en función del ancho del pulso utilizado y además ver su repetibilidad.
4. Comprobación de veracidad. Para estar seguros de que las pruebas realizadas y que los fallos insertados se están haciendo correctamente se realizará el siguiente procedimiento:

Funcionamiento – inserción – parada – muestreo – funcionamiento – parada – muestreo.

El cifrador debe de funcionar unos ciclos determinados. Tras ese número de ciclos se inserta el pulso en la línea de reloj, se para el reloj y se capturan los datos de los registros de desplazamiento. Tras esta captura, se volverá a activar el funcionamiento durante una serie de ciclos lo suficientemente alto como para que, en el caso de una inyección de fallos, ambos cifradores implementados en paralelo y atacados presenten un registro completamente distinto al que deberían tener. Tras esto, se detendrán los cifradores y se volverán a capturar los datos que serán comparados mediante simulación para poder demostrar la veracidad de los resultados.

Esta parte de los test se realizará una vez por cifrador, pareja de clave e IV, ciclo de inserción y valores de tensión debido a que debe simularse manualmente y reproducir el error introducido. Por lo tanto, estas pruebas serán independientes de las anteriores e incluso pueden realizarse como primera prueba antes del reinicio del sistema, sabiendo que el resultado a comparar posteriormente con la simulación será el primer resultado obtenido para cada uno de los ciclos de inserción.

En resumen, la Figura 4.15 presenta un diagrama del planteamiento del proceso de test que se debe realizar para poder obtener conclusiones de los ataques realizados, tanto de su veracidad como de su repetibilidad.

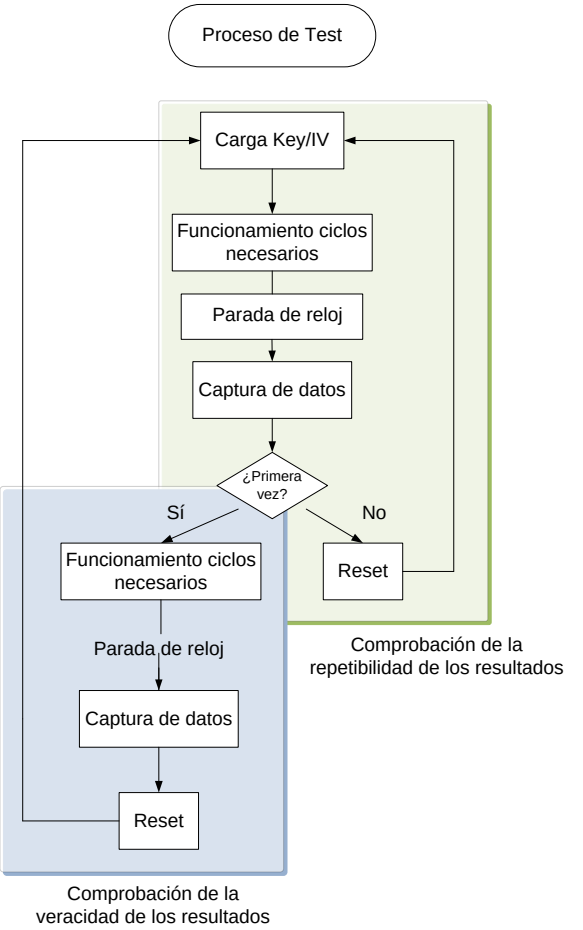

Figura 4.15.: Proceso de test para repetibilidad y veracidad de resultados.

Además de los procesos considerados anteriormente, una vez realizados los test funcionales y de ataque sobre los cifradores, se plantean distintos escenarios dado que los cifradores presentan diferentes comportamientos en función de las condiciones de las pruebas. Estos escenarios contemplan las siguientes condiciones de inyecciones de fallos mediante la combinación de ataques:

1. Ataques utilizando el cambio de los valores de las tensiones de alimentación. Todas las pruebas realizadas para el valor óptimo de alimentación del core serán realizadas para un valor mínimo de alimentación. Esto se

realiza para conocer si aumenta la vulnerabilidad del circuito debido al cambio de las condiciones.

2. Análisis con Vcc. Análisis combinado de la inserción de fallos en función del ancho de pulso y el valor de la tensión de alimentación.

3. Análisis con Vcc variando la temperatura. Observar la evolución del comportamiento del cifrador en el caso anterior, pero esta vez variando la temperatura del circuito.

4. Fallos por cambio de tiempo en entradas de control. Analizar el comportamiento del cifrador si se manipula la señal de control en determinados instantes de tiempo.

Este último punto implica una combinación de manipulación de la señal de reloj y además de las señales de control (descrito en el apartado anterior como caso especial). Esta combinación de ataque se describe con más detalle en la sección de resultados 4.4.

4.3.3. Determinación de rangos de funcionamiento correcto y con fallos

Una vez estructurado el flujo de operación para llevar a cabo los ataques y pruebas de funcionamiento sobre la implementación, es necesario determinar el valor del los tiempos que deben tener los pulsos en la señal de reloj para introducir un fallo en el cifrador. Este proceso permite conocer el valor de la variable T_2 que modela el ancho de pulso de las ventanas descritas en la subsección 4.3.1 y por el periodo de reloj máximo de funcionamiento de cada cifrador. La determinación de este periodo permitirá establecer un rango de tiempos en torno al cual se inducirán errores en los registros internos de los cifradores.

Como se explicó anteriormente, esta determinación de rangos de periodos de tiempo de T_2 se realiza bajo diferentes condiciones de tensión y temperatura. Las pruebas fueron realizadas con tensión nominal, tensión mínima, tensión máxima, temperatura nominal, temperatura mínima y temperatura máxima. Estos valores de temperatura y tensiones son los recogidos en la Tabla 4.1.

Tabla 4.1.: Escenarios contemplados para la realización de las pruebas.

	Mámimo	Nominal	Mínimo
Temperatura (°C)	80	25	0
Tensión (V)	2.5	1.2	0.8

Conjunto 1: utilización de un sólo pulso de reloj.

En primer lugar se ha determinado el conjunto de rangos de periodos útiles para la inserción de errores en el caso de la forma de onda de un solo pulso de reloj. Los resultados de nuestro estudio indican que los periodos útiles de inserción son prácticamente insensibles frente a la temperatura. Efectivamente, los valores obtenidos para la temperatura mínima y máxima son prácticamente iguales a los obtenidos para temperatura ambiente. En cuanto a la variación de dichos periodos frente a la tensión, hemos comprobado que no existen diferencias entre la aplicación de la tensión nominal y de la tensión máxima al circuito. Por tanto los únicos resultados interesantes que se presentan corresponden a temperatura ambiente en combinación con los dos casos de tensión: nominal y mínima.

En la Tabla 4.2 se presentan los valores límite del tiempo entre dos flancos de subida en la señal de reloj para condiciones de funcionamiento correcto y fallo, en dos casos: el Caso 1 corresponde al de tensión nominal y el Caso 2, al de tensión mínima.

En el Caso 1 los valores del tiempo límite entre flancos del pulso de reloj necesario para inducir errores en los cifradores varían entre algo más de 7.1 ns y 15 ns. En los casos de los cifradores Estándar, x2 Estándar y x8 Estándar, sin embargo, no se han podido medir sus valores límites. La razón está en que, al ser tan pequeño el tiempo necesario entre flancos, el pulso de corta duración que se genera es filtrado por los pads del circuito y, por tanto, no se induce ningún fallo en el cifrador. En el Caso 2 (tensión mínima) se produce un notable aumento en los valores del tiempo necesario para inducir errores, que ahora se sitúan entre 9 ns y 32.5 ns, sin que esté excluido ninguno de los cifradores. Este caso trae consigo una mejora a la hora de poder insertar errores en los registros internos de los cifradores, haciendo más fácil los ataques.

Tabla 4.2.: Límites de tiempo entre flancos de subida para T_2 cuando se utiliza un solo pulso de reloj para inducir fallos. Caso 1: Tensión nominal. Caso 2: Tensión mínima.

	Caso 1		Caso 2	
Trivium	Máx. (ns)	Mín. (ns)	Máx. (ns)	Mín. (ns)
Estándar	-	-	15	9
Low Power	7.5	7.1	22	15
x2 Estándar	-	-	10	9.5
x2 Low Power	8	7.15	21	17
x8 Estándar	-	-	16	14.3
x8 Low Power	13	12.5	29	25
x16 Estándar	8	7.05	20	17
x16 Low Power	15	14.89	32.5	31.5

Conjunto 2: utilización de dos pulsos de reloj.

En este segundo conjunto se ha tenido en cuenta la forma de onda de generación de señal de reloj con dos pulsos cortos seguidos. Ahora lo que se determinará son los límites de tiempo entre el primer flanco de subida y el tercero, es decir, los dos pulsos de reloj en los que habría que moverse para poder insertar fallos efectivos en los cifradores de flujo. Al igual que en el apartado anterior, este conjunto de límites de tiempo entre flancos de reloj se divide en dos casos, los cuales se muestran en la Tabla 4.3: los valores del Caso 1 han sido obtenidos con tensión nominal y los del Caso 2 con tensión mínima.

Como puede observarse en la Tabla 4.3, el uso de dos pulsos de reloj en ambos casos permite determinar los límites de tiempo de los pulsos de reloj para insertar errores en todos los cifradores. Esto presenta una notable ventaja frente al uso de un solo pulso de reloj donde insertar fallos en los cifradores estándar se veía afectado por el filtrado de los pads debido el alto valor del límite de tiempo necesario. Por el contrario, al igual que el caso de un solo pulso de reloj, el uso de las diferentes combinaciones de condiciones de funcionamiento del circuito en cuanto a temperatura mínima y máxima o tensión nominal y máxima, no ofrece datos relevantes que favorezcan su utilización, ya que los resultados son iguales a los presentados en la Tabla 4.3.

Tabla 4.3.: Límites de tiempo entre flancos de subida para T_2 cuando se utilizan dos pulsos de reloj para inducir fallos. Caso 1: Tensión nominal. Caso 2: Tensión mínima.

	Caso 1		Caso 2	
Trivium	Máx. (ns)	Mín. (ns)	Máx. (ns)	Mín. (ns)
Estándar	7.5	6.95	16	12
Low Power	8.92	6.92	22	15
x2 Estándar	7.1	7	14	11
x2 Low Power	9	7	21	18
x8 Estándar	8.8	6.9	16	14
x8 Low Power	14.7	13	29	27
x16 Estándar	9.8	7.4	21	17
x16 Low Power	17.1	14.5	32.5	31.5

Caso especial: manipulación señal de control.

Para el caso especial donde la forma de onda manipulada es la señal de control, se realizaron test para conocer el rango de tiempos de cambio respecto a la señal de reloj solo a temperatura ambiente y tensión nominal. Esto se debe a que si se tienen en cuenta los casos antes considerados, el más desfavorable para inducir errores en los cifradores es el de tensión nominal. Por lo tanto, analizando la posibilidad de inserción de un único error en los cifradores bajo la condición de tensión nominal es suficiente para concluir que con tensión mínima también es posible. En la Tabla 4.4 se puede observar el rango de tiempos, cerca del flanco de subida de reloj, en torno al cual debe producirse el cambio en la señal de control para poder inyectar errores en los cifradores de flujo[2]. Cabe destacar que en este caso, el valor de la variable T_3, en nanosegundos hace referencia a la proximidad en tiempo del flanco de bajada de la señal de control respecto del flanco activo de reloj del sistema. Esta proximidad provocaría una violación de tiempos de setup en diferentes biestables del circuito que se traducirían en inyecciones de errores en los registros de desplazamiento.

[2]Las pruebas de inserción de errores utilizando este método, sólo se han realizado sobre la implementación estándar ya que el comportamiento y resultados, al igual que en los casos anteriores, es extrapolable a los demás cifradores.

Tabla 4.4.: Límites de los valores de tiempo de la variable T_3 para inducir errores mediante la señal de control.

Trivium	T_3 (ns) Mín.	T_3 (ns) Máx.
Estándar	15.05	16.05

Los valores recogidos en la Tabla 4.4 representan el mínimo y máximo valor de la variable T_3 que debe multiplicar al flanco de bajada "d2" de la señal de control para que se produzca una inserción de fallo en los registros internos del cifrador por violaciones de tiempos. En la Figura 4.16 puede verse este resultado de forma gráfica. Las franjas verdes representan los valores de tiempo de T_3 para los cuales, si se produce el flanco de bajada de la señal de control, no produciría un funcionamiento anómalo del dispositivo, mientras que la franja roja representa el intervalo de tiempos en los cuales se inducirían errores en el registro.

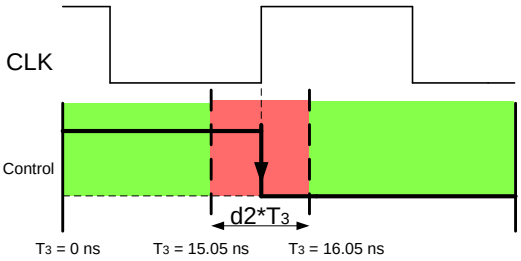

Figura 4.16.: Representación del rango de tiempos de T_3 para los cuales es posible inducir fallos mediante la señal de control.

Conclusiones en cuanto a la selección de rangos de tiempo para inducción de fallos.

Una vez determinados los rangos temporales en torno a los cuales se deben de mover los pulsos de reloj para la inserción de errores, se puede observar que la opción de utilizar dos pulsos cortos de reloj es más eficiente que la de un solo pulso. Este hecho se debe, como ya se explicó anteriormente, a que en el caso de un pulso existe la posibilidad de que se produzca un filtrado de la señal por

parte de los pads del circuito. Sin embargo en el caso de operación de tensión mínima, el uso de un solo pulso de reloj es suficiente, puesto que el valor de T_2 necesario disminuye notablemente.

4.3.4. Captura de datos y análisis

Tras definir las señales, flujo de test y rangos de tiempos en los pulsos de la señal de reloj para llevar a cabo los ataques, la siguiente etapa consiste en la realización de los test específicos, captura de datos y análisis de resultados. En la Figura 4.17 puede apreciarse el conjunto de la etapa de captura de datos. Como puede observarse en esta etapa, los test realizados pasan a ser específicos en cuanto al periodo de reloj seleccionado respecto de los rangos determinados anteriormente. A partir de este punto y una vez determinado el tiempo del pulso insertado en la señal de reloj, se realizan los tests mediante carga de clave e IV, funcionamiento de número de ciclos (variable o no en función de si se está repitiendo la misma prueba o no) con un reloj lento que garantice el correcto funcionamiento e inserción del pulso de reloj. La captura de datos se realiza muestreando el registro de estado de los cifradores a un registro interno con una señal de reloj "clk2" diferente a la de los cifradores Triviums. El contenido de este registro interno se saca al exterior de forma serie mediante un pin de salida durante 288 ciclos de reloj, uno por cada bit del registro interno. Estos datos son exportados a diferentes ficheros para posteriormente ser analizados en conjunto.

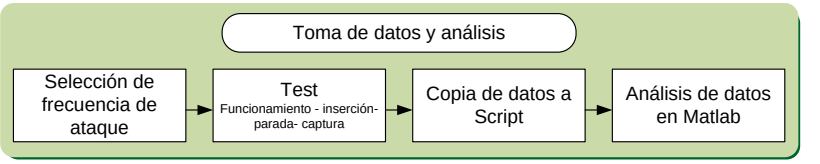

Figura 4.17.: Etapa de toma y procesado de datos de las pruebas

El análisis de datos se lleva a cabo mediante la utilización de la herramienta Matlab con la creación de una serie de scripts que permiten la lectura, limpieza datos innecesarios y comparación de datos de interés. En la Figura 4.18 puede observarse el flujo seguido a la hora de obtener los resultados finales de cada ataque. En este proceso se lleva a cabo una lectura de ficheros de datos, obtenidos a partir del ataque en el equipo 93000, una limpieza en cuanto a da-

tos del fichero innecesarios y por último una comparación mediante un script que muestra por pantalla el número de fallos inyectados y la posición de los mismos en los registros de estados interno de los cifradores.

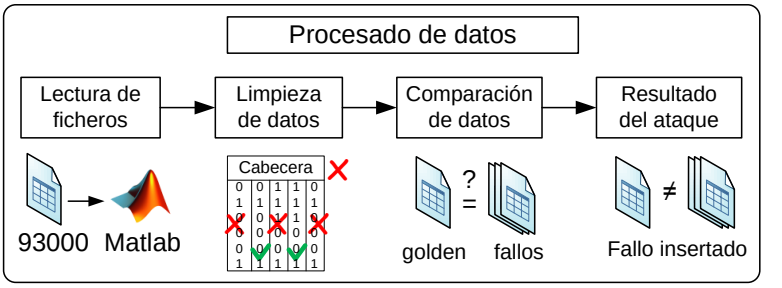

Figura 4.18.: Procesamiento de datos y obtención de resultados finales.

4.4. Resultados obtenidos

A partir de lo explicado anteriormente, se ha obtenido un conjunto de resultados, de los cuales en este capítulo solo se presentan los correspondientes al cifrador Trivium Estándar y Trivium Low Power. La totalidad de los resultados obtenidos se puede consultar en el Apéndice A. Respecto a la variación de parámetros, solo se presentan los resultados obtenidos en condiciones de tensión de alimentación nominal y mínima. Algunos resultados se muestran haciendo uso de la herramienta "*Shmoo Plot*", como por ejemplo los mostrados en la subsección de resultados variando tensión de alimentación y periodo de reloj. Esta herramienta permite visualizar de forma gráfica los errores inyectados cuando se van variando estos dos parámetros. Por otro lado, los resultados obtenidos cuando se manipula la señal de control se presentan en una subsección diferente al final de este apartado.

Teniendo en cuenta que los resultados preliminares y los obtenidos sobre implementaciones en FPGA mostraban que los biestables que tendían a fallar eran los de realimentación, debido a sus mayores retrasos temporales, se estudió qué ciclos de reloj permitían la inserción de fallos en cada uno de estos bits realimentados. Este estudio contempla el hecho de que un biestable cambie su valor en el ciclo de inserción del pulso de reloj, puesto que, en caso de no hacerlo, no se le puede introducir un fallo.

En la Figura 4.19 puede observarse una representación de los casos mencionados. Por un lado se puede apreciar el "Caso a" donde la inducción del pulso de reloj se realiza en un instante donde el próximo valor del biestable de realimentación coincide con su valor actual. Como consecuencia de esto, aunque el nuevo valor en su entrada incumpla las restricciones temporales del biestable debido a los retrasos de las líneas de realimentación, no se puede producir un fallo en el biestable porque no tiene que cambiar su valor. Esto hace que la inyección de fallos no sea efectiva. Por el contrario en el "Caso b" el valor que debería realimentar al biestable llega tarde y el biestable no actualiza su valor (de cero a uno) y por tanto se mantiene el valor antiguo, es decir cero. Esto provoca una inserción de error efectiva y por consiguiente que el error se propague en los ciclos posteriores a través del registro de estados interno.

Para contemplar estos casos, se realizó un análisis de las transiciones de los biestables en varios ciclos de inserción para asegurar que todos los ciclos seleccionados contenían todos los casos posibles, es decir, que todas las realimentaciones tuviesen transición, de cero a uno o de uno a cero, en cada ciclo de inserción.

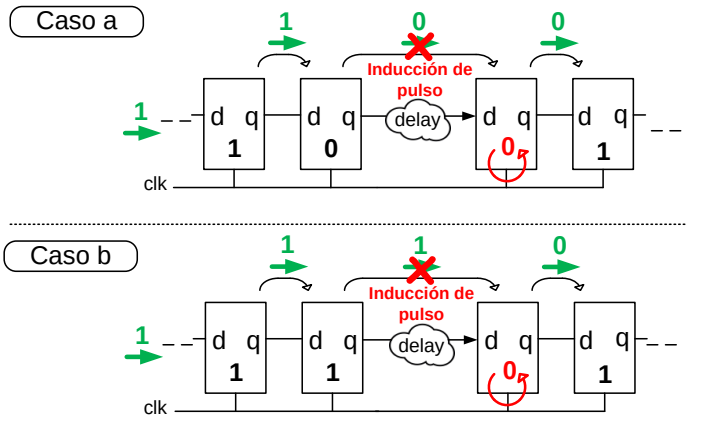

Figura 4.19.: Representación de inserción en función de la realimentación.

4.4.1. Resultados obtenidos sobre Trivium Estándar

Transiciones de biestables en varios ciclos de inserción

Para llevar a cabo este análisis, se seleccionó una clave e IV fijos para poder analizar las transiciones de los biestables realimentados en diferentes ciclos de reloj. En la Tabla 4.5 se presentan los ciclos de reloj seleccionados y las transiciones en los biestables de realimentación del registro de estados interno, es decir, los bits 0, 93 y 177. De entre todos los ciclos de reloj posibles, se han tenido en cuenta aquellos donde al menos en uno de los biestables realimentados se produzca transición y un caso donde no se produzca transición en ninguno de ellos. Además, se tienen en cuenta diferentes ciclos de reloj donde el biestable que presenta transición es el mismo. Esto último se realiza con el objetivo de poder analizar el comportamiento del biestable realimentado bajo un mismo ataque donde se varíe el ciclo de inserción. Por tanto, la Tabla 4.5 permite determinar el por qué fallan unos biestables de realimentación y otros no debido a que la inserción de errores depende de si se produce un cambio de valor de dichos biestables en el ciclo elegido.

Tabla 4.5.: Transición en los bits de realimentación en varios ciclos de reloj.

	Bits realimentación		
Ciclo	0	93	177
1307	No	No	No
1312	Si	Si	No
1351	No	No	Si
1352	No	No	Si
1402	No	No	Si
1403	No	Si	No
1545	Si	Si	Si
1546	No	Si	Si

Resultado con tensión nominal

Se han realizado varias pruebas de inyección de fallos utilizando los períodos de reloj de la Tabla 4.6 en condiciones de tensión nominal (1.2 V). Estos

valores están comprendidos entre los obtenidos en la sección 4.3.3 donde se determinaba el rango de periodos de reloj válidos para conseguir la inyección de errores en los registros internos.

Tabla 4.6.: Periodos de reloj del ancho de pulso T_2 para cada test con tensión nominal.

	Test 1	Test 2	Test 3	Test 4	Test 5
Periodo (ns)	7.045	7.035	7.03	7.025	6.95

Haciendo uso de estos periodos de reloj, en la Tabla 4.7 se presenta el número de fallos insertados en el registro de estado interno del cifrador por cada ciclo de inserción. Esta tabla presenta resultados para dos copias del mismo cifrador, ya que el ASIC incorpora dos implementaciones de cada cifrador para poder evaluar si el routing influye o no en la inyección de fallos. *Trv 1* y *Trv 2* se corresponden con los resultados para cada una de estas implementaciones. En esta tabla se puede ver cómo se produce una dependencia del número de fallos inyectados en función del ancho del pulso utilizado. A medida que se utiliza un periodo de reloj para el pulso cada vez más pequeño, aumenta el número de fallos insertados hasta el punto de fallar un gran número de ellos. En concreto, en los casos del Test 4 y del Test 5, se puede apreciar un aumento brusco en el número de fallos inyectados respecto de los demás test.

La Tabla 4.8 muestra las posiciones de los fallos insertados, es decir, los biestables específicos de cada uno de los cifradores, que han fallado tras la inducción del pulso en la señal de reloj. Por otro lado, hay que tener en cuenta que, en los casos en los que se producen numerosos fallos a la vez, no se muestran las posiciones de los bits por simplicidad de los resultados. En su lugar, en la tabla aparece un "$*$". Los casos donde no se han podido inyectar fallos se representan mediante "-".

Teniendo en cuenta estos resultados, se puede ver cómo los biestables con realimentaciones tienden a fallar antes que los demás biestables. Por ejemplo, en cada uno de los test, en los ciclos de inserción 1312 y 1545, los biestables que han fallado son las posiciones realimentadas 0 y 93, presentando ambas copias una tendencia a fallar en las mismas posiciones. Además puede apreciarse cómo la posición menos vulnerable es la 177 donde no se han podido inyectar fallos en estas pruebas.

Tabla 4.7.: Número de fallos insertados en Trivium Estándar en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2
1307	0	0	0	0	0	0	127	127	127	127
1312	2	1	1	1	2	1	2	1	2	1
1351	0	0	0	0	0	0	0	0	0	60
1352	0	0	0	0	0	0	122	61	122	122
1402	0	0	0	0	0	0	0	0	138	138
1403	0	0	0	0	0	0	0	0	1	0
1545	1	0	1	0	1	0	1	0	1	0
1546	0	0	0	0	0	0	0	0	142	142

Tabla 4.8.: Posiciones de los fallos insertados en Trivium Estándar en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2
1307	-	-	-	-	-	-	*	*	*	*
1312	0/93	0	93	0	0/93	0	0/93	0	0/93	0
1351	-	-	-	-	-	-	-	-	-	*
1352	-	-	-	-	-	-	*	*	*	*
1402	-	-	-	-	-	-	-	-	*	*
1403	-	-	-	-	-	-	-	-	93	-
1545	93	-	93	-	93	-	93	-	93	-
1546	-	-	-	-	-	-	-	-	*	*

Por otro lado, estos resultados muestran implícitamente que la correcta selección del ciclo de inserción permite obtener mejores resultados dado que, como puede observarse, existen ciclos de inserción más efectivos que otros, es decir, ciclos de inserción donde para cada test es posible inyectar fallos, mientras que en otros no es posible inyectarlos. Por último, como resultado más importante, puede observarse que es posible inyectar un solo fallo efectivo en los registros internos, utilizando diferentes periodos de reloj y ciclos de inserción para am-

bos cifradores implementados. Con ello se puede concluir que, con esta técnica de ataque, este tipo de cifradores es vulnerable frente a este tipo de ataques cuando son implementados en ASIC.

Resultado con tensión mínima

Una vez presentados los resultados con tensión nominal, se presentan los resultados correspondientes a tensión de alimentación mínima (0.8V). Para estas pruebas, se han utilizado los valores de periodos de reloj presentados en la Tabla 4.9. Como puede observarse, los periodos de reloj son prácticamente el doble que para el caso de tensión nominal (1.2V). Por otro lado, los ciclos de inserción utilizados son los mismos que para las pruebas realizadas a tensión nominal, puesto que no se ha cambiado la clave ni el IV.

Tabla 4.9.: Periodo de reloj del ancho de pulso T_2 para cada test con tensión mínima.

	Test 1	Test 2	Test 3	Test 4	Test 5
Periodo (ns)	15	14.5	14	12.5	9

En la Tabla 4.10 se presentan los resultados del número de fallos inyectados para cada uno de los ciclos de inserción y cifrador. Al igual que anteriormente puede apreciarse como a medida que el periodo de reloj disminuye del Test 1 al Test 5, aparecen mayor número de inyecciones de fallos en los registros internos.

Al igual que anteriormente, en la Tabla 4.11 se muestran las posiciones de los bits que han fallado durante las pruebas, es decir, los bits donde se han inyectado errores. Como puede observarse, la dinámica continúa siendo la misma, es decir, se sigue produciendo una tendencia de fallo en los biestables de realimentación. Por otro lado, cabe destacar que en estas pruebas, no sólo han fallado los biestables realimentados, sino también aquellos cercanos a estos. Este hecho muestra que tanto los biestables realimentados como sus vecinos tienden a presentar mayor debilidad que el resto de biestables de los registros.

Tabla 4.10.: Número de fallos insertados en Trivium Estándar en cada ciclo con tensión mínima.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2
1307	0	0	0	0	0	0	0	0	1	1
1312	0	0	0	1	2	1	2	2	2	2
1351	0	0	0	0	0	0	1	1	2	1
1352	0	0	0	0	0	0	0	0	1	1
1402	0	0	0	0	0	0	1	2	2	2
1403	0	0	0	0	1	0	1	1	3	3
1545	0	1	0	1	1	1	3	2	4	4
1546	0	0	1	0	1	0	3	2	3	3

Tabla 4.11.: Posiciones de los fallos insertados en Trivium Estándar en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2
1307	-	-	-	-	-	-	-	-	94	0
1312	-	-	-	0	0/93	0	0/93	0/93	0/93	0/93
1351	-	-	-	-	-	-	94	94	94/177	94
1352	-	-	-	-	-	-	-	-	0	0
1402	-	-	-	-	-	-	177	1/177	1/177	1/177
1403	-	-	-	-	93	-	93	93	0/93/178	0/93/178
1545	-	1	-	1	93	1	93/94/177	1/93	*	*
1546	-	-	94	-	94	-	0/93/94	0/94	0/94/178	0/94/178

Como puede observarse, mediante el uso de tensión mínima, el rango de variación del periodo de reloj necesario para inyectar fallos es más grande que en el caso de tensión nominal. Esto trae consigo que sea posible introducir fallos en un mayor número de ciclos de inserción, siendo estos en muchos casos inserciones efectivas de un solo error. Por otro lado, al igual que en el caso de tensión nominal, el comportamiento de ambos cifradores en paralelo es similar, es decir, tienden a fallar de igual forma frente a un mismo ataque.

Además, como puede observarse en las dos tablas anteriores, la combinación de tensión mínima de alimentación junto con el uso de inserciones de pulsos de reloj hace que la inyección de errores sea más fácil. Esto quiere decir que mediante dicha combinación es posible insertar los fallos efectivos en un mayor número de ciclos de reloj, presentando además vulnerabilidad en los biestables que anteriormente no fallaban, como son el biestable 177 y los vecinos a todas las realimentaciones. Teniendo en cuenta esto, a continuación se presenta un análisis de la dependencia de la inserción de fallos en función de la variación de la tensión de alimentación V_{cc}.

Resultados variando tensión de alimentación y periodo

Teniendo en cuenta que cuando se utiliza la tensión mínima como tensión de alimentación es más fácil la inyección de errores, se ha utilizado la herramienta *Shmoo Plot,* que ofrece el equipo de test, para poder estudiar la relación entre disminución de tensión e inserción de fallos. Gracias a esta herramienta se han realizado una serie de test modificando la tensión de alimentación y el periodo de reloj del pulso inducido.

En la Figura 4.20 puede observarse el número de fallos en el registro interno a medida que la tensión de alimentación va disminuyendo a la vez que aumenta el periodo de reloj del pulso utilizado. Como se puede observar, cuando se utiliza la tensión nominal junto con pequeños cambios en el periodo de reloj del pulso inducido, se producen grandes cambios en el número de fallos inyectados en los cifradores, es decir, se produce un mayor número de fallos para pequeñas variaciones. Sin embargo, este fenómeno cambia en función de la disminución de la tensión de alimentación desde la tensión nominal hasta la mínima, es decir, a medida que disminuye la tensión de alimentación, se puede variar el periodo de reloj del pulso en busca de fallos efectivos sin que esto produzca un mayor número de fallos en un mismo ciclo, obteniéndose por tanto un mayor rango para la inserción de fallos.

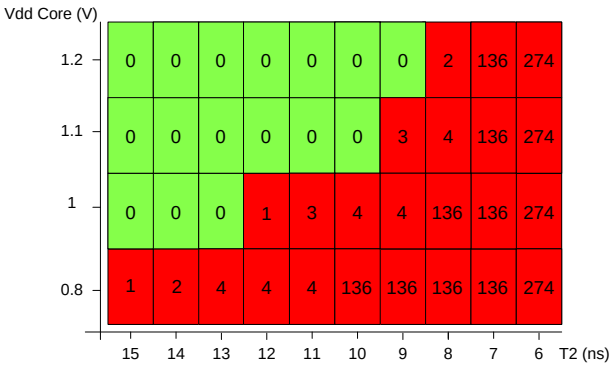

Figura 4.20.: Shmoo plot de Trivium Estándar modificando tensión de alimentación y periodo T_2.

Resultados para el caso especial señal de control

Para este caso, como se explicó anteriormente, es necesaria la utilización de la forma de onda representada en dos casos de la Figura 4.21. Para el primer caso, Caso a, se puede ver que si el valor de la variable T_3 es menor que 15 ns el sistema no insertará fallos en el registro interno del cifrador, se encuentra dentro del rango de periodo mínimo de funcionamiento estable. Por el contrario, en el Caso b, si la variable T_3 es mayor a 15 ns, el sistema comienza a inyectar errores ya que el cambio está muy próximo al flanco activo de reloj.

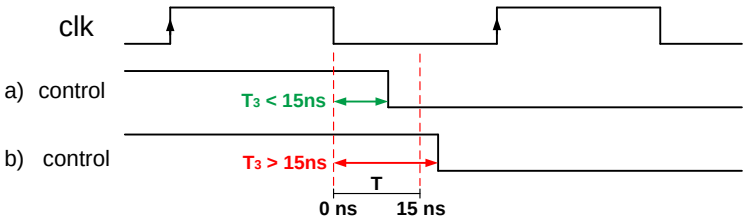

Figura 4.21.: Representación de los casos producidos por la manipulación de la señal de control.

La Tabla 4.12 muestra los valores de la variable T_3 empleados para la inyección de fallos. Como puede observarse, estos valores son muy cercanos a 15 ns, siendo la diferencia entre el mayor y el menor de tan solo 1 ns. Estos valores

Tabla 4.12.: Valor de variable T_3 para cada test de manipulación de señal de control.

	Test 1	Test 2	Test 3	Test 4	Test 5
Periodo (ns)	15.051	15.051	15.051	16.051	16.051

Tabla 4.13.: Número de fallos insertados en Trivium Estándar mediante manipulación señal de control.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2
1312	2	0	10	0	1	0	149	0	149	6
1403	19	0	2	0	14	0	132	9	132	1
1545	10	0	1	0	10	0	136	1	136	6

pueden llamar la atención debido a que se repiten para los distintos test, pero el hecho de su repetición, se debe a que durante las pruebas se observó que para un mismo valor, se obtenían resultados diferentes de una vez a otra. Es debido a esto por lo que los resultados presentados han sido realizados utilizando valores de T_3 iguales entre sí.

En la Tabla 4.13 se presentan los resultados de los fallos insertados en los cifradores mediante el uso de esta técnica. Como puede observarse, el ajuste, a la hora de inyectar un sólo error efectivo, no es tan preciso como en los casos de la modificación de la señal de reloj, dado que para un mismo periodo de reloj es posible insertar diferentes tipos de fallos. A pesar de esto, es posible la inducción de un sólo error efectivo en cada ciclo de reloj seleccionado.

En la Tabla 4.14 se muestran las posiciones de los biestables que han fallado cuando son uno o dos los que fallan. Como puede verse, además de continuar la tendencia de fallo de los biestables de realimentación o sus vecinos, se presenta la particularidad de que se producen fallos en biestables que no son los de realimentación. En estas pruebas se ha podido observar que los fallos se presentan en biestables localizados en el centro de los registros entre las realimentaciones, lo cual sugiere que las señales de control no se encuentran balanceadas para todos los componentes por igual. Además puede apreciarse una diferencia en cuanto al comportamiento de los dos cifradores implementados en paralelo frente a este tipo de ataques. Esto puede deberse a que el

routing de cada uno de los cifradores y por lo tanto el skew deribado de estos, sea diferente el uno del otro y por lo tanto esto puede influir en la inyección de errores mediante esta técnica.

Tabla 4.14.: Posición de los fallos insertados en Trivium Estándar mediante manipulación señal de control.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2
1312	13/17	-	*	-	17	-	*	-	*	*
1403	*	-	17/73	-	*	-	*	*	*	74
1545	*	-	17	-	*	-	*	0	*	*

4.4.2. Resultados obtenidos sobre Trivium Low Power

Pasamos ahora a mostrar los resultados obtenidos para la implementación del cifrador Trivium Low Power tras la aplicación de los ataques por inserción de pulsos en la línea de reloj. Del conjunto de datos, aquí aparecen los resultados de un solo cifrador debido a que en la implementación ASIC utilizada para llevar a cabo las pruebas, solo funcionaba correctamente uno de los dos cifradores implementados en paralelo. Esto no impidió realizar los test y obtener resultados generales, puesto que como se ha podido ver anteriormente para los cifradores estándar, si bien los comportamientos de las implementaciones en paralelo son diferentes, estos difieren en cuanto a los rangos de frecuencias necesarios para la inducción de errores pero no en el resultado final.

A lo largo de los siguientes subapartados se realizan, como en la subsección anterior, un análisis de los cambios de valor de los biestables de realimentación en varios ciclos de reloj, un análisis de los resultados de los ataques con tensión nominal y un análisis de los resultados obtenidos mediante la modificación del periodo de reloj del pulso y la tensión de alimentación del core del circuito. En el estudio de este cifrador no se han llevado a cabo test específicos bajo condiciones de tensión mínima, puesto qué como se explicó anteriormente, realizar todo el conjunto de pruebas sobre los cifradores, podría haber generado un extenso conjunto con datos bastante similares. A pesar de esto, se realizaron pruebas mediante el uso de la herramienta *Shmoo Plot* del equipo de test. Por

tanto, se decidió realizar pruebas preliminares de tensión mínima para observar si el comportamiento era similar al Trivium Estándar y centrarse solo en el escenario de tensión nominal de alimentación.

Transiciones biestables en varios ciclos de inserción

Para este cifrador, debido a que los registros de desplazamiento se dividen en dos (par e impar), las realimentaciones son dobles, es decir, en este cifrador se realimenta en los biestables 0, 1, 93, 94, 177 y 178. Por lo tanto, en la Tabla 4.15 se presentan las transiciones de valor de estos biestables en varios ciclos de reloj. Estos datos servirán para poder ver en las tablas posteriores, por qué en un ciclo determinado se produce un comportamiento u otro en cuanto a inyección de fallos.

Tabla 4.15.: Transición en los bits de realimentación por cada ciclo de inserción.

	Bits realimentación					
Ciclo	0	1	93	94	177	178
1307	Si	No	Si	No	No	No
1312	No	No	No	No	No	No
1351	Si	No	Si	No	No	No
1352	No	No	No	Si	No	Si
1402	No	No	No	No	No	No
1403	No	No	No	No	Si	No
1545	Si	No	Si	No	No	No
1546	No	Si	No	No	No	Si

Resultados con tensión nominal

Para llevar a cabo los test de inserción de fallos, se han utilizado los valores de periodo de reloj mostrados en la Tabla 4.16. Con estos valores se han obtenido los resultados mostrados en la Tabla 4.17. Como puede observarse, la tendencia a producirse un solo fallo, al igual que en el cifrador anterior, sigue apareciendo en función del periodo de reloj, produciéndose un aumento en el número de

errores a medida que se reduce el periodo de reloj del Test 1 al Test 5. Por otro lado, cuando se produce un gran número de fallos, este es menor que en el caso del cifrador estándar. Esto es debido a que si se fuerza en exceso el cifrador, tienden a fallar los biestables que tienen transición de estado de los registros pares o impares (en función del ciclo de inserción), los cuales son la mitad de los originales del Trivium estándar.

Tabla 4.16.: Periodo de reloj del ancho de pulso T_2 para cada test.

	Test 1	Test 2	Test 3	Test 4	Test 5
Periodo (ns)	8.91	8.60	7.3	7	6.93

Tabla 4.17.: Número de fallos insertados en Trivium LP en cada ciclo.

Ciclo	Test 1	Test 2	Test 3	Test 4	Test 5
1307	0	0	0	0	0
1312	0	0	0	60	60
1351	0	0	1	35	69
1352	0	0	0	0	69
1402	0	0	0	66	66
1403	0	0	0	0	76
1545	1	1	2	2	73
1546	0	1	1	1	1

En la Tabla 4.18 se presentan las posiciones específicas de los fallos insertados en el cifrador. Al igual que en el cifrador estándar, los biestables que tienden a fallar son los de realimentación, presentando una especial vulnerabilidad los biestables 0 y 93, los cuales se corresponden con los biestables realimentados de los registros impares.

Resultado variando la tensión de alimentación

En la Figura 4.22 puede observarse la evolución y comportamiento del cifrador en función de la tensión de alimentación y del valor del periodo del pulso de reloj empleado para la inserción de fallos. Como puede apreciarse, se produce una cierta incertidumbre en cuanto a la inserción de un solo error efectivo,

aumentando la probabilidad a medida que se disminuyen tanto el periodo de reloj como la tensión de alimentación. Esta figura muestra que a tensión nominal, la frecuencia necesaria para la inserción de errores debe de ser más alta que en los casos donde se disminuye la tensión de alimentación. Por otro lado, la diferencia de resultados entre pequeñas variaciones de T_2, muestran que para una misma situación de ataque (en cuanto a tensión), produce resultados diferentes, es decir, ataques efectivos o no efectivos.

Tabla 4.18.: Posiciones de los fallos insertados en Trivium LP en cada ciclo.

Ciclo	Test 1	Test 2	Test 3	Test 4	Test 5
1307	-	-	-	-	-
1312	-	-	-	*	*
1351	-	-	93	*	*
1352	-	-	-	-	*
1402	-	-	-	*	*
1403	-	-	-	-	*
1545	93	93	0/93	0/93	*
1546	-	93	93	93	93

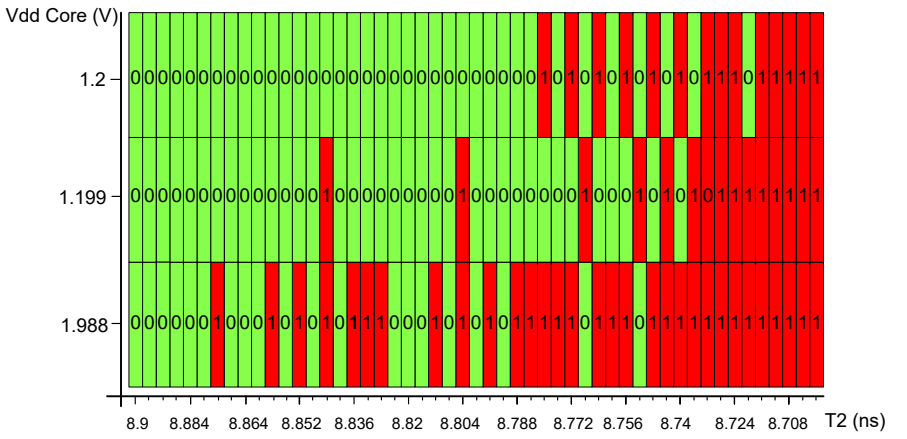

Figura 4.22.: Shmoo plot de Trivium LP modificando tensión de alimentación y periodo T_2

4.4.3. Resumen de resultados del resto de cifradores utilizados

A lo largo de este capítulo se han presentado los resultados para el cifrador Trivium estándar y Trivium Low Power. En el Apéndice A, se recogen el resto de los resultados obtenidos para los cifradores Trivium x2 estándar, Trivium x2 Low Power, Trivium x8 estándar, Trivium x8 Low Power, Trivium x16 estándar y Trivium x16 Low Power. De este conjunto de datos, a continuación se describen los aspectos más relevantes que se desprenden de los resultados obtenidos tras someter a cada uno de los cifradores al sistema de inserción de fallos por alteración de la señal de reloj con tensión nominal[3].

Como primer aspecto importante de estos resultados, hay que tener en cuenta que al tratarse de cifradores cuyos registros de estados internos han sido duplicados, el número de realimentaciones aumenta a medida que se duplican dichos registros. Esto trae consigo que se presenten un mayor número de posiciones vulnerables o puntos débiles en los registros internos de los cifradores cuando se realizan ataques por alteración de la señal de reloj. Teniendo en cuenta esto, se ha podido ver que para los ataques realizados en los ciclos de inserción seleccionados, es posible insertar fallos en diferentes posiciones de los registros de estados interno, habiendo conseguido inserciones de fallos efectivas de un solo biestable. Estas inserciones de fallos efectivas se consiguen al igual que en los casos presentados en este capítulo, es decir, siempre que se ajuste lo suficiente el periodo de reloj del pulso inducido en la señal de reloj. Este aspecto hay que tenerlo en cuenta debido a que estos cifradores poseen una estructura de reloj muy compleja y por tanto, al manipular la señal de reloj, se puede pasar fácilmente de inyectar un solo error efectivo a provocar el cambio en la gran mayoría de biestables. Sin embargo, al tener los cifradores dichas estructuras de reloj complejas, sus frecuencias máximas de funcionamiento son menores que en el caso del cifrador Trivium estándar (salvo el cifrador Trivium x2 estándar) y por tanto no es necesario utilizar periodos de reloj para los pulsos inducidos demasiado pequeños. Así, los periodos de reloj para inyectar fallos son menos exigentes, permitiendo utilizar un mayor rango de tiempos para la inserción de fallos.

De entre todos los cifradores cuyos resultados se presentan en el Apéndice A, es el cifrador Trivium x2 estándar el único que ha presentado un periodo de

[3]Como se explicó anteriormente, los ataques realizados a tensión nominal son el caso más desfavorable y por tanto si es posible la inserción de fallos en este escenario es extrapolable al caso donde la tensión de alimentación es mínima.

reloj para los pulsos similar al cifrador Trivium estándar, véase Tabla A.1. Los resultados de los experimentos realizados sobre este cifrador no han logrado insertar fallos efectivos en el interior de su registro interno. Para las pruebas realizadas se obtienen dos resultados posibles, por un lado, la imposibilidad de inyectar fallos efectivos de un solo biestable y, por otro lado, obtener múltiples inserciones de fallos en un mismo ataque, es decir, errores no efectivos. Estos resultados pueden verse en la Tabla A.3 donde, como puede observarse para los diferentes test, se pasa de no inyectar fallos a inyectar un gran número de ellos. Este número de fallos se corresponde con el total de transiciones posibles en dicho ciclo de reloj presentados en la Tabla A.2. Esto se debe a que este cifrador puede estar implementado de forma más balanceada en cuanto a sus señales de datos que los demás, haciendo que la inserción de fallos sea más difícil. Sin embargo, cabe destacar que los periodos de reloj utilizados pueden ser más ajustados, realizando pruebas más exhaustivas como por ejemplo en combinación con una tensión de alimentación mínima, y que esto lleve a la inducción de fallos efectivos en los registros internos. Teniendo en cuenta los resultados obtenidos, el cifrador Trivium x2 estándar ha sido el cifrador cuya resistencia a este tipo de ataques ha presentado ser mayor.

En cuanto al resto de cifradores, se puede observar en sus respectivas tablas que sus periodos de reloj de los pulsos necesarios para la inyección de fallos son mayores que los del cifrador Trivium estándar, siendo en los casos de los cifradores Low Power mayores que en sus versiones estándar. A pesar de esto, en todos estos cifradores ha sido posible inyectar un solo error efectivo en numerosos casos y por tanto presentan vulnerabilidades frente a este tipo de ataques.

4.5. Conclusiones

En este capítulo se ha llevado a cabo el análisis de la vulnerabilidad de cifradores Trivium implementados en ASIC frente a la inserción de fallos por la alteración de la señal de reloj y en menor medida, de señales de control. Para ello se ha hecho uso del equipo de test Agilent 93000 mediante el cual se han llevado a cabo las diferentes pruebas e implementaciones de ataques.

Estos ataques han consistido en la manipulación de la señal de reloj mediante la inducción de pulsos cortos, los cuales provocan un mal funcionamiento del

cifrador, traducido en inserciones de fallos en los registros internos de los cifradores bajo test. Además, se ha realizado otro ataque por inserción de fallos mediante el desfase temporal de la señal de control de los cifradores, consiguiendo de este modo una violación de tiempos de setup de los biestables que componen los registros internos de los cifradores.

Los ataques mediante la alteración de la señal de reloj se han llevado a cabo junto con la manipulación de las condiciones de funcionamiento de temperatura y tensión de alimentación. Las pruebas realizadas donde se manipulaba la temperatura, han demostrado que su alteración junto con la inducción de pulsos de reloj no presentan diferencias significativas respecto de los ataques realizados a temperatura nominal. Sin embargo, los ataques realizados junto con la manipulación de la tensión de alimentación han demostrado ser muy efectivos. Se ha podido comprobar que a medida que la tensión de alimentación disminuye, es posible insertar fallos efectivos de forma más fácil, es decir, utilizando periodos de reloj para el pulso inducidos mayores. Además, se ha podido observar que, alimentando con tensión nominal, pequeños cambios en el periodo de reloj del pulso inducido provocan un gran cambio en cuanto al número de fallos insertados, mientras que alimentando con tensión mínima es posible modificar el periodo de reloj del pulso notablemente sin que esto se traduzca en un gran número de inserciones de fallos.

Mediante el uso de los ataques diseñados, se ha comprobado que los cifradores Trivium implementados en ASIC presentan vulnerabilidades localizadas en los biestables realimentados de sus registros internos o sus celdas vecinas, donde ha sido posible insertar fallos efectivos de un solo biestable. Estas vulnerabilidades son dependientes del ciclo de inserción del pulso de reloj, dado que dependen de si en dicho ciclo de reloj se produce una transición, de uno a cero o de cero a uno, de sus biestables realimentados. Dichas vulnerabilidades son debidas al hecho de que los biestables realimentados son aquellos que presentan un mayor retraso en sus líneas de transmisión de datos.

En el caso donde se hace uso de la manipulación de la señal de control como forma de ataque, los biestables que tienden a fallar son siempre los de las mismas posiciones en el interior de los registros, siendo estos diferentes a los biestables realimentados. Esto demuestra que mediante el uso de esta técnica es posible insertar fallos efectivos de un solo biestable en posiciones que no se encuentran realimentadas, donde los retrasos en las líneas de transmisión de

datos son mayores que los de otros biestables del registro interno y donde las señales de control no se encuentran totalmente balanceadas.

Por otro lado, se ha comprobado que los dos cifradores idénticos implementados en paralelo para cada modelo de cifrador presentan un comportamiento similar frente a estos ataques, siendo posible inyectar fallos efectivos en ambos para cada modelo de cifrador, exceptuando el cifrador Trivium x2 estándar. Este caso en particular del Trivium x2 estándar ha presentado mayor robustez frente a estos ataques con las pruebas llevadas a cabo. Además, otro factor importante a tener en cuenta ha sido el aumento de vulnerabilidad de los modelos de los cifradores cuyos registros son divididos por diseño, aumentando sus realimentaciones y por tanto sus puntos débiles.

Por lo tanto, se puede concluir que, mediante el uso de las técnicas de inserción de fallos por alteración de la señal de reloj y por desfase temporal de la señal de control es posible inyectar errores de un solo fallo efectivo en los registros internos de los cifradores implementados en la tecnología ASIC. Además, se ha podido demostrar que estas técnicas en combinación con la modificación de la tensión de alimentación facilitan la inyección de fallos efectivos, permitiendo concluir que este tipo de cifradores es vulnerable mediante este tipo de ataques.

5. Criptoanálisis en cifradores Trivium con datos experimentales

5.1. Introducción

En este capítulo se presenta la realización experimental de un ataque DFA (*Differential Fault Analysis*) sobre implementaciones ASIC del cifrador Trivium. La consecución de este ataque conlleva la unión de varios elementos. Por un lado, el mecanismo experimental para la inserción de fallos y la captura de los bits del flujo cifrado necesarios. Por otro lado, un programa informático que a partir de los flujos cifrados sea capaz de obtener el mayor número de posiciones del registro interno del cifrador en un instante de la realización del ataque. Finalmente se necesita también un modelo de funcionamiento inverso del cifrador Trivium para obtener el estado interno inicial y así conocer la clave secreta y el IV. El montaje de todo este sistema requiere la combinación de herramientas de desarrollo de varios tipos, tales como programas de simulación VHDL, el entorno Matlab y el entorno de programación Eclipse.

Los puntos de partida de este capítulo son por una parte los análisis teóricos de vulnerabilidad del cifrador Trivium presentados en la literatura y recogidos en el Capítulo 2 de esta memoria y, por otra parte, el mecanismo de inserción de fallos desarrollado para implementaciones ASIC que se ha mostrado en el Capítulo 4 de esta memoria.

El capítulo se encuentra estructurado de la siguiente forma: en un primer apartado se describe el escenario general y los requisitos que se exigen para poder realizar el ataque DFA. A continuación se describe el sistema DFA desarrollado para, en el apartado siguiente, comprobar dicho sistema con datos obtenidos mediante simulación. Posteriormente se pasa a describir el plan de

ataque así como las modificaciones que han sido necesarias introducir a los ataques para conseguir que sea efectivo y explotable. Por último se presentan los resultados obtenidos y las conclusiones derivadas de este estudio.

El contenido de este capítulo no se muestra porque se encuentra publicado en C.V. 6.

5.2. Conclusiones

En este capítulo se ha presentado el sistema de ataque experimental mediante la manipulación de la señal de reloj y criptoanálisis sobre implementaciones ASIC del cifrador de flujo Trivium. Por un lado, se ha llevado a cabo la construcción del sistema completo para poder hacer uso del sistema DFA teórico presentado en [77, 78] junto con datos obtenidos experimentalmente. Además se ha diseñado un cifrador Trivium inverso el cual permite volver hacia atrás los ciclos necesarios para determinar la clave secreta una vez conocido el registro interno del cifrador atacado en cualquier instante.

Una vez construido el sistema se ha llevado a cabo una combinación entre un sistema para el análisis diferencial DFA teórico junto con ataques por simulaciones, para posteriormente utilizarlo con ataques experimentales a la implementación ASIC de dos cifradores Trivium. Utilizando simulaciones se ha podido ver que la ruptura del sistema de cifrado Trivium es posible a partir de 12 inyecciones de errores (utilizando fuerza bruta) o 16 usando únicamente inyecciones de fallos, mientras que haciendo uso de implementaciones reales es necesario el uso de unas 32 inyecciones de errores en total, siendo realmente útiles 22 inyecciones.

Por otro lado, se ha comprobado la imposibilidad de inyectar más de 3 errores en un mismo ciclo de reloj que permitan la ruptura del sistema dada la estructura del cifrador y su comportamiento frente a este tipo de ataques. Debido a esto, se ha presentado un nuevo método para la inserción de fallos que permite realizar las inyecciones necesarias aprovechando las características propias del cifrador. Con esta nueva visión de ataque se ha podido solventar la problemática en cuanto a las inyecciones de fallos en un mismo ciclo de reloj y se ha llevado a cabo el ataque de forma experimental, consiguiendo con ello el número de inyecciones necesarias para poder realizar el DFA.

Los ataques experimentales han demostrado que es posible recuperar el registro de estados interno en el instante $T_0 + 800$ tras los ataques y que por tanto el sistema es vulnerable. En cuanto al tiempo empleado para el ataque completo es de unas 6 horas, ocupando la práctica totalidad la fase del proceso experimental, la cual no está automatizada. Teniendo en cuenta que se podría automatizar, el realizar el ataque completo para obtener la clave secreta se reduciría a unos pocos minutos.

En definitiva, se ha podido demostrar que el cifrador de flujo Trivium implementado en ASIC es vulnerable a los ataques de inserción de fallos mediante la manipulación de la señal de reloj de forma experimental. La clave ha sido recuperada en el 100 % de los casos, tanto experimentalmente como en las simulaciones, siendo el coste de estos ataques mínimo en cuanto a tiempo requerido y recursos.

6. Conclusiones y líneas futuras

Los principales resultados obtenidos mediante la realización de la presente Tesis Doctoral pueden resumirse en los siguientes puntos:

En este trabajo de Tesis Doctoral se han realizado aportaciones en el campo de la seguridad de implementaciones hardware de criptosistemas con bajos recursos (lightweight cryptography), en particular sobre cifradores Trivium implementados tanto en FPGA como en ASIC. En concreto, se han desarrollado sistemas de ataques experimentales que permiten comprometer su seguridad y se ha montado un sistema de criptoanálisis sobre el cifrador Trivium que permite, a partir de los datos obtenidos de forma experimental, romper el cifrador recuperando su clave secreta. En lo que conocemos de la comunidad científica internacional, se trata de la primera vez que se ha logrado recuperar la clave secreta del cifrador Trivium de forma experimental.

Los trabajos realizados en la presente Tesis Doctoral han generado una serie de aportaciones a la criptografía de bajos recursos usando cifradores Trivium. De ellas se pueden destacar los principales resultados y conclusiones:

- Se ha realizado una amplia revisión del estado del arte de la criptografía de bajos recursos cuyo análisis ha permitido concretar las principales especificaciones del trabajo, como son:
    - Establecer el cifrador de flujo Trivium como criptosistema hardware objeto de la investigación. Las tecnologías de implementación utilizadas han sido tanto FPGA como ASIC, siendo las versiones del cifrador Trivium utilizadas la estándar, las multi-bit y las de bajo consumo.
    - Seleccionar como principal mecanismo de inserción de fallos la manipulación temporal de las entradas del cifrador (ataques del tipo activo no invasivo), combinándolo además con la manipulación de la tensión de alimentación y la temperatura.

  - Establecer como objetivo concreto de ataque la inserción de un solo fallo en el registro de estados interno del cifrador Trivium.
  - Seleccionar el sistema DFA (Differential Fault Analysis) de Michal y Bohuslav como fundamento teórico para el desarrollo de un sistema de criptoanálisis, dado que no existe ningún sistema de criptoanálisis experimental publicado para el cifrador Trivium.
- Se han desarrollado dos sistemas de ataque, uno para FPGA y otro para ASIC. Para la tecnología FPGA el mecanismo de ataque ha consistido en la manipulación de la señal de reloj mediante la inserción de pequeños pulsos. Para la tecnología ASIC, se ha desarrollado, además, otro mecanismo donde se manipulan las señales de control del cifrador, y se han combinado estos mecanismos con la manipulación de la tensión de alimentación y la temperatura.
  - En el caso de la tecnología FPGA el mecanismo de inserción de fallos se ha llevado a cabo de forma interna usando las capacidades del DCM (Digital Clock Manager). Se ha comprobado que para insertar un único fallo se necesita operar transitoriamente a una frecuencia mayor que la máxima del dispositivo: en la Spartan 3E, la frecuencia máxima del dispositivo es de 311 MHz y la utilizada para inyectar fallos es de 316 MHz, mientras que en el caso de la Spartan 6 son de 400 MHz y de 550 MHz, respectivamente.
  - En el caso de la tecnología ASIC, el mecanismo de inserción de fallos es externo y se ha utilizado el equipo de test Agilent 93000. En el caso de la manipulación de la señal de reloj, la inserción de un solo fallo requiere generar dos ciclos de frecuencias transitorias muy altas (superiores a la máxima del pad) y debiendo estar su valor extraordinariamente afinado. La frecuencia transitoria que hay que generar se reduce si se utiliza de forma combinada con una bajada de la tensión de alimentación, llegando a ser hasta la mitad de la frecuencia necesaria en condiciones de tensión nominal. En cambio, con la variación de la temperatura es inapreciable.
- En su conjunto las metodologías de ataques se han aplicado a toda la gama de cifradores Trivium: estándar, multi-bit y de bajo consumo. Los resultados generales más destacables son:
  - En todos los casos se ha podido insertar un solo fallo en una posición del estado interno del cifrador.

- La vulnerabilidad es independiente de la clave y del vector de inicialización con el que opere el cifrador Trivium.
- Los fallos inyectados en el cifrador se presentan en los biestables que reciben las realimentaciones entre los registros de desplazamientos o sus celdas vecinas consecutivas en el desplazamiento. Los cifradores idénticos implementados en paralelo y en iguales condiciones tecnológicas (salvo el posicionamiento y el rutado on-chip), que son atacados bajo unas mismas condiciones, aunque muestran comportamientos diferentes en cuanto a la celda concreta en la que se inserta el fallo, presentan siempre el mismo comportamiento general.

Por tanto, las técnicas de ataque aportadas en esta Tesis demuestran ser completamente eficaces a la vez que eficientes para insertar un solo fallo, aunque no es posible predeterminar con total seguridad la celda que fallará.

- Se ha desarrollado un sistema de criptoanálisis completo que, a partir del sistema DFA teórico presentado por Michal y Bohuslav, permite analizar los datos del flujo cifrado obtenidos experimentalmente y determinar el valor del registro interno del cifrador Trivium en un instante determinado. Este sistema permite recuperar el estado interno del cifrador con 22 inserciones de fallos. Las principales aportaciones de este trabajo que han hecho posible que el DFA teórico sea operativo experimentalmente son:
  - Desarrollar un método para poder insertar fallos efectivos en un mayor número de posiciones en el ciclo de reloj establecido para llevar a cabo el análisis diferencial. Este método consiste en insertar un error en un ciclo dado y desplazarlo varios ciclos de reloj para convertirlo en un error efectivo en el ciclo establecido para el análisis. Esta modificación permite obtener el número de fallos necesarios de forma experimental para poder aplicar el sistema DFA, siendo esencial para poder hacer operativo el DFA experimental.
  - Completar el software del DFA teórico dado que originalmente el código solo determinaba el número de valores del registro interno descubiertos, pero no devolvía su valor y posición. Además, se completó el código para que realizase el DFA a partir de datos tomados experimentalmente.

Este sistema experimental de DFA se ha aplicado sobre dos versiones diferentes del cifrador estándar implementado en ASIC. El análisis del flujo cifrado con este DFA permite conocer los 288 bits del estado interno en un instante concreto sin conocer a priori los valores de la clave ni del IV.

- Se ha diseñado un cifrador Trivium inverso mediante el cual, a partir de un estado interno conocido, es posible volver al estado inicial, donde se encuentran la clave secreta y el IV en las posiciones predeterminadas por diseño en el algoritmo. De esta forma, se determinan los datos secretos (clave e IV) completando así con éxito el ataque experimental al cifrador Trivium. Esto prueba por primera vez la vulnerabilidad experimental de las implementaciones físicas de este tipo de cifradores. Además, el sistema es robusto ya que se ha tenido éxito en el 100 % de las pruebas realizadas.

Como conclusión final se puede establecer que la combinación de un sistema de ataque no invasivo que permite insertar un solo error en el estado interno del cifrador Trivium, junto con el análisis mediante DFA del flujo cifrado correcto y erróneo y el uso de un Trivium inverso, forman un sistema que es capaz de romper experimentalmente la seguridad de este cifrador.

Resultados publicados

De los resultados obtenidos a partir del desarrollo de la presente Tesis Doctoral se han generado diferentes artículos que han sido publicados o que se encuentran en fase de revisión, los cuales se presentan a continuación:

- **C.V. 1:** F.E. Potestad-Ordóñez, C.J. Jiménez-Fernández and M. Valencia-Barrero, "*Vulnerability Analysis of Trivium FPGA Implementations*", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume: 25, Issue:12, pp. 3380-3389, December 2017, Print ISSN: 1063-8210, DOI: 10.1109/TVLSI.2017.2751151.
- **C.V. 2:** F.E. Potestad-Ordóñez, C.J. Jiménez-Fernández and M. Valencia-Barrero, "*Fault Injection on FPGA implementations of Trivium Stream Cipher using Clock Attacks*", Workshop on Trustworthy Manufacturing and Utilization of Secure Devices (Trudevice'16), 2016.

- **C.V. 3:** F.E. Potestad-Ordóñez, C. J. Jiménez-Fernández and M. Valencia-Barrero, "*Experimental and timing analysis comparison of FPGA trivium implementations and their vulnerability to clock fault injection*", Conference on Design of Circuits and Integrated Systems (DCIS'16), Granada, 2016, pp. 1-6. DOI: 10.1109/DCIS.2016.7845270.
- **C.V. 4:** F.E. Potestad-Ordóñez, C.J. Jiménez-Fernández and M. Valencia-Barrero, "*Fault Attack on FPGA implementations of Trivium Stream Cipher", IEEE International Symposium on Circuits and Systems* (ISCAS'16), pp. 562-565, 2016.
- **C.V. 5:** F.E. Potestad-Ordóñez, C.J. Jiménez-Fernández, C. Baena-Oliva, P. Parra-Fernández and M. Valencia-Barrero, "*Floorplanning as a practical countermeasure against clock fault attack in Trivium stream cipher*", Conference on Design of Circuits and Integrated Systems (DCIS'18), Lyon, 2018. Aceptado para su publicación.
- **C.V. 6:** F.E. Potestad-Ordóñez, C.J. Jiménez-Fernández and M. Valencia-Barrero, "*Breaking Trivium Stream Cipher implemented on ASIC using experimental attacks and DFA*", IEEE Transactions On Computers, 2019. Enviado y pendiente de aceptación.

Líneas de investigación futuras

Creemos que la Tesis presenta un avance en la investigación de la vulnerabilidad experimental del cifrador Trivium. No obstante podemos indicar algunas de las líneas de investigación que dan continuidad a este campo de investigación. Así, como líneas futuras se plantean las siguientes:

- Automatización del sistema de ataque para reducir los tiempos de ejecución y ataque y poder así estudiar la vulnerabilidad de los sistemas en un menor tiempo.
- Estudio y desarrollo de contramedidas que permitan evitar los ataques llevados a la práctica con éxito en esta Tesis Doctoral. Estas contramedidas deberán contemplar las diferentes técnicas de ataques activos no invasivos, como pueden ser la detección de fallos inyectados en el interior del registro mediante alguna técnica de redundancia (total o parcial), la utilización de una firma de clave, el uso de un detector de pulsos cortos

en la señal de reloj, el diseño adecuado de señales críticas o rutado y colocación de los componentes críticos de los cifradores.

- Aplicación de los sistemas de ataques diseñados a otros dispositivos criptográficos, como son los cifradores de bloque. Dada la versatilidad de los sistemas de ataque diseñados, estimamos que también permitirán estudiar la vulnerabilidad de otros dispositivos criptográficos. Así mismo estudiaremos si las contramedidas también ayudan a hacer más seguros dichos cifradores.

A. Resultados obtenidos para cada uno de los cifradores Trivium

En este apéndice se muestran los resultados obtenidos tras los ataques realizados para cada uno de los diseños del cifrador Trivium no presentados en el Capítulo 4. Se presentan en el orden: Trivium x2 Estándar, Trivium x2 Low Power, Trivium x8 Estándar, Trivium x8 Low Power, Trivium x16 Estándar y Trivium x16 Low Power. Estos resultados se estructuran de la siguiente manera:

1. Periodos de reloj utilizados para el pulso T_2 en cada test.
2. Número de transiciones de los biestables producidos en cada ciclo de inserción, tanto de cero a uno, como de uno a cero.
3. Número de fallos inyectados en cada ciclo de reloj seleccionado para los ataques.
4. Posiciones de los fallos inyectados en el registro interno de cada cifrador.

En las tablas donde se presenta el número de inserciones de fallos insertados en cada uno de los cifradores, Tabla A.3, Tabla A.7, Tabla A.11, Tabla A.15, Tabla A.19 y Tabla A.23, los casos donde aparece un cero son aquellos donde no se ha podido inyectar ningún fallo. Los casos donde aparecen un número superior a uno son aquellos donde se inyectaron más de un error y por tanto son inyecciones no efectivas. Por último, los casos donde se presenta un uno son aquellos donde se consiguió inyectar un solo error efectivo.

Por otro lado, en las tablas donde se presentan las posiciones de los fallos insertados, Tabla A.4, Tabla A.8, Tabla A.12, Tabla A.16, Tabla A.20 y Tabla A.24, los casos donde no se inyectaron errores aparecen representados con "-", mientras que los casos de inserción de fallos múltiples de más de cuatro fallos simultáneos aparecen representados por "*".

Teniendo en cuenta que cada diseño del cifrador es diferente en cuanto a su estructura interna, es decir, sus realimentaciones dependen de diferentes posiciones en el interior del registro y sus desplazamientos internos varían en función de un diseño a otro, los ciclos de inserción para cada cifrador serán diferentes. Debido a esto, se realizó un estudio de cuales son los ciclos de inserción óptimos para cada cifrador (los ciclos donde se producen las transiciones de valor de los biestables realimentados) con el objetivo de poder estudiar la vulnerabilidad de cada uno de estos puntos y sus celdas vecinas. Con ello se consigue que los ciclos de inserción utilizados contemplen todos los casos donde hay un cambio de valor en los biestables realimentados, que como se ha podido ver son los más débiles frente a los ataques estudiados. Por lo tanto, el valor de los ciclos de inserción se encuentra personalizado para cada uno de los diseños del cifrador.

A.1. Trivium x2 Estándar

Tabla A.1.: Periodos de reloj del ancho de pulso T_2 para cada test.

	Test 1	Test 2	Test 3	Test 4	Test 5
Periodo (ns)	6.990	6.970	6.950	6.940	6.930

Tabla A.2.: Número de transiciones de los biestables en cada ciclo de reloj.

Ciclo inserción	Total	$0 \rightarrow 1$	$1 \rightarrow 0$
1307	141	70	71
1312	146	74	72
1351	143	71	72
1352	139	70	69
1402	136	68	68
1403	136	68	68
1545	131	64	67
1546	136	67	69

Tabla A.3.: Número de fallos insertados en Trivium x2 en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2
1307	0	0	0	0	0	0	0	0	71	0
1312	0	0	0	0	146	146	146	146	146	146
1351	0	0	0	0	143	143	143	143	143	143
1352	0	0	0	0	0	0	0	0	0	0
1402	0	0	0	0	136	136	136	136	136	136
1403	0	0	0	0	136	136	136	136	136	136
1546	0	0	0	0	0	0	0	0	0	0
1550	0	0	0	0	69	0	69	136	32	136

Tabla A.4.: Posiciones de los fallos insertados en Trivium x2 en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2
1307	-	-	-	-	-	-	-	-	*	-
1312	-	-	-	-	*	*	*	*	*	*
1351	-	-	-	-	*	*	*	*	*	*
1352	-	-	-	-	-	-	-	-	-	-
1402	-	-	-	-	*	*	*	*	*	*
1403	-	-	-	-	*	*	*	*	*	*
1546	-	-	-	-	-	-	-	-	-	-
1550	-	-	-	-	*	-	*	*	*	*

A.2. Trivium x2 Low Power

Tabla A.5.: Periodos de reloj del ancho de pulso T_2 para cada test.

	Test 1	Test 2	Test 3	Test 4	Test 5
Periodo (ns)	8.85	8.8	8.5	8.2	7

Tabla A.6.: Número de transiciones de los biestables en cada ciclo de reloj.

Ciclo inserción	Total	$0 \rightarrow 1$	$1 \rightarrow 0$
1307	75	37	38
1315	73	36	37
1352	67	34	33
1354	70	36	34
1403	62	30	32
1406	75	37	38
1548	64	32	32
1553	77	39	38

Tabla A.7.: Número de fallos insertados en Trivium x2 LP en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2
1307	1	0	1	0	1	0	2	0	4	3
1315	1	0	1	0	1	0	2	1	74	74
1352	0	0	0	0	0	0	0	0	2	1
1354	0	0	0	0	1	0	1	0	35	2
1403	0	0	1	0	1	0	1	1	2	2
1406	0	0	0	0	0	0	0	0	2	4
1548	0	0	0	0	0	1	1	1	1	64
1553	0	0	0	0	0	0	0	0	2	2

Tabla A.8.: Posiciones de los fallos insertados en Trivium x2 LP en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2
1307	1	-	1	-	1	-	0/1	-	0/1/94/177	0/1/177
1315	1	-	1	-	1	-	0/1	0	*	*
1352	-	-	-	-	-	-	-	-	96/179	96
1354	-	-	-	-	3	-	3	-	*	3/95
1403	-	-	177	-	177	-	177	177	94/177	94/177
1406	-	-	-	-	-	-	-	-	3/96	2/3/96/179
1548	-	-	-	-	-	95	95	95	95	*
1553	-	-	-	-	-	-	-	-	93/178	93/178

A.3. Trivium x8 Estándar

Tabla A.9.: Periodos de reloj del ancho de pulso T_2 para cada test.

	Test 1	Test 2	Test 3	Test 4	Test 5
Periodo (ns)	8	7.7	7.5	7.1	6.96

Tabla A.10.: Número de transiciones de los biestables en cada ciclo de reloj.

Ciclo inserción	Total	$0 \rightarrow 1$	$1 \rightarrow 0$
1307	149	74	75
1315	150	75	75
1352	144	73	71
1354	144	70	74
1403	140	69	71
1406	135	70	65
1548	148	75	73
1553	140	70	70

Tabla A.11.: Número de fallos insertados en Trivium x8 en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2
1306	0	0	1	0	2	0	2	1	150	149
1312	0	0	0	0	0	0	2	1	2	1
1351	0	0	0	0	2	1	2	1	4	4
1353	1	0	1	0	1	0	1	2	1	3
1402	0	0	1	0	1	0	1	0	140	139
1403	0	0	0	0	0	0	3	1	3	2
1546	0	0	0	1	1	2	2	2	4	75
1550	1	0	1	0	2	0	2	0	139	140

Tabla A.12.: Posiciones de los fallos insertados en Trivium x8 en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2
1306	-	-	178	-	9/178	-	9/178	94	*	*
1312	-	-	-	-	-	-	93/97	97	93/97	97
1351	-	-	-	-	1/184	99	1/184	1	0/1/5/184	0/95/99/182
1353	183	-	183	-	183	-	183	100/181	183	94/100/181
1402	-	-	2	-	2	-	2	-	*	*
1403	-	-	-	-	-	-	1/3/183	98	1/3/183	3/98
1546	-	-	-	98	97	97/98	97/184	97/98	9/97/98/184	*
1550	178	-	178	-	5/178	-	5/178/183	-	*	*

A.4. Trivium x8 Low Power

Tabla A.13.: Periodos de reloj del ancho de pulso T_2 para cada test.

	Test 1	Test 2	Test 3	Test 4	Test 5
Periodo (ns)	13.9	13.7	13.5	13.3	13.1

Tabla A.14.: Número de transiciones de los biestables en cada ciclo de reloj.

Ciclo inserción	Total	$0 \rightarrow 1$	$1 \rightarrow 0$
1307	77	37	40
1315	67	34	33
1352	66	35	31
1354	73	36	37
1403	73	38	35
1406	62	31	31
1548	73	36	37
1553	73	38	35

Tabla A.15.: Número de fallos insertados en Trivium x8 LP en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2
1333	0	0	0	0	0	0	2	1	2	2
1472	1	0	2	0	2	1	2	4	6	5
1474	0	0	1	0	1	1	5	2	6	2
1475	0	0	0	0	0	0	0	0	2	1
1510	0	0	0	0	2	0	3	2	6	4
1511	0	0	2	0	3	0	4	1	6	3
1547	0	0	0	0	0	0	0	0	3	0
1550	0	0	0	0	0	0	1	0	2	1

Tabla A.16.: Posiciones de los fallos insertados en Trivium x8 LP en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2
1333	-	-	-	-	-	-	2/103	4	2/103	3/4
1472	1	-	1/2	-	1/2	4	1/2	2/4/8/9	*	*
1474	-	-	1	-	1	1	*	1/4	*	1/4
1475	-	-	-	-	-	-	-	-	97/178	4
1510	-	-	-	-	1/97	-	1/2/97	1/97	*	*
1511	-	-	0/2	-	0/1/2	-	0/1/2/179	1	*	0/1/2
1547	-	-	-	-	-	-	-	-	3/10/179	-
1550	-	-	-	-	-	-	0	-	0/178	0

A.5. Trivium x16 Estándar

Tabla A.17.: Periodos de reloj del ancho de pulso T_2 para cada test.

	Test 1	Test 2	Test 3	Test 4	Test 5
Periodo (ns)	9	8.7	8.5	8.1	7.51

Tabla A.18.: Número de transiciones de los biestables en cada ciclo de reloj.

Ciclo inserción	Total	$0 \rightarrow 1$	$1 \rightarrow 0$
1307	153	78	75
1315	140	72	68
1352	152	75	77
1354	149	75	74
1403	140	68	72
1406	141	72	69
1548	145	75	70
1553	139	66	73

Tabla A.19.: Número de fallos insertados en Trivium x16 en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2
1307	0	0	0	1	0	1	0	1	1	3
1312	1	0	2	0	2	0	2	0	3	2
1351	0	0	0	1	0	1	1	1	7	4
1352	0	0	0	0	1	0	3	0	4	4
1402	1	0	2	0	2	0	2	2	3	5
1403	0	1	0	1	1	1	2	2	5	3
1548	1	0	1	0	1	0	1	0	3	3
1583	0	0	0	1	0	1	0	2	2	2

Tabla A.20.: Posiciones de los fallos insertados en Trivium x16 en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Trv 2	Tr 1	Tr 2
1307	-	-	-	100	-	100	-	100	100	12/102/105
1312	104	-	104/107	-	104/107	-	104/107	-	98/104/107	101/108
1351	-	-	-	104	-	104	104	104	*	*
1352	-	-	-	-	101	-	101/103/108	-	*	*
1402	104	-	5/104	-	5/104	-	5/104	14/104	5/104/183	*
1403	-	101	-	101	120	101	101/120	2/101	*	2/15/101
1548	104	-	104	-	104	-	104	-	104/105/106	11/13/99
1583	-	-	-	104	-	104	-	14/104	10/105	14/104

A.6. Trivium x16 Low Power

Tabla A.21.: Periodos de reloj del ancho de pulso T_2 para cada test.

	Test 1	Test 2	Test 3	Test 4	Test 5
Periodo (ns)	16.5	15.8	15.5	15	14.8

Tabla A.22.: Número de transiciones de los biestables en cada ciclo de reloj.

Ciclo inserción	Total	$0 \rightarrow 1$	$1 \rightarrow 0$
1307	73	35	38
1315	74	35	39
1352	75	38	37
1354	80	37	43
1403	66	37	29
1406	63	30	33
1548	70	36	34
1553	75	42	33

Tabla A.23.: Número de fallos insertados en Trivium x16 LP en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2	Trv 1	Trv 2
1356	0	0	0	0	0	2	2	2	5	3
1371	0	1	0	1	0	1	1	4	1	4
1389	0	0	0	1	0	1	0	3	1	7
1428	0	0	1	0	3	1	4	3	7	3
1444	0	0	0	1	0	1	2	1	3	2
1453	0	0	0	0	0	0	0	4	1	5
1456	0	0	0	0	0	1	0	1	2	2
1553	0	0	0	0	0	0	0	2	0	2

Tabla A.24.: Posiciones de los fallos insertados en Trivium x16 LP en cada ciclo.

	Test 1		Test 2		Test 3		Test 4		Test 5	
Ciclo	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2	Tr 1	Tr 2
1356	-	-	-	-	-	103/104	1/4	103/104	*	99/103/104
1371	-	17	-	17	-	17	17	*	17	*
1389	-	-	-	17	-	17	-	17/19/117	203	*
1428	-	-	1	-	1/5/6	1	*	1/6/104	*	1/6/104
1444	-	-	-	17	-	17	4/107	17	4/106/107	17
1453	-	-	-	-	-	-	-	*	203	*
1456	-	-	-	-	-	1	-	1	1/5	1/97
1553	-	-	-	-	-	-	-	19/25	-	19/25

Bibliografía

[1] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," *International Conference for Internet Technology and Secured Transactions (ICITST'15)*, pp. 336–341, 2015.

[2] T. Xu, J. B. Wendt, and M. Potkonjak, "Security of IoT systems: Design challenges and opportunities," 2014, pp. 417–423.

[3] L. H. Encinas, *La criptografía.* Editorial CSIC Consejo Superior de Investigaciones Científicas, 2016.

[4] W. Stallings, L. Brown, M. D. Bauer, and A. K. Bhattacharjee, *Computer security: principles and practice.* Pearson Education, 2012.

[5] M. E. Hellman, "An overview of public key cryptography," *IEEE Communications Magazine*, vol. 40, no. 5, pp. 42–49, 2002.

[6] N. F. PUB, "46-3. data encryption standard," *Federal Information Processing Standards, National Bureau of Standards, US Department of Commerce*, 1977.

[7] W. C. Barker, "Recommendation for Triple Data Encryption Algorithm (TDEA) Block Cipher," *National Institute of Standards and Technology NIST special publication*, vol. 800, p. 67, 2008.

[8] S. Heron, "Advanced encryption standard (AES)," *Network Security*, vol. 2009, no. 12, pp. 8–12, 2009.

[9] A. P. Jian Guo, Thomas Peyrin and M. Robshaw, "The LED Block Cipher," *Cryptographic Hardware and Embedded Systems (CHES 2011)*, vol. 6917 of LNCS, pp. 326–341, 2011.

[10] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," in *Design Automation Conference (DAC'15) 52nd ACM/EDAC/IEEE.* IEEE, 2015, pp. 1–6.

[11] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An ultra-lightweight block cipher," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2007, pp. 450–466.

[12] R. A. Rueppel, "Stream ciphers," in *Analysis and Design of Stream Ciphers*. Springer, 1986, pp. 5–16.

[13] A. F. Sabater, L. H. Encinas, A. M. Muñoz, F. M. Vitini, and J. M. Masqué, *Criptografía, protección de datos y aplicaciones: una guía para estudiantes y profesionales*. Ra-Ma, 2012.

[14] S. W. Golomb, *Shift register sequences*. Aegean Park Press, 1967.

[15] M. Robshaw and O. Billet, *New stream cipher designs: the eSTREAM finalists*. Springer, 2008, vol. 4986.

[16] ECRYPT, "The eSTREAM Project." [Online]. Available: http://www.ecrypt.eu.org/stream/project.html

[17] M. Hell, T. Johansson, and W. Meier, "Grain: a stream cipher for constrained environments," *International Journal of Wireless and Mobile Computing*, vol. 2, no. 1, pp. 86–93, 2007.

[18] S. Babbage and M. Dodd, "The MICKEY stream ciphers," in *New Stream Cipher Designs*. Springer, 2008, pp. 191–209.

[19] C. D. Cannière, "Trivium: A stream cipher construction inspired by block cipher design principles," *Information Security*, pp. 171–186, 2006.

[20] C. De Canniere, A. Biryukov, and B. Preneel, "An introduction to block cipher cryptanalysis," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 346–356, 2006.

[21] K. A. McKay, K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, *Report on lightweight cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2017.

[22] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems." Springer, 1997, pp. 513–525.

[23] S. Maitra, A. Siddhanti, and S. Sarkar, "A differential fault attack on PLANTLET," *IEEE Transactions on Computers*, vol. 66, no. 10, pp. 1804–1808, 2017.

[24] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks: Revealing the secrets of smart cards*. Springer Science & Business Media, 2008, vol. 31.

[25] O. Kömmerling and M. G. Kuhn, "Design principles for tamper-resistant smartcard processors," *Smartcard*, vol. 99, pp. 9–20, 1999.

[26] R. J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 2001.

[27] S. P. Skorobogatov, "Semi-invasive attacks: a new approach to hardware security analysis," Ph.D. dissertation, Citeseer, 2005.

[28] D. Samyde, S. Skorobogatov, R. Anderson, and J.-J. Quisquater, "On a new way to read data from memory," in *Proceedings Security in Storage Workshop*. IEEE, 2002, pp. 65–69.

[29] S. P. Skorobogatov and R. J. Anderson, "Optical fault induction attacks," in *International workshop on cryptographic hardware and embedded systems(CHES'02)*, vol. 2523. Springer, 2002, pp. 2–12.

[30] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Annual International Cryptology Conference*. Springer, 1996, pp. 104–113.

[31] P. Kocher, J. Jaffe, and B. Jun, "Introduction to differential power analysis and related attacks," 1998.

[32] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Cryptology Conference*. Springer, 1999, pp. 388–397.

[33] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2004, pp. 16–29.

[34] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2001, pp. 251–261.

[35] J.-J. Quisquater and D. Samyde, "Electromagnetic analysis (ema): Measures and counter-measures for smart cards," in *Smart Card Programming and Security*. Springer, 2001, pp. 200–210.

[36] J. Schmidt and M. Hutter, "Optical and em fault-attacks on crt-based rsa: Concrete results," *Proceedings of the Austrochip*, 2007.

[37] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The Sorcerer's Apprentice Guide to Fault Attacks," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 370–382, 2006.

[38] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache, "Fault Injection Attacks on Cryptographic Devices : Theory, Practice, and Countermeasures," *Proceedings of the IEEE*, vol. 100, no. 11, pp. 3053–3076, 2012.

[39] C. Giraud and H. Thiebeauld, "A survey on fault attacks," *Card Research and Advanced Applications VI*, pp. 159–176, 2004.

[40] S. Govindavajhala and A. W. Appel, "Using Memory Errors to Attack a Virtual Machine," *Proc. the IEEE Symp. Security and Privacy (SP)*, pp. 154–156, 2003.

[41] A. Barenghi, G. Bertoni, L. Breveglieri, M. Pellicioli, and G. Pelosi, "Low voltage fault attacks to AES," *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'10)*, pp. 7–12, 2010.

[42] P. Street and W. Lafayette, "Low Cost Attacks on Tamper Resistant Devices," *Security Protocols*, pp. 125–136, 1998.

[43] T. Fukunaga and J. Takahashi, "Practical Fault Attack on a Cryptographic LSI with ISO / IEC 18033-3 Block Ciphers," *Workshop on Fault Diagnosis and Tolerance in Criptography*, pp. 84–92, 2009.

[44] M. Agoyan and J. Dutertre, "When clocks fail: On critical paths and clock faults," *Smart Card Research and Advanced Application*, pp. 182–193, 2010.

[45] Y. Ren, A. Wang, and L. Wu, "Transient-Steady Effect Attack on Block Ciphers," *Cryptographic Hardware and Embedded Systems, Springer*, vol. 2, pp. 433–450, 2015.

[46] N. F. Ghalaty, B. Yuce, M. Taha, P. Schaumont, and V. Tech, "Differential Fault Intensity Analysis," *Workshop on Fault Diagnosis and Tolerance in Criptography*, pp. 49–58, 2014.

[47] N. F. Ghalaty, B. Yuce, and P. Schaumont, "Differential fault intensity analysis on PRESENT and LED block ciphers," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 2015, pp. 174–188.

[48] A. Kaminsky, M. Kurdziel, and S. Radziszowski, "An overview of cryptanalysis research for the Advanced Encryption Standard," in *Military*

Communications Conference (MILCOM'2010). IEEE, 2010, pp. 1310–1316.

[49] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of CRYPTOLOGY*, vol. 4, no. 1, pp. 3–72, 1990.

[50] E. Biham and A. Shamir, "Differential cryptanalysis of Feal and N-hash," *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 1–16, 1991.

[51] A. Shimizu and S. Miyaguchi, "Fast data encipherment algorithm FEAL," in *Workshop on the Theory and Application of of Cryptographic Techniques.* Springer, 1987, pp. 267–278.

[52] E. Biham and A. Shamir, "Differential cryptanalysis of the full 16-round DES," in *Differential Cryptanalysis of the Data Encryption Standard.* Springer, 1993, pp. 79–88.

[53] E. Biham and A. Shamir, "Differential cryptanalysis of the Data Encryption Standard," *Springer*, 1997.

[54] D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the importance of checking cryptographic protocols for faults," in *International conference on the theory and applications of cryptographic techniques (EUROCRYPT'97).* Springer, 1997, pp. 37–51.

[55] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[56] R. Anderson and M. Kuhn, "Low cost attacks on tamper resistant devices," in *International Workshop on Security Protocols.* Springer, 1997, pp. 125–136.

[57] J. J. Hoch and A. Shamir, "Fault Analysis of Stream Ciphers," *International association for Cryptologic Research*, pp. 240–253, 2004.

[58] R. Rivest, "The RC4 Encryption Algorithm, RSA Data Security," *Inc.*, 1992.

[59] E. Dawson, A. Clark, J. Golic, W. Millan, L. Penna, and L. Simpson, "The LILI-128 keystream generator," in *Proceedings of first NESSIE Workshop*, 2000.

[60] P. Hawkes and G. Rose, "Primitive specification and supporting documentation for SOBER-t32 submission to NESSIE," in *Proceedings of the first open NESSIE workshop*, 2000, pp. 13–14.

[61] P. Dusart, "Differential Fault Analysis on A.E.S." *Applied Cryptography and Network Security*, pp. 293–306, 2003.

[62] C. Giraud, "DFA on AES," *Lecture Notes in Computer Science*, vol. 3373, pp. 27–41, 2005.

[63] C. H. Kim, "Differential fault analysis against AES-192 and AES-256 with minimal faults," in *Fault Diagnosis and Tolerance in Cryptography (FDTC'10)*. IEEE, 2010, pp. 3–9.

[64] G. Wang and S. Wang, "Differential Fault Analysis on PRESENT Key Schedule," *2010 International Conference on Computational Intelligence and Security*, pp. 362–366, 2010.

[65] N. Bagheri, R. Ebrahimpour, and N. Ghaedi, "New differential fault analysis on PRESENT," *EURASIP Journal on Advances in Signal Processing*, pp. 1–10, 2013.

[66] K. Jeong and C. Lee, "Differential fault analysis on block cipher LED-64," in *Future Information Technology, Application, and Service*. Springer, 2012, pp. 747–755.

[67] H. Tupsamudre, S. Bisht, and D. Mukhopadhyay, "Differential fault analysis on the families of SIMON and SPECK ciphers," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2014 Workshop on*. IEEE, 2014, pp. 40–48.

[68] H. Alkhzaimi and M. M. Lauridsen, "Cryptanalysis of the SIMON Family of Block Ciphers," *IACR Cryptology ePrint Archive*, vol. 2013, p. 543, 2013.

[69] E. Tena-Sánchez and A. J. Acosta, "Dpa vulnerability analysis on trivium stream cipher using an optimized power model," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 1846–1849.

[70] P. Luo, Y. Fei, L. Zhang, and A. A. Ding, "Differential fault analysis of SHA3-224 and SHA3-256," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2016 Workshop on*. IEEE, 2016, pp. 4–15.

[71] E. Biham and O. Dunkelman, "Differential Cryptanalysis in Stream Ciphers," *IACR Cryptology ePrint Archive*, vol. 2007, p. 218, 2007.

[72] S. Karmakar and D. Chowdhury, "Differential Fault Analysis of MICKEY-128 2.0," *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC'13)*, pp. 52–59, 2013.

[73] S. Karmakar and D. Chowdhury, "Differential Fault Analysis of MICKEY Family of Stream Ciphers." *IACR Cryptology ePrint Archive*, 2014.

[74] S. Banik, S. Maitra, and S. Sarkar, "A differential fault attack on the Grain family of stream ciphers," *Cryptographic Hardware and Embedded Systems (CHES'12)*, pp. 122–139, 2012.

[75] S. Sarkar, S. Banik, and S. Maitra, "Differential Fault Attack against Grain Family with Very Few Faults and Minimal Assumptions," *IEEE Transactions on Computers*, vol. 64, no. 6, pp. 1647–1657, 2015.

[76] Y. Watanabe, Y. Todo, and M. Morii, "New Conditional Differential Cryptanalysis for NLFSR-based Stream Ciphers and Application to Grain v1," in *11th Asia Joint Conference on Information Security (AsiaJCIS'16)*. IEEE, 2016, pp. 115–123.

[77] M. Hojsík and B. Rudolf, "Differential Fault Analysis of Trivium," *Fast Software Encryption, Springer*, pp. 158–172, 2008.

[78] M. Hojsík and B. Rudolf, "Floating Fault Analysis of Trivium," *Progress in Cryptology, Springer*, pp. 239–250, 2008.

[79] Y. Hu, J. Gao, Q. Liu, and Y. Zhang, "Fault analysis of Trivium," *Designs, Codes and Cryptography, Springer*, vol. 62, no. 3, pp. 289–311, 2012.

[80] M. Saied, E. Mohamed, and S. Bulygin, "Improved Differential Fault Analysis of Trivium," *COSADE'11*, pp. 147–158, 2011.

[81] P. Dey and A. A. B, "Improved Multi-Bit Differential Fault Analysis of Trivium," *Progress in Cryptology-INDOCRYPT, Springer*, vol. 48, no. 2, pp. 37–52, 2014.

[82] M. Saied, E. Mohamed, and J. Buchmann, "Mutant Differential Fault Analysis of Trivium MDFA," *Information Security and Cryptology-ICISC'15*, pp. 433–446, 2015.

[83] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita, "Camellia: A 128-bit block cipher suitable for multiple

platforms design and analysis," in *International Workshop on Selected Areas in Cryptography*. Springer, 2000, pp. 39–56.

[84] C. M. Adams, "Constructing symmetric ciphers using the CAST design procedure," *Designs, Codes and cryptography*, vol. 12, no. 3, pp. 283–316, 1997.

[85] J. Lee, J. Park, S. Lee, and J. Kim, "The SEED encryption algorithm," *Encryption Algorithm*, 2005. [Online]. Available: http://tools.ietf.org/html/rfc4269

[86] M. Matsui, "New block encryption algorithm MISTY," in *International Workshop on Fast Software Encryption*. Springer, 1997, pp. 54–68.

[87] G. Piret and J.-J. Quisquater, "A differential fault attack technique against SPN structures, with application to the AES and KHAZAD," in *International Workshop on Cryptographic Hardware and Embedded Systems (CHES'03)*. Springer, 2003, pp. 77–88.

[88] M. Tunstall, D. Mukhopadhyay, and S. Ali, "Differential fault analysis of the Advanced Encryption Standard using a single fault," *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication*, pp. 224–233, 2011.

[89] Francisco Eugenio Potestad Ordóñez, "Sistema para ataques a cifradores Triviums implementados en FPGAs," *Trabajo Final de Grado, Universidad de Sevilla*, 2018.

[90] Francisco Eugenio Potestad Ordóñez, "Análisis de vulnerabilidad de cifradores de flujo en FPGAs," *Trabajo Final de Máster, Universidad de Sevilla*, 2014.

Notación

I_s	Registro de estados interno del cifrador.
Is_{t0}	Registro de estados interno sin fallo insertado en instante inicial.
Is'_{t0}	Registro de estados interno con fallo insertado en instante inicial.
t_0	Instante inicial del ataque o ciclo seleccionado para el ataque.
T_3	Variable temporal para manipulación de señales de control.
T_2	Variable temporal para manipulación de señales de reloj.
z_i	Flujo cifrado.
$z_{i(i=1)}^{\infty}$	Flujo cifrado correcto.
$z'^{\infty}_{i(i=1)}$	Flujo cifrado incorrecto.

Nomenclatura

AES	Advanced Encryption Standard
ASIC	Application-Specific Integrated Circuits
BUFGMUX	Multiplexor especial para señales de reloj
CAST	Carlisle Adams and Stafford Tavares
CITIES	Circuitos integrados para transmisión de información especialmente segura
Delay	Retardo
DES	Data Encryption Standard
DFA	Differential Fault Analysis
DFIA	Differential Fault Intensity Analysis
DIB	Device Interface Board
DPA	Differential Power Analysis
ECRYPT	European Network of Excellence in Cryptology
eSTREAM	European Stream Cipher Project
Flip-Flop	Biestable
FPGA	Field Programmable Gate Array
GND	Ground
HW	Hardware

IO	Input/Output
IoT	Internet of Things
ISE	Integrated Synthesis Environment
IV	Initialization Vector
K	Key
Key Stream	Flujo cifrado
LED	Light Encryption Device
LFSR	Linear Feedback Shift Register
LP	Low Power
LUT	Look Up Table
MB	MegaByte
MDFA	Mutant Differential Fault Analysis
MHz	Megahercios
MICKEY	Mutual Irregular Clocking KEY stream generator
MISTY	Mitsubishi Improved Security Technology
NIST	National Institute of Standards and Technology
NLFSR	Non Linear Feedback Shift Register
ns	Nanosegundo
PLL	Phase Locked-Loop
RC4	Rivest Cipher 4
Routing	Rutado
RSA	Rivest-Shamir-Adleman

SASEBO	Side-channel Attack Standard Evaluation BOard
SAT	Satisfiability
SOBER-t32	Seventeen Octet Byte Enabled Register 32-bit words
SOC	System On Chip
SW	Software
TDES	Triple Data Encryption Standard
TSE	Transient-Steady Effect attack
VHDL	VHSIC Hardware Description Language

Índice de figuras

Índice de tablas