TESIS DE LA UNIVERSIDAD DE ZARAGOZA
2024
272

Guillermo Díez Señoráns

Nuevas estrategias de diseño FPGA para la implementacion de funciones no-clonables fisicamente en ecosistemas IoT

Director/es
Celma Pueyo, Santiago
García Bosque, Miguel

ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

Universidad de Zaragoza
Servicio de Publicaciones

ISSN 2254-7606

Universidad Zaragoza
1542

Tesis Doctoral

NUEVAS ESTRATEGIAS DE DISEÑO FPGA PARA LA IMPLEMENTACION DE FUNCIONES NO-CLONABLES FISICAMENTE EN ECOSISTEMAS IOT

Autor

Guillermo Díez Señoráns

Director/es

Celma Pueyo, Santiago
García Bosque, Miguel

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

Programa de Doctorado en Física

2024

Nuevas estrategias de diseño FPGA para la implementación de funciones no-clonables físicamente en ecosistemas IoT

Guillermo Díez Señorans

Diciembre de 2023

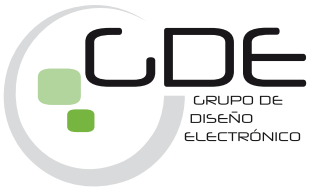
Nuevas estrategias de diseño FPGA para la implementación de funciones no-clonables físicamente en ecosistemas IoT

Tesis presentada a la Universidad de Zaragoza para
optar al grado de Doctor en Física por
D. Guillermo Díez Señorans

bajo la dirección de los doctores
Dr. Santiago Celma Pueyo
y
Dr. Miguel García Bosque

Departamento de Ingeniería Electrónica y Comunicaciones
Instituto Universitario de Investigación en Ingeniería de Aragón
Grupo de Diseño Electrónico

Universidad Zaragoza



Diciembre de 2023

A mi padre
y la memoria de mi madre.

Agradecimientos

Querría dar las gracias en primer lugar a mis directores, Dr. Santiago Celma Pueyo y Dr. Miguel García Bosque, por su guía a lo largo de estos años, sin los cuales esta tesis doctoral hubiera sido imposible. Igualmente indispensable ha sido el apoyo de los doctores Dr. Carlos Sánchez Azqueta y Dr. Francisco Aznar Tabuenca, así como del resto de miembros del Grupo de Diseño Electrónico de la Universidad de Zaragoza, Óscar, Nicolás, Belén, Concha, Pedro y Pepe . Tampoco puedo dejar de agradecer a mis compañeros de despacho, Abel, Guillermo, Diego, Uxua, Antonio, Jorge(s) y Raúl, el haber hecho siempre más ameno y en ocasiones incluso fácil este doctorado.

De forma especial, agradezco a Laura, a mi padre José Blas y a mi hermana Aurora la paciencia y el cariño con el que me han esperado en el mundo exterior que —creo recordar— hay más allá de los muros de la facultad.

Por último, agradezco también a todas las instituciones que han financiado la investigación recogida en este trabajo: Agencia Estatal de Investigación, Fondo Europeo de Desarrollo Regional a través de los proyectos TEC2017-85867-R, RTC2019-007039-7 y PID2020-114110RA-I00; Gobierno de Aragón a través de la beca para la contratación de personal investigador predoctoral en formación 2018-2022.

Listado de figuras

Listado de tablas

Listado de acrónimos

ADC Analog-Digital Converter.
AES Advanced Encryption Standard.
ASCII American Standard Code for Information Interchange.
ASIC Application Specific Integrated Circuit.
AXI Advanced Extensible Interface.
BEL Basic Element.
biw Buffer-in Width.
bow Buffer-out Width.
CCA Chosen-ciphertext Attack.
CIA Central Intelligence Agency.
CLB Configurable Logic Block.
CMOS Complementary Metal-Oxide-Semiconductor.
COTS Commercial Off-the-shelf.
CPA Chosen-plaintext Attack.
CPU Central Processing Unit.
CRP Challenge-Response pair.
DAC Digital-Analog Converter.
DES Data Encription Standard.
DPA Differential Power Analysis.
DSA Digital Signature Algorithm.
dw Data Width.
ECC Error Correcting Code.
EDA Electronics Design Automation.
EER Equal Error Rate.
FAR False Acceptance Rate.
FIA Fault Injection Attack.
FIFO First-In-First-Out.
FPGA Field Programmable Gate Array.
FRR False Rejection Rate.
GARO-PUF Galois Ring Oscillator PUF.
GC&CS Government Code and Cipher School.
GPU Graphic Processing Unit.
HD Hamming Distance.
hd Helper Data.
HDL Hardware Description Language.
IoT Internet of Things.
KCA Known-ciphertext Attack.
KPA Known-plaintext Attack.
LC Logic Cell.
LFSR Linear Feedback Shift Register.
LUT Look-up Tables.
MAC Message Authentication Code.
MEMS Micro-Electro-Mechanical System.
MITM Man-In-The-Middle.
MOSFET Metal-Oxide-Semiconductor Field Effect Transistor.
NBS National Bureau of Standards.
NIST National Institute of Standards and Technology.
NRE Nonrecurring Engineering Costs.
NVM Non-volatile Memory.
OTP One-time Password.
PDL Programmable Delay Lines.
PL Programmable Logic.
PMU Power Management Unit.
POK Physically Obfuscated Keys.
PS Processing System.
PUF Physically Unclonable Function.
RO-PUF Ring Oscillator PUF.

ROC Receiver-Operating Characteristic.

RoT Root of Trust.

RRAM Resistive Random Access Memory.

RTN Random Telegraph Noise.

SCA Side Channel Attack.

SDK Software Development Kit.

SoC System-on-Chip.

SRAM Static Random Access Memory.

STT-MRAM Spin Torque Transfer Magnetic Random Access Memory.

TEMPEST Telecommunication Electronic Material Protected from Emanating Spurious Transmissions.

TRNG True Random Number Generator.

UART Universal Asynchronous Receiver-Transmitter.

Listado de símbolos

Prob Probabilidad.

$*$ Operador AND lógico.

$R(L)$ Autocorrelación de una secuencia de bits a longitud L.

$R(0)$ Sesgo de una secuencia de bits.

$\mathcal{O}$ "O grande" de Landau.

$\mathrm{Bin}_{n,p}$ Distribución binomial de n número de ensayos y sesgo p.

$\mathrm{bit}(i,j)$ Bit extraído mediante la medida compensada de las celdas físicas "i" y "j".

I Cantidad de información.

θ Condición ambiental (abstracta) en la que se evalúa una función física.

$\mathcal{C}$ Coste de una topología PUF.

Decod Función de decodificación binaria.

$\equiv$ Definición.

ϱ Densidad de entropía por celda.

$\varrho^{\mathbf{m}}$ Densidad de minentropía por celda.

dist Distancia métrica.

D_{KS} Distancia de Kolmogorov-Smirnov.

Encod Función de codificación binaria.

H Entropía.

ECC Aplicación de corrección de errores.

ξ Estímulo físico de una función física.

FA Evento de que se produzca una falsa aceptación en un proceso de autenticación.

FR Evento de que se produzca un falso rechazo en un proceso de autenticación.

FAR Tasa de falsa aceptación en función del umbral de identificación.

$\mathcal{F}$ Función física.

Firma Aplicación de firma digital.

Flip Evento de que se produzca la inversión de un bit al repetir sucesivamente dos medidas de una misma instancia PUF.

ν Frecuencia de oscilación.

$\nu^{\mathbf{ref}}$ Frecuencia de referencia (reloj).

ν_s Frecuencia de muestreo de un oscilador de Galois.

FRR Tasa de falso rechazo en función del umbral de identificación.

Enroll Función que extrae el identificador de referencia de una PUF.

HD Distancia de Hamming.

Hash Función hash.

$\mathbf{hd}_{ij}^{\mathcal{P}}$ Síndrome de la i-ésima instancia PUF, j-ésimo reto.

λ Índice de fabricación de una función física.

$\mathrm{HD}_{ii'jk}^{\mathbf{inter}}$ Inter-distancia de Hamming entre las instancias i, i' para el j-ésimo reto y la k-ésima repetición.

$\mathrm{HD}_{ijkk'}^{\mathbf{intra}}$ Intra-distancia de Hamming entre las repeticiones k, k' para la

i-ésima instancia y el j-ésimo reto.

$\mathrm{HD}^{\textbf{intra-env}}_{ijkk'll'}$ Intra-distancia entre las respuestas obtenidas en diferentes condiciones ambientales l, l'.

MAC Función MAC.

$\mathcal{C}^{\mathbf{m}}$ Mincoste de una topología PUF.

$H^{\mathbf{m}}$ Minentropía.

$\mu^{\textbf{inter}}$ Inter-distancia media.

$\mu^{\textbf{intra}}$ Intra-distancia media.

N_b Número de bits de un vector binario.

$N^{\textbf{inst}}$ Número de instancias que participan en un experimento PUF.

$\mathrm{Norm}_{\mu,\sigma}$ Densidad de probabilidad normal de media μ y desviación σ.

$N^{\textbf{rep}}$ Número de repeticiones de cada medida en un experimento PUF.

$N^{\textbf{retos}}$ Número de retos que utilizados en un experimento PUF.

$p^{\textbf{inter}}$ Probabilidad de que la inter-distancia de dos bits valga "1".

$p^{\textbf{intra}}$ Probabilidad de que la intra-distancia de dos bits valga "1".

$\times$ Producto cartesiano.

$\mathcal{P}$ Función no-clonable físicamente.

DAC Interfaz decodificadora de una PUF.

ADC Interfaz codificadora de una PUF.

$\vec{y}^{\mathcal{P}}_{ij,\textbf{ref}}$ Identificador de referencia de una PUF para la i-ésima instancia, j-ésimo reto.

$\vec{y}^{\mathcal{P}}_{ijk}$ Respuesta binariade una PUF para la i-ésima instancia, j-ésimo reto, k-ésima repetición.

$\vec{y}^{\mathcal{P}}_{ijkl}$ Respuesta de la PUF $\mathcal{P}$ para la i-ésima instancia, j-ésimo reto, k-ésima repetición, l-ésima condición ambiental.

$q^{\textbf{inst}}$ Probabilidad de que el bit nominal de una respuesta PUF valga "1".

$q^{\textbf{rep}}$ Probabilidad de que un bit se invierta al repetir una medida PUF.

$\delta\tilde{\nu}^{\textbf{min}}$ Resolución de una medición de frecuencia.

ψ Respuesta física de una función física.

$\boldsymbol{\Xi}$ Espacio de estímulos físicos de una función física.

$\boldsymbol{\Lambda}$ Espacio de índices de fabricación de una función física.

$\mathbf{D}^{\textbf{inter}}$ Conjunto de inter-distancias medidas en un determinado experimento PUF.

$\mathbf{D}^{\textbf{intra}}$ Conjunto de intra-distancias medidas en un determinado experimento PUF.

$\mathbf{D}^{\textbf{intra-env}}$ Conjunto de intra-distancias medidas en un determinado experimento PUF ampliado para contener medidas ambientales.

$\mathbf{Y}^{\mathcal{P}}$ Conjunto de respuestas medidas durante la evaluación experimental de la PUF $\mathcal{P}$.

$\mathbf{Y}^{\mathcal{P}*}$ Experimento PUF ampliado para incluir medidas en diferentes condiciones ambientales.

$\boldsymbol{\Psi}$ Espacio de respuestas físicas de una función física.

h Tasa de entropía por bit.

$h^{\mathbf{m}}$ Tasa de minentropía por bit.

$\mathcal{T}$ Topología abstracta de una PUF de medida compensada.

$u^{\mathbf{EER}}$ Distancia umbral de identificación correspondiente a una "tasa de igual error" para una PUF.

$u^{\mathbf{id}}$ Distancia umbral de identificación para una PUF.

VA Evento de que se produzca una verdadera aceptación en un proceso de autenticación.

VR Evento de que se produzca un verdadero rechazo en un proceso de autenticación.

Verif Función de verificación.

$\oplus$ Operador XOR lógico.

$\mathcal{N}^2$ Topología completa de una PUF de medida compensada (todas las comparaciones).

$\mathcal{N}^2_{/K}$ Topología modular en grupos de K celdas.

$\mathcal{N}_{-1}$ Topología de una PUF de medida compensada repitiendo una celda en cada comparación.

$\mathcal{N}_{/2}$ Topología *one-out-of-2* de una PUF de medida compensada (sin repetición de celdas en cada comparación).

Índice general

1 Introducción

En este capítulo introductorio se proporciona una visión general de la criptografía en su estado actual a través de su evolución histórica (sección 1.1), enfatizando el papel activo de la capa física sobre la que se sustenta la tecnología de las telecomunicaciones a la hora de proporcionar seguridad. En la sección 1.2 se discuten las particularidades de los sistemas digitales embebidos de bajo rendimiento con conexión a la red (Internet de las Cosas) desde el punto de vista de la seguridad de la información. Así mismo, se exponen los objetivos generales y específicos de esta tesis doctoral. Finalmente, en la sección 1.3 se describe la organización de esta memoria.

1.1. Breve historia de la criptografía

1.1.1. Criptografía clásica

La criptografía es hoy una ciencia formal bien establecida cuyo objeto de estudio es la seguridad de la información. Esta es un *desiderata* de los sistemas de comunicación, en el cual el contenido de un mensaje transmitido a través de un cierto canal se mantiene en secreto para entidades ajenas a la comunicación y, paralelamente, cada una de las instancias participantes en dicho proceso disponen de garantías sobre la identidad de las demás. Sin embargo, el establecimiento de la criptografía como ciencia formal fue precedido de siglos durante los cuales el envío eficaz de mensajes secretos era considerado un arte exclusivo del ámbito militar y, aún antes, en los albores del lenguaje escrito, un simple juego artístico con fines meramente decorativos. En este sentido, la evidencia más antigua de un trabajo sistemático de sustitución de símbolos escritos se encuentra en la tumba del noble egipcio Jnumhotep II, la cual data de aproximadamente el 1900 a.C. Se cree que estas sustituciones no buscaban la transmisión de información secreta, sino que tenían un sentido decorativo y místico-religioso. Se pueden encontrar aplicaciones lúdicas de sistemas primitivos de sustitución similares en otras civilizaciones incipientes;

por ejemplo, la mesopotámica o hebrea. En el caso de esta última, resulta llamativa la aparición del criptograma *Sheshach* para referirse a la ciudad de Babilonia[1] en el Libro de Jeremías —uno de los 46 que componen el Antiguo Testamento— generado mediante un método de sustitución denominado "atbash", y que consiste en reemplazar cada carácter de una palabra por su imagen especular respecto de la mitad del alfabeto hebreo (*i.e.*, primero por último, segundo por penúltimo, etcétera). La primera referencia bibliográfica de que se tiene constancia en la que se hace referencia explícita al secreto como una propiedad de la comunicación escrita es el mito de Belerofonte, narrado en la Ilíada de Homero: mientras el héroe griego era huésped en la corte del rey Preto de Tirinto, la esposa de este, despechada, urde con el monarca un plan para asesinar a Belerofonte sin faltar a la hospitalidad debida. Para ello, envía al héroe a su suegro, el rey Yóbates de Licia, junto con una supuesta carta de recomendación que en realidad esconde un mensaje secreto manifestando la intención homicida de hija y yerno. El rey licio trata de satisfacer la petición encomendando a Belerofonte la suicida misión de dar muerte a la Quimera, un monstruo híbrido de león y serpiente que aterrorizaba la región. No obstante, este regresa victorioso y el propio Yóbates, admirado, le entrega la mano de su hija en matrimonio. A pesar de la naturaleza evidentemente poética de la Ilíada, sí que existen referencias históricas del envío de comunicaciones secretas en la Antigüedad clásica, por ejemplo en la recopilación de los "Nueve Libros de Historia" de Heródoto de Halicarnaso, considerado el padre de la historiografía occidental, quien en el siglo V a.C documentó de forma exhaustiva los conflictos entre persas y griegos en el contexto de las guerras médicas. En su obra, Heródoto describe un método esteganográfico[2] por el cual Demarato, antiguo rey de Esparta exiliado en Persia, habría protagonizado en el 484 a.C. uno de los primeros episodios de espionaje militar de la historia occidental, al tener conocimiento de la inminente marcha del ejército persa a las órdenes del rey Jerjes sobre Grecia; Demarato comunicó los planes persas a Leónidas, rey de la ciudad griega de Esparta, a través de un mensaje oculto en una tablilla de cera —material de escritura habitual en la época—, de la cual había removido la cera, grabado el mensaje sobre el soporte de madera, y depositado de nuevo la cera encima. Esta tablilla aparentemente inocente llegó a las manos de Gorgo, esposa de Leónidas, quien descubrió el mensaje secreto, permitiendo movilizar una fuerza griega que, tras las batallas de Las Termópilas (480 a.C), Salamina (480 a.C) y Platea (479 a.C), terminaron con las aspiraciones expansionistas persas sobre el occidente. Además, se sabe que en este siglo los

[1] Sustituyendo las grafías hebreas *bet*, *bet*, *lamed* por *shin*, *shin*, *kaph*: **bbl** → **ŸŸr**

[2] La esteganografía es un conjunto de técnicas dirigidas a esconder mensajes secretos dentro de objetos cotidianos, de forma que la información contenida en ellos pase desapercibida. Se distingue de la criptografía en que permite ocultar el propio establecimiento del hecho comunicativo, además del contenido de la información.

Fig. 1.1.: Escítala, un primitivo sistema de criptografía por transposición descubierto en la Grecia clásica.

espartanos ya utilizaban un precoz sistema normalizado de criptografía para las comunicaciones secretas militares denominado "escítala" (figura 1.1). Este consistía en un cilindro de madera sobre el cual se enrollaba una cinta de cuero o pergamino y se escribía un mensaje. La cinta se transmitía desenrollada mostrando una sucesión ininteligible de letras, que sólo cobraba sentido al enrollarse de nuevo en un cilindro de diámetro idéntico al original. De acuerdo con la taxonomía moderna de la criptografía, este método se clasificaría como cifrado por transposición (*i.e.*, el texto cifrado se construye a partir del mensaje mediante una permutación de sus letras), cuya clave secreta es el diámetro de la escítala. También en la Edad Antigua, el historiador romano Cayo Suetonio, biógrafo de Julio César, describe un método de cifrado utilizado por este para enviar comunicaciones de índole militar a sus generales, el cual consistía en la sustitución de cada carácter en un mensaje por la letra que se encuentra tres posiciones por delante en el alfabeto con condiciones periódicas (esto es, considerando que la primera letra del alfabeto es precedida de la última). Este sistema de cifrado por sustitución, aunque naif para la criptología moderna, hubiera resultado eficaz en el contexto histórico de la Roma clásica, y ha perdurado hasta nuestros días la nomenclatura de "cifrado de César" para referirse de forma general a los algoritmos criptográficos de sustitución por desplazamiento [1], [2].

La Edad Media supuso un estancamiento para la evolución de la criptografía (o, al menos, para el registro documental de su uso, que es escaso), con la notable excepción del egipcio al-Qalqashandi, quien en el año 1412 completó el *Subh al-a 'sha*, una recopilación enciclopédica de 14 tomos que serviría como manual administrativo para los funcionarios estatales del Sultanato Mameluco de Egipto. Esta obra es una enorme compilación del saber medieval árabe, abarcando materias como historia, política, matemáticas o astronomía, e incluyendo un capítulo "Sobre la ocultación de mensajes secretos dentro de las cartas", donde se hace la primera mención explícita al criptoanálisis como una disciplina relacionada con, pero distinta

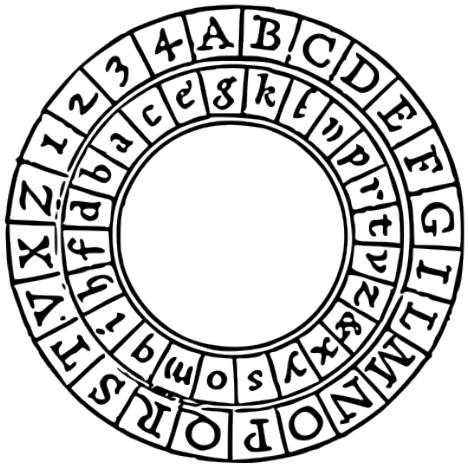


Fig. 1.2.: Disco de Alberti. Primer cifrador polialfabético, en el que la rotación del disco interior permite definir un alfabeto de sustitución diferente.

de, la criptografía. Durante el siglo XV, la contribución occidental más significativa a la criptografía es el trabajo del genovés Leon Battista Alberti, quien en torno a 1466 escribe un tratado sobre criptografía titulado *De cifris* donde recopila el criptoanálisis frecuencial (que probablemente hubiese sido descubierto con anterioridad), basado en deducir el alfabeto de sustitución en un "cifrado de César" mediante el análisis de las frecuencias de aparición de cada símbolo, poniendo estas en relación con las frecuencias de aparición conocidas de cada letra en el lenguaje natural en que está redactado el mensaje sin encriptar. Además, Alberti propuso una nueva técnica criptográfica robusta frente a este criptoanálisis frecuencial valiéndose del "disco de Alberti" (figura 1.2). Este dispositivo permitiría aplicar una cifra de sustitución diferente para cada carácter rotando el disco interior respecto del exterior, destruyendo así el patrón de frecuencias de un lenguaje natural. De este modo, Alberti había descubierto el hoy llamado "cifrado polialfabético". Pese a todo, hubo que esperar hasta finales del siglo XV y la llegada del Renacimiento para que resurgiera el interés por las comunicaciones secretas, en un contexto de globalización incipiente propiciada por el progreso en la tecnología del transporte marítimo, la enorme expansión del comercio y, con ello, el papel renovado de la diplomacia y las relaciones exteriores entre estados. Los siglos XVI y XVII fueron testigos de notables avances científicos e intelectuales de los que también participó la criptografía, y la historiografía de la época ya describe como prácticas habituales el cifrado de comunicaciones entre embajadas, así como el reclutamiento y mantenimiento de personal dedicado tanto a la encriptación de mensajes como a la resolución de criptogramas interceptados a potencias rivales. Desde el siglo XV hasta mediados del XIX, esta encriptación de

mensajes se realizaba por medio de cifras de sustitución homofónicas[3], junto con un código secreto utilizado para codificar algunas palabras específicas. El cifrado basado en claves secretas no se descubriría hasta los trabajos del italiano Giovan Battista Bellaso en 1553 (erróneamente atribuido al matemático francés Blaise de Vigenère), y aún entonces resultaba de poca utilidad práctica ya que el proceso algorítmico de cifrado había de realizarse a mano mediante tablas o dispositivos como el disco de Alberti, lo que aumentaba enormemente la probabilidad de cometer errores catastróficos que volvieran el mensaje ilegible. En su lugar, los códigos secretos se reproducían en listas llamadas "nomenclátor", donde se relacionaba cada letra con su grupo de sustitución, y un pequeño conjunto de palabras con su correspondiente código; por ejemplo, durante Las Guerras de Religión de Francia del siglo XVII, Juan de Moreo, representante de Felipe II en la Liga Católica, se comunicaba con La Corona Española utilizando un nomenclátor que incluía una lista de 413 términos codificados en grupos de dos o tres letras, *e.g.*, *España* $\rightarrow$ *LO*, *Navarra* $\rightarrow$ *PUL*, o *Rey de España* $\rightarrow$ *POM*. En 1628, el matemático francés Antoine Rossignol obtuvo el favor del Cardenal Richelieu, primer ministro de Francia y valido del rey Luis XIII, al resolver en pocas horas comunicaciones encriptadas de los rebeldes hugonotes durante los asedios a las villas sublevadas de Réalmont y La Rochelle. La información interceptada resultó crítica para la victoria de la facción real, y Rossignol pasó inmediatamente al servicio de la corona francesa. Rossignol perfeccionó los métodos criptológicos de su época, sin embargo su mayor contribución a la historia de la criptografía y la seguridad de la información no fue técnica sino administrativa, ya que fundó el "Gabinete del Secreto de Correos" (*Cabinet du Secret des Postes*), más conocido como "Gabinete Negro" (*Cabinet Noir*), precursor de los servicios de inteligencia normalizados propios del estado moderno (figura 1.3). A principios del siglo XVIII, los "Gabinetes Negros" a imagen del francés eran organismos estatales comunes en todos los estados europeos, destacando por su eficiencia y productividad el gabinete secreto de Viena, que se convirtió en una de las principales fuentes de inteligencia para la estrategia geopolítica de Austria hasta mediados del siglo XIX. En este momento, la crisis que arrastraba el absolutismo como modelo de gobierno precipitó el fin de muchas estructuras represivas del estado, entre ellas los gabinetes secretos dedicados a la intervención de correo postal. En la década de 1840, estas oficinas ya habían desaparecido en la mayor parte de Europa, incluyendo Francia, Austria e Inglaterra [2], [3].

[3] El cifrado homofónico es la sustitución de letras por grupos de símbolos (*i.e.* transformación suprayectiva de un alfabeto natural en un alfabeto de símbolos), de tal modo que el tamaño del grupo asociado a cada letra es proporcional a la frecuencia de aparición de dicha letra en el lenguaje natural en que se redacta el mensaje plano. De este modo, se evita el criptoanálisis frecuencial de Alberti.

Fig. 1.3.: Grabado anónimo francés de 1815, donde se caricaturiza el *cabinet noir*. Procedente del Museo Carnavalet de París.

En el año 1844, el estadounidense Samuel Morse inventa el primer telégrafo práctico, revolucionando la historia de las telecomunicaciones y, con ello, el envío de información de forma segura. Este nuevo invento se expande con rapidez por América y Europa, y sólo seis años después de su aparición ya es utilizado en la totalidad de América del Norte e Inglaterra. En cuanto al modo radical en que transformó las comunicaciones seguras, esto se debe fundamentalmente a la velocidad con la que el telégrafo permitía corregir un mensaje cifrado defectuoso, lo cual eliminaba el principal escollo para la criptografía basada en cifradores tal y como había sido propuesta por Alberti y Battiste más de dos siglos antes. Esto significó el abandono del nomenclátor en favor de los cifrados polialfabéticos tipo Vigenère, que a mediados del siglo XIX eran considerados indescifrables. Esta ilusión habría de terminar en 1864, cuando el comandante retirado del ejército prusiano Friedrick Kasiski publicó el ensayo *Die Geheimschriften und die Dechiffrer-Kunst*, donde presenta un método criptoanalítico general contra el cifrado de sustitución periódico polialfabético. Este hecho marcó el inicio de la carrera frenética entre propuestas criptográficas y contrapropuestas criptológicas —o viceversa— que caracteriza la relación entre ambas disciplinas a día de hoy. La obra más representativa del conocimiento criptográfico en este siglo XIX es el tratado *La Cryptographie Militaire*, escrita por el matemático y criptógrafo Holandés Auguste Kerchoffs para la revista francesa "*Le journal des Sciences Militaires*" mientras trabajaba como profesor de alemán en

París, y que colocó a Francia a la vanguardia de la técnica criptográfica en puertas del siglo XX.

El 28 de junio de 1914 el archiduque Francisco Fernando de Austria, heredero al trono Austro-Húngaro, es asesinado en las calles de Sarajevo por un nacionalista serbio. Este hecho precipitó que Austria-Hungría declarara la guerra a Serbia el 28 de julio, lo cual hizo caer toda una compleja red de alianzas tejida a lo largo del siglo XIX entre las principales potencias europeas con el fin de contener mutuamente su expansión colonial y su poderío militar: Rusia comienza a movilizarse inmediatamente en apoyo de Serbia y en contra de Austria-Hungría, desairando a Alemania, que declara la guerra a Rusia el 31 de julio, y a Francia y Bélgica el 3 de agosto, a cuyo auxilio acude el Imperio Británico (actual Reino Unido junto con Canadá, Australia y la India Británica), declarando la guerra a Alemania el 4 de agosto, dando así comienzo a la Primera Guerra Mundial. La primera acción hostil británica contra Alemania es el seccionamiento de los cables de telecomunicaciones submarinos del Canal de la Mancha utilizando el buque cablero *C.S. Alert*, forzando a Alemania a utilizar señales de radio y líneas de telégrafo enemigas para su comunicación exterior, lo cual suponía entregar completamente la confidencialidad de sus comunicaciones a un buen sistema criptográfico. En octubre de 1914, el oficial británico Alfred Ewing funda la sección de criptoanálisis del Almirantazgo Británico (conocida popularmente como "Habitación 40"), con el fin de extraer inteligencia de la interceptación de mensajes cifrados alemanes durante la guerra. El logro más notable de esta institución fue la resolución del "telegrama Zimmermann" (figura 1.4), cuando en enero de 1917 los criptólogos William Montgomery y Nigel de Grey resolvieron un telegrama cifrado remitido por el ministro de asuntos exteriores alemán, Arthur Zimmermann, y dirigido a la embajada alemana en México, el cual había sido interceptado por la inteligencia británica. El contenido del telegrama era una propuesta para que el gobierno mexicano atacara a los Estados Unidos de América (EUA) y retomara las regiones de Texas, Nuevo México y Arizona, arrebatadas a México en 1848, en caso de que EUA entrara en la guerra en contra de Alemania. El contenido sibilino del telegrama causó una gran indignación tanto en el gobierno como en la opinión pública estadounidense, y se considera el *casus belli* que precipitó la entrada de Estados Unidos en la Primera Guerra Mundial, marcando con ello un punto de inflexión en el devenir de la contienda a favor de la alianza franco-británica [2].

El estallido de la guerra había renovado el interés por las comunicaciones secretas, y la compañía estadounidense *American Telephone and Telegraph* (*AT&T*) comenzó a colaborar con el ejército de los EUA en un proyecto secreto para estudiar

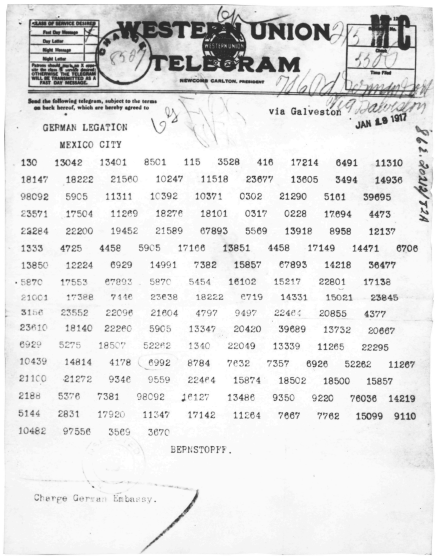

WESTERN UNION
TELEGRAM
via Galveston
JAN 19 1917
GERMAN LEGATION
MEXICO CITY
130 13042 13401 8501 115 3528 416 17214 6491 11310
18147 18222 21560 10247 11518 23677 13605 3494 14936
98092 5905 11311 10392 10371 0302 21290 5161 39695
23571 17504 11269 18276 18101 0317 0228 17694 4473
23284 22200 19452 21589 67893 5569 13918 8958 12137
1333 4725 4458 5905 17166 13851 4458 17149 14471 6706
13850 12224 6929 14991 7382 15857 67893 14218 36477
5870 17553 67893 5870 5454 16102 15217 22801 17138
21001 17388 7446 23638 18222 6719 14331 15021 23845
3186 23552 22096 21604 4797 9497 22464 20855 4377
23610 18140 22260 5905 13347 20420 39689 13732 20667
6929 5275 18507 52262 1340 22049 13339 11265 22295
10439 14814 4178 6992 8784 7632 7357 6926 52262 11267
21100 21272 9346 9559 22464 15874 18502 18500 15857
2188 5376 7381 98092 16127 13486 9350 9220 76036 14219
5144 2831 17920 11347 17142 11264 7667 7762 15099 9110
10482 97556 3569 3670
BERNSTORFF.
Chrge German Embassy.

Fig. 1.4.: Telegrama Zimmermann (*U.S. National Archives Catalog*).

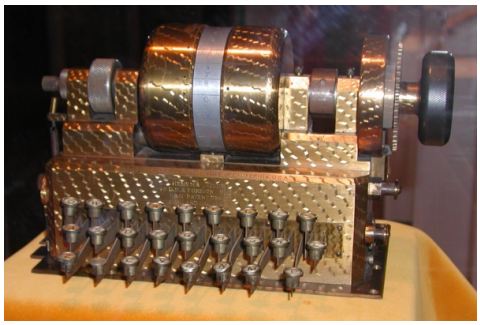
la seguridad de las comunicaciones telegráficas. En 1917, Gilbert Vernam, uno de los ingenieros trabajando para la *AT&T*, propuso una manera de encriptar la información que circulaba a lo largo de la línea de telégrafo codificada mediante código Baudot[4]. El "cifrado de Vernam" consistía en combinar ("sumar") cada símbolo de entrada con un símbolo marcado sobre una cinta perforada que actuaría como clave secreta. Esta clave debía tener la misma longitud que el mensaje a transmitir, y la combinación debía ser reversible para el receptor del mensaje siempre y cuando dispusiera de la misma clave utilizada para el cifrado. La operación propuesta por Vernam fue la siguiente:

Entrada	Clave	Entrada $\oplus$ Clave $=$ Cifrado
■	■	□
■	□	■
□	■	■
□	□	□

Donde "□" representa un espacio y "■" un punto en código Baudot. De este modo, el bit cifrado se obtiene a partir del bit de entrada y de la clave mediante la

[4]Sistema de codificación propuesto por el ingeniero francés Émile Baudot en 1874, el cual utiliza sucesiones de cinco símbolos binarios ("bits") para representar cada una de las letras del alfabeto, los números del cero al nueve, y algunos caracteres de escape (espacios, saltos de línea, etcétera). Es un precursor del Código Estándar Estadounidense para el Intercambio de Información (*American Standard Code for Information Interchange*, ASCII) moderno.

(a)



(b)

Fig. 1.5.: (a) Máquina de rotores de Hebern en el *National Cryptologic Museum* de EUA. (b) Reproducción en detalle del sistema de rotores en una máquina Enigma.

operación lógica XOR ($\oplus$), que resulta trivial de revertir ya que esta operación es involutiva (*i.e.*, coincide con su inversa), Entrada $=$ Cifrado $\oplus$ Clave. Vernam acababa de descubrir el "cifrado de flujo", en contraposición al clásico "cifrado de bloque", donde se utiliza una clave secreta para encriptar todo un bloque de información. Posteriormente, un comandante del Cuerpo de Señales del Ejército de EUA, Joseph Mauborgne, propuso que los símbolos de las claves utilizadas fueran completamente aleatorios y de un solo uso, evitando toda estructura a lo largo de la cinta perforada. Este cifrador fue llamado "*one-time pad*", y en 1918 se conjeturaba que era indescifrable[5]; lo cual en efecto se probó una intuición acertada, sin embargo, no hubo una demostración matemática rigurosa hasta los trabajos de Claude Shannon en 1949.

En 1919, una vez terminada la Primera Guerra Mundial, la "Habitación 40" fue integrada con la unidad de inteligencia militar británica para formar la Escuela Gubernamental de Codificación y Cifrado (*Government Code and Cipher School*, GC&CS). En torno a la misma fecha apareció la máquina de rotores, un dispositivo electromecánico que actuaba como un cifrador de flujo, transformando los caracteres introducidos por un operador a través de un teclado en otros diferentes en función de la posición angular de una serie de rotores internos que cambiaban de posición cada vez que se pulsaba una nueva tecla. Esto produce un cifrado polialfabético en el que la probabilidad de que se utilice un mismo alfabeto más de una vez se puede hacer arbitrariamente pequeña mediante la incorporación de un número sucesivamente mayor de rotores, inutilizando el criptoanálisis polialfabético general

[5]En el ámbito de la criptología, "indescifrable", "irrompible" o "imposible de descifrar" significa que no existe un método para encontrar la solución a un criptograma más rápido que probando exhaustivamente todas las claves secretas posibles hasta encontrar aquella que resuelve el texto cifrado.

de tipo Kasiski. Sin embargo, la operación de la máquina era determinista dada una posición inicial de sus rotores, lo cual actuaba como clave secreta y permitía al receptor del mensaje cifrado deshacer inequívocamente la operación de encriptación para recuperar el mensaje original. Este dispositivo fue propuesto simultáneamente por al menos cuatro inventores de forma independiente: Alexander Koch (Holanda), Arvid Damm (Suecia), Edward Hebern (EUA) y Arthur Scherbius (Alemania). De estos, el único que gozó de éxito comercial fue el alemán Scherbius, quien inventó la máquina "Enigma" en 1918, y a mediados de los años veinte ya se había convertido en el estándar criptográfico del ejército alemán, interesado en actualizar su sistema de encriptación después de descubrir que la inteligencia británica había sido capaz de descifrar las comunicaciones alemanas a lo largo de toda la guerra. No obstante, en 1932, el matemático y criptógrafo polaco Marian Rejewski había deducido el funcionamiento de la máquina Enigma mediante documentación proporcionada por la inteligencia militar francesa. Tras la invasión alemana de Polonia en 1939 que da comienzo a la Segunda Guerra Mundial, el personal de la Oficina de Cifrado polaca en la que trabajaba Rejewski fue evacuada a Francia, desde donde colaboró con la GC&CS británica. Esta colaboración dio sus frutos en la máquina criptológica "Bombe", diseñada y fabricada por los británicos Alan Turing y Harold Keen en 1940. Esta máquina simulaba mecánicamente un conjunto de varios dispositivos Enigma, y permitió a la inteligencia británica desencriptar muchas de las claves de cifrado alemanas, que eran cambiadas diariamente, hasta el final de la guerra en 1945. Estimaciones modernas sobre el impacto de este hito sostienen un acortamiento en la duración de la guerra de al menos dos años, con el muy significativo número de vidas que ello hubiera supuesto [2], [4], [5].

1.1.2. Criptografía moderna

En el año 1949, el matemático e ingeniero estadounidense Claude E. Shannon publicó el trabajo *A Mathematical Theory of Secrecy Systems* [6], basado en gran medida en su *A Mathematical Theory of Commmuncation* [7] que había visto la luz un año antes y que se considera el trabajo fundacional de la Teoría de la Información. En su teoría de los sistemas secretos, Shannon formaliza con éxito la noción de "criptosistema" utilizando las herramientas elaboradas en su trabajo previo, y es capaz de aplicar estas a la demostración de la invulnerabilidad del cifrado *one-time pad*. El trabajo de Shannon proporcionó un lenguaje formal común para la criptología, dotando a esta de todos los elementos propios de la disciplina científica. Por otra parte, el hito de Shannon fue de una naturaleza tan general en cuanto a los aspectos

fundamentales de la criptografía que se considera el nacimiento de la criptografía moderna. El segundo gran hito en esta nueva etapa es el artículo *New Directions in Cryptography* [8], publicado por Whitfield Diffie y Martin Hellman en 1976. En este, los investigadores de la Universidad de Stanford proponen un sistema de intercambio de claves secretas entre los participantes de un proceso de comunicación sin la necesidad de establecer un canal seguro entre ellos. La innovación fundamental propuesta por Diffie y Hellman, y que constituye la piedra angular del algoritmo de intercambio de claves de Diffie-Hellman, fueron las *trapdoor one-way functions*, *i.e.*, funciones fáciles de calcular pero difíciles de invertir (*one-way functions*), a menos que se disponga de una cierta información (*trapdoor*), en cuyo caso es sencillo obtener la preimagen de un determinado valor. La inspiración para las *trapdoor one-way functions* bebe de la noción de "seguridad probable" propuesta por Shannon, quien hizo explícita la equivalencia entre la robustez de un sistema criptográfico y la dificultad para resolver un cierto problema matemático. En el caso del algoritmo propuesto por Diffie y Hellman, esta es la dificultad en general para calcular el logaritmo discreto en un cuerpo finito. Además de su propuesta, que todavía hoy se considera uno de los esquemas de intercambio de claves más prácticos y seguros, Diffie y Hellman son considerados los padres de la criptografía de clave pública como nuevo paradigma de las comunicaciones seguras, y su trabajo estimuló una investigación frenética en este área. Esta ha fructificado en soluciones criptográficas con enorme prevalencia en los sistemas de telecomunicaciones modernos, como el protocolo de intercambio de claves RSA diseñado por Ron Rivest, Adi Shamir y Leonard Adleman en 1977, o el protocolo de firma digital (*Digital Signature Algorithm*, DSA), propuesto por el Instituto Nacional de Estándares y Tecnología (*National Institute of Standards and Technology*, NIST) como estándar de firma digital en 1991.

La criptografía se diversificó gradualmente más allá de sus aplicaciones militares con el advenimiento de la computación personal y el comercio computerizado, y muy especialmente con la aparición de Internet y su llegada al gran público. En 1973, la Oficina Nacional de Estándares de los Estados Unidos (*National Bureau of Standards*, NBS) propuso un concurso público para un algoritmo de cifrado comercial estándar. Este fue concedido al grupo de criptosistemas de la tecnológica IBM, encabezado por el criptógrafo de origen alemán Horst Feistel, que presentó un sistema de cifrado desarrollado a principios de los años setenta denominado *Lucifer*. El algoritmo fue rebautizado "Estándar de Encriptación de Datos" (*Data Encription Standard*, DES), y sirvió como cifrador digital de referencia hasta 1997. En este año, la agencia de estándares estadounidense —antigua NBS, ahora NIST— presentó un nuevo concurso para un sustituto de DES. En esta ocasión el ganador

fue el algoritmo de cifrado en bloque *Rijndael*, desarrollado por los criptógrafos belgas Joan Daemen y Vincent Rijmen. En el 2000, este algoritmo fue rebautizado "Estándar de Cifrado Avanzado" (*Advanced Encryption Standard*, AES), siendo el sistema de cifrado simétrico más utilizado en la actualidad [9], [10].

1.1.3. Criptografía física y estado de la técnica

En el año 1943, la compañía estadounidense *Bell Labs* trabajaba en el diseño de un teletipo encriptado (*131-B2*) para el ejército de los Estados Unidos mediante la técnica *one-time pad*, entonces conjeturada como fundamentalmente segura. Sin embargo, los ingenieros que trabajaban con el prototipo observaron unos pulsos característicos en los osciloscopios físicamente próximos al dispositivo cada vez que se encriptaba un carácter. Estos se debían a la emanación espuria de radiación electromagnética como consecuencia de la operación eléctrica normal del teletipo. El estudio cuidadoso de los patrones captados por los osciloscopios permitió a los técnicos de los Laboratorios Bell interceptar hasta el 75 % del texto original introducido en el teletipo. Las modificaciones realizadas por la compañía para apantallar estas emanaciones fueron consideradas inaceptables por el ejército al restar movilidad al aparato, y en su lugar se estableció una zona de exclusión segura de 30 metros a la redonda en torno al punto de operación del teletipo, una vez este fuera desplegado en campaña. En 1951, la Agencia Central de Inteligencia estadounidense (*Central Intelligence Agency*, CIA) informó de que, mediante la amplificación de la señal emanada por el *131-B2*, había sido capaz de recuperar texto plano hasta una distancia de 400 metros: se habían descubierto los ataques de canal lateral (*Side Channel Attack*, SCA), una nueva forma de vulneración criptográfica cuyo objetivo es la información filtrada por el dispositivo físico sobre el cual se implementa una solución criptográfica. Este descubrimiento dio lugar a la certificación TEMPEST (*Telecommunication Electronic Material Protected from Emanating Spurious Transmissions*, TEMPEST), hoy utilizado por la NSA estadounidense y la OTAN, que califica material de comunicación como adecuadamente blindado frente al filtrado de información. El primer ataque SCA conocido, sin embargo, fue efectuado por el servicio de inteligencia interior británico (MI5) en el año 1965 sobre la embajada egipcia en Londres; este consistió en la disposición de una serie de micrófonos próximos a una máquina de cifrado mecánico utilizada por la embajada. El funcionamiento de esta máquina producía unos sonidos característicos, que los técnicos del MI5 pudieron correlacionar con la clave secreta utilizada en cada instante, lo que permitió a la

agencia británica espiar las comunicaciones secretas de la embajada durante años, a pesar de que la clave era modificada diariamente [11]-[13].

En la actualidad, la mayor parte de los esfuerzos de la comunidad criptológica se centran en la seguridad en la capa física. En este aspecto, mucho del trabajo pionero en ataques SCA sobre implementaciones físicas de sistemas criptográficos de clave pública se debe al investigador estadounidense Paul Kocher, quien en 1996 publicó una serie de ataques exitosos sobre protocolos RSA y firmas electrónicas DSA, y propuso en 1999 el análisis diferencial de potencia (*Differential Power Analysis*, DPA), una forma avanzada para relacionar el sesgo en el consumo de energía exhibido por un procesador al ejecutar un determinado algoritmo criptográfico con el contenido de la clave secreta que está siendo procesada. En el año 2004, Kocher *et al.* preconizaron la adopción de la seguridad *hardware* como una dimensión más a tener en cuenta en el diseño de sistemas embebidos [14].

Además de la seguridad en la capa física, otra línea de investigación activa es la protección frente a un hipotético ordenador cuántico, que presumiblemente podría resolver muchos de los problemas matemáticos considerados "irresolubles en la práctica" sobre los cuales se basa la seguridad probable de los protocolos de clave pública. En 2016 el NIST hizo un nuevo llamamiento público para concursar por un algoritmo de criptografía poscuántica que fuera robusto frente a las capacidades teóricas de un ordenador cuántico. Entre agosto y noviembre de 2023, el Instituto mantuvo abierto un período de revisión pública para tres candidatos finalistas a estándares de criptografía poscuántica: un algoritmo de establecimiento seguro de claves denominado *Module-Lattice-Based Key-Encapsulation* [15], y dos algoritmos de firma digital *Module-Lattice-Based Digital Signature* [16] y *Stateless Hash-Based Digital Signature* [17]. Durante este lapso, la comunidad criptográfica ha tenido la oportunidad de proporcionar comentarios técnicos específicos sobre estos finalistas. Como resultado, se espera que el NIST publique una versión estándar definitiva en 2024.

1.2. Motivación y objetivos

A pesar de que la necesidad de transmitir información de manera segura ha existido a lo largo de toda la historia, esta necesidad se ha incrementado enormemente en los últimos años debido al crecimiento exponencial de las comunicaciones

digitales. Actualmente se transmiten enormes volúmenes de información sensible y confidencial a gran velocidad, y garantizar la privacidad de todos estos datos supone un desafío mayúsculo. Tradicionalmente, todas las soluciones criptográficas eran analizadas desde un punto de vista estrictamente formal, asumiendo que los dispositivos físicos sobre los que estas eran implementadas constituían una suerte de "caja negra". De esta forma, un adversario disponía de acceso a la información de entrada/salida, pero no podía modificar la operación interna del dispositivo criptográfico, incluso aunque los detalles de la implementación fueran públicos y conocidos. Esto tiene la consecuencia de que cualquier información secreta podría considerarse segura siempre que no abandonara el dispositivo y, en particular, la generación y almacenamiento de claves sería un problema menor en el ámbito de la seguridad de la información. Sin embargo, tal y como se ha discutido en la sección anterior, estas suposiciones se han demostrado difíciles de satisfacer en la práctica, requiriendo medidas de seguridad implementadas en la capa física. Debido a ello, el modelo de comunicación segura ha pivotado a un modelo de "caja gris", en el cual los terminales criptográficos contribuyen a la superficie de ataque de un sistema de comunicación. En este contexto, las funciones no-clonables físicamente (*Physically Unclonable Function*, PUF) emergen como una alternativa prometedora para la generación y almacenamiento seguro de claves, así como la identificación y autenticación de dispositivos.

1.2.1. Introducción y antecedentes de las funciones no-clonables físicamente

Una PUF es una primitiva criptográfica con propiedades de seguridad en la capa física, potencialmente robusta frente ataques físicamente invasivos y a la manipulación física del dispositivo criptográfico. Para ello, una PUF explota la variabilidad estocástica inherente al proceso de fabricación de un dispositivo semiconductor, por la cual cada realización física de un mismo diseño PUF ("instancia" de una PUF) presenta pequeñas desviaciones, medibles pero imposibles de reproducir incluso por el fabricante original. Aplicado a sistemas digitales, las PUF producen una firma binaria capaz de identificar unívocamente un dispositivo electrónico. Estas características son muy parecidas a las mostradas por un sistema de seguridad biométrico, lo que conduce a comparar una PUF con la "huella dactilar" de un dispositivo electrónico [18]-[20].

Los casos de aplicación de las PUF, que serán discutidos en profundidad en la sección 2.3, incluyen la identificación y autenticación de entidades durante un protocolo de comunicación, así como la generación y almacenamiento seguro de claves secretas. En relación con estas aplicaciones, la ventaja más destacada de esta tecnología es proporcionar un elevado nivel de seguridad manteniendo un coste reducido en cuanto a recursos *hardware*. Esto se debe, principalmente, al hecho de que una PUF es capaz de generar identificadores únicos para un dispositivo físico sin necesidad de incluir costosas memorias no-volátiles (*Non-volatile Memory*, NVM) externas al módulo criptográfico, lo cual redunda en una mayor seguridad al evitar que el secreto deba transmitirse entre elementos diferentes dentro de un mismo sistema. Adicionalmente, las funciones no-clonables físicamente presentan ventajas como una alta sensibilidad a la manipulación, lo cual permite detectar con rapidez cualquier intento de acceso físico a la electrónica protegida por una PUF, o el hecho de que el secreto es regenerado a partir de las propiedades físicas del sustrato y no almacenado, evitando que este se exponga a un adversario durante la mayor parte de la vida del dispositivo. Por otra parte, el principal inconveniente de esta tecnología radica en la naturaleza difusa de la respuesta funcional, debida a la acción de fenómenos físicos que modifican las propiedades del material, predominantemente fluctuaciones estocásticas en la temperatura y la tensión de alimentación del módulo PUF, así como la modificación de la microestructura del dispositivo debido al envejecimiento. Esta variabilidad se traduce en la presencia de algunos bits inestables en las respuestas binarias de una PUF, lo cual obliga a implementar soluciones de corrección de errores a fin de poder generar respuestas estables [21].

El trabajo considerado como pionero en el ámbito de la seguridad *hardware* basada en la aleatoriedad de la microestructura de los dispositivos físicos (donde las funciones no-clonables físicamente representan el estado del arte al contar con el soporte de una estructura formal y una taxonomía bien establecida) se debe a Lofstrom *et al.*, quienes en el año 2000 utilizaron la variabilidad en la tensión umbral debida a defectos en la fabricación de un conjunto de transistores de efecto de campo metal-óxido-semiconductor (*Metal-Oxide-Semiconductor Field Effect Transistor*, MOSFET) para identificar un circuito concreto de entre un conjunto de circuitos idénticos por diseño [22]. El primer precursor de la PUF moderna propiamente formalizada fue la función física no-invertible (*physical one-way function*) de Pappu *et al.*, que en 2002 propusieron una técnica para extraer una firma binaria a partir del patrón de interferencia de un láser He-Ne sobre una lámina de material óptico dispersor [23], utilizando el ángulo de incidencia del láser sobre la lámina como estímulo para construir la función física (*i.e.*, la relación estímulo-respuesta). Los

autores de este trabajo fueron capaces de identificar inequívocamente una lámina particular de entre todo un conjunto de láminas diseñadas para ser idénticas, debido a la naturaleza estocástica de la distribución espacial de centros dispersores inherente a la fabricación del material óptico.

El siguiente hito en el desarrollo de PUF es la función física aleatoria (*physical random functions*), propuesta por Gassend *et al.* en 2002 [24]. En este trabajo, Gassend acuña el acrónimo "PUF" y propone una arquitectura auto-oscilante, precursora de la PUF basada en osciladores de anillo moderna, capaz de evaluar el retardo característico de un circuito digital implementado electrónicamente, siendo además una solución adecuada para su implementación en FPGA. En dicho trabajo se argumenta en contra de la no-invertibilidad postulada por Pappu como la propiedad fundamental de las funciones físicas no-invertibles en favor de su aleatoriedad: a efectos de la autenticación de dispositivos es preferible disponer de una aplicación cuyas respuestas son impredecibles dado un conjunto de desafíos por encima de una función no-invertible. A partir de este año, el concepto de PUF atrajo una considerable atención por parte de la comunidad criptológica y de diseño electrónico, convirtiéndose en objeto de una intensa investigación [25]-[29] con el fin de maximizar sus ventajas, *i.e.*, diseñar alternativas PUF seguras y más ligeras en cuanto a consumo de recursos [30], [31], así como de proporcionar bits más estables [32]-[34] o, alternativamente, estrategias para mitigar dicha aleatoriedad, *e.g.*, códigos de corrección de errores [35]-[37] y protocolos de autenticación difusa [38].

Actualmente, la investigación en el campo del diseño de funciones no-clonables físicamente está determinada por los escenarios de aplicación previstos, de tal modo que la práctica totalidad de este esfuerzo se centra en PUF electrónicas aplicadas a tecnología CMOS (*Complementary Metal-Oxide-Semiconductor*, CMOS), las cuales permiten una integración sencilla en los sistemas digitales a proteger. Esto incluye la utilización de defectos característicos de esta tecnología, como el ruido telegráfico (*Random Telegraph Noise*, RTN) [39] o los fenómenos de metaestabilidad presentes en estructuras CMOS biestables [40], [41]; así como el desarrollo de nuevas propuestas PUF y la adaptación de soluciones existentes a los dispositivos nanoelectrónicos que han emergido en los últimos años, *e.g.*, transistores de efecto de campo basados en nanotubos de carbono [42], [43], memorias RAM resistivas (*Resistive Random Access Memory*, RRAM) [44] o memorias RAM magnéticas basadas en transferencia de espín (*Spin Torque Transfer Magnetic Random Access Memory*, STT-MRAM) [45]-[47]. En el ámbito de las aplicaciones potenciales de la tecnología PUF, ocupa un lugar destacado la protección de sistemas embebidos con capacidad para intercambiar

información a través de Internet (*Internet of Things*, IoT). La investigación en esta área abarca el diseño de estructuras PUF óptimas implementadas sobre dispositivos digitales reprogramables (*i.e.*, FPGA) [48]-[52], así como la integración de estas soluciones con protocolos de comunicación estándar en IoT [53]-[55]

1.2.2. Internet de las cosas - IoT

El "Internet de las cosas" es una tecnología en la que los objetos físicos "cotidianos" disponen de la capacidad de interactuar con el entorno y de cooperar entre sí de forma local y con servicios a través de Internet. Este ecosistema de dispositivos inteligentes (*smart devices*) está siendo una realidad debido, en parte, a la reducción en tamaño, consumo y coste propiciado por el desarrollo imparable de la tecnología microelectrónica. Las capacidades aumentadas del IoT incluyen la adquisición de datos sobre el entorno físico a través de sensores, actuadores, computación "en el borde" (*edge computing*), *i.e.*, computación realizada en los nodos periféricos de una red a partir de información obtenida localmente, evitando la latencia de tener que remitir esta a un nodo de procesamiento central. Estas capacidades hacen del IoT un actor sustancial en la automatización de procesos y por ello su ámbito de aplicación se extiende virtualmente a todas las actividades productivas y de servicios: existen aplicaciones IoT integradas en mayor o menor medida junto con soluciones estándar en la medicina y sanidad [56]-[59], procesos industriales [60], [61], defensa [62], [63], etc. Sin embargo, a medida que aumentan los procesos apoyados sobre redes IoT, estas deben actualizarse a su vez para responder mejor a las necesidades específicas de cada ámbito. En [64] se enumeran algunos aspectos en los que el estado actual del IoT debe mejorarse para dar cabida a la creciente variedad de procesos apoyados en esta tecnología, incluyendo: (*i*) reducción del consumo de potencia, (*ii*) reducción del factor de forma y (*iii*) mejora en la seguridad de las redes IoT.

El aspecto de la eficiencia en recursos *hardware* resulta crucial para las redes IoT de bajo rendimiento, entendidas como tal aquellas cuyos dispositivos carecen de un suministro de potencia virtualmente ilimitado, por ejemplo, dispositivos alimentados por baterías. Dado que las operaciones criptográficas como el cifrado y la autenticación pueden consumir recursos elevados de energía y silicio, la eficiencia energética y la seguridad en el IoT están interconectadas y deben abordarse de manera equilibrada para garantizar un ecosistema IoT más robusto y confiable. Algunos ejemplos paradigmáticos de este caso son dispositivos biomédicos como marcapasos o bombas de insulina con capacidades ampliadas mediante la conexión

a la red, lo que permite, por ejemplo, analizar y configurar el dispositivo en tiempo real en función de la información recogida por sus sensores; pero cuyo consumo de energía está limitado a pequeñas baterías de difícil reemplazo. En estos casos, además, es frecuente aumentar el margen de ahorro energético a costa de unas soluciones de ciberseguridad limitadas debido al impacto energéticamente elevado que tienen las primitivas criptográficas estándar basadas en *software* [65], [66]. En este contexto, la tecnología PUF emerge como una candidata prometedora para la autenticación y el almacenamiento seguro de claves, sin necesidad de alimentar un módulo específico destinado al almacenamiento seguro de información.

Por otra parte, la tendencia a sustituir microcontroladores por matrices de puertas programables (*Field Programmable Gate Array*, FPGA) ha llegado también a su empleo como dispositivos en redes IoT [67], [68]. Estos dispositivos permiten actualizar y rediseñar un componente sin necesidad de retirar el *hardware* obsoleto, y a la vez presentan unos perfiles de consumo energético y silicio reducidos en nodos de integración sucesivamente menores, incluso en sistemas tan complejos como celdas IoT de procesamiento de vídeo [69]. Este hecho, junto con la discusión previa a propósito del impacto negativo de las soluciones de ciberseguridad en el consumo energético de los dispositivos IoT, constituyen el *leitmotiv* del presente trabajo, donde nos centraremos en buscar estrategias para reducir la tasa "consumo energético/nivel de seguridad".

1.2.3. Objetivos de la tesis

La implementación de algoritmos convencionales de autenticación, encriptación y almacenamiento seguro de claves en memorias no volátiles supone un coste prohibitivo en silicio y consumo de potencia para los dispositivos IoT, por lo que en estos casos se requiere una solución más económica, más eficiente energéticamente y menos vulnerable a ataques. En esta tesis queremos avanzar en el desarrollo de un nuevo paradigma de seguridad basado en PUF, el cual dispone de potencial para proporcionar excelentes resultados en cuanto a la seguridad a nivel de la capa física de los ecosistemas IoT.

El objetivo general de esta tesis es la síntesis, diseño e implementación de arquitecturas PUF eficientes y compactas en dispositivos FPGA, aptas para su despliegue en sistemas embebidos, y que proporcionen capacidades de autenticación y generación de claves en dispositivos IoT seguros.

Los principales objetivos específicos perseguidos en esta tesis son:

- Analizar el estado de la técnica en cuanto a la teoría y práctica del diseño e implementación de PUF microelectrónicas, así como su potencial para proporcionar servicios de seguridad a sistemas con importantes restricciones en consumo de potencia y superficie de silicio, concretamente dispositivos de bajo rendimiento conectados a la red como parte del ecosistema IoT. En particular, buscamos obtener una imagen precisa de las propuestas existentes de PUF integrables en la plataforma FPGA más relevantes y de las limitaciones que presentan, así como proponer diversas soluciones a las mismas.

- Proponer un marco formal unificado capaz de capturar adecuadamente las propiedades de los sistemas PUF generales, abarcando desde sus características fundamentales hasta su aplicación práctica en sistemas de seguridad, así como su respuesta a diferentes condiciones ambientales y de operación. Este formalismo permitirá describir las funciones no-clonables físicamente en varios niveles de idealización y, en particular, será utilizado para definir un modelo cuasi-ideal de PUF que constituirá un estándar de comparación. Dicho modelo facilitará la evaluación de las diferentes propuestas de PUF realizadas en este trabajo atendiendo a sus propiedades de identificabilidad y grado de idealidad.

- Introducir un modelo físico de fabricación de funciones físicas basadas en la repetición de subunidades idénticas por diseño (celdas), así como el concepto de "topología" sobre la estructura celular de estas. Mediante esta noción se proporcionará un método sistemático para deducir la entropía y minentropía de varias propuestas PUF típicas basadas en el método de la medida compensada. Estas métricas fundamentales se utilizarán para proponer topologías alternativas novedosas, más eficientes y compactas desde el punto de vista de los recursos *hardware*, así como más seguras criptográficamente, de forma que resulten adecuadas para su despliegue en dispositivos IoT.

- Analizar de manera exhaustiva las propiedades de una PUF basada en osciladores de anillo sobre FPGA, identificando para ello todos los grados de libertad del diseño capaces de afectar el rendimiento de estas arquitecturas. Para ello, se diseñará un entorno integrado de desarrollo que permita evaluar de manera sistemática el impacto del soporte físico en las propiedades de seguridad de la PUF. Se llevará a cabo una evaluación experimental utilizando el *system-on-chip* Zynq-7000 y la herramienta de diseño Vivado, ambas de Xilinx. Las conclusiones extraídas a propósito de la manera en que influyen las

variables de diseño identificadas previamente nos permitirán proponer opciones de arquitecturas PUF basadas en osciladores de anillo que sean óptimas en términos de eficiencia energética y aprovechamiento de la superficie de silicio en FPGA.

- Proponer una arquitectura de PUF novedosa basada en osciladores de anillo de Galois, que sea capaz de solucionar algunos de los inconvenientes paradigmáticos de las propuestas PUF basadas en el retardo de celdas y que son encontrados típicamente en las alternativas PUF más habituales, en particular la correlación espacial entre osciladores de anillo estándar en función de su ubicación sobre la matriz FPGA. Estas correlaciones existen tanto en el largo alcance, debido a variaciones sistemáticas durante la fabricación de cada chip, como a nivel local debido a efectos de acoplo electromagnético entre anillos próximos, lo que lleva a la sincronización de sus oscilaciones. En este trabajo se busca proponer una solución a esta problemática en particular. Adicionalmente, los osciladores de anillo de Galois pueden ser utilizados como generadores de números verdaderamente aleatorios (*True Random Number Generator*, TRNG). Dado que las PUF —como primitiva de seguridad— a menudo se despliegan conjuntamente con TRNG como parte de un protocolo criptográfico, las soluciones que se propondrán en esta tesis permitirán combinar en una única estructura ambas funcionalidades, logrando un consumo de recursos reducido.

1.3. Estructura de la tesis

Esta memoria de tesis se divide en seis capítulos. El capítulo primero y segundo contienen respectivamente la introducción a este documento y los conceptos teóricos utilizados en él, mientras que en el sexto y último se recogen las conclusiones obtenidas a lo largo de este trabajo y se introducen algunas líneas de investigación futuras abiertas por este tesis. El grueso del trabajo de investigación realizado se expone en los capítulos tres, cuatro y cinco. La memoria ha sido estructurada de forma modular, de manera tal que los resultados obtenidos en sucesivos capítulos son complementarios y mutuamente inclusivos.

Capítulo 1. Introducción a la criptografía desde una perspectiva histórica, así como una descripción de las necesidades y desafíos actuales en cuanto a seguridad

de la información, que sirve como motivación para los objetivos perseguidos en esta tesis doctoral.

Capítulo 2. Compendio de los conceptos fundamentales y cuerpo teórico en los que se apoya el trabajo realizado en esta tesis. Este capítulo incluye una breve introducción a la teoría de la información, la seguridad de la información y la criptología moderna. Finalmente, se hace una exposición detallada y exhaustiva de las funciones no-clonables físicamente como primitivas de seguridad, incluyendo sus aplicaciones y métodos de clasificación, así como una propuesta de formalismo general capaz de capturar adecuadamente las propiedades de seguridad de esta tecnología.

Capítulo 3. Análisis de las propiedades de seguridad teóricas de PUF de medida compensada, las cuales constituyen una familia muy general de funciones no-clonables físicamente y particularmente adecuada para su implementación física en sistemas digitales.

Capítulo 4. Introducción de la plataforma FPGA como dispositivo digital CMOS, incidiendo en su papel actual y potencial como vector de implementación para el IoT. A continuación, se describe la PUF basada en matrices de osciladores de anillo (*Ring Oscillator PUF*, RO-PUF) y se propone una implementación física concreta sobre FPGA. Finalmente, se estudian alternativas de diseño dirigidas a solventar algunos inconvenientes reportados en la literatura a propósito los sistemas RO-PUF, así como a la propuesta de soluciones para mejorar sus propiedades de seguridad.

Capítulo 5. Propuesta y análisis experimental de una arquitectura PUF novedosa basada en osciladores de anillo de Galois, adecuada para su implementación en FPGA, y que resuelve el problema de la correlación espacial entre respuestas típico de la PUF basada en osciladores de anillo estándar.

Capítulo 6. Conclusiones de esta tesis, así como posibles líneas de investigación abiertas.

2 Definiciones y conceptos básicos

En este capítulo se detallan las definiciones generales que se seguirán a lo largo de esta tesis, así como diversos conceptos básicos. La primera parte del mismo se dedica a la teoría de la información, que constituye el marco formal natural para una teoría de la seguridad de la información. A continuación, se discuten en profundidad las propiedades de las funciones no-clonables físicamente, su clasificación y las métricas características utilizadas para su evaluación experimental, con especial atención a las alternativas adecuadas para su implementación en FPGA. En esta sección se propone una descripción formal de las PUF, las cuales se definen como la composición de una función física con una interfaz digital. También se exploran las propiedades ideales de esta construcción, y se propone un modelo cuasi-ideal capaz de capturar la no-idealidad observada experimentalmente en estos sistemas. Finalmente, se introduce la FPGA como plataforma de diseño digital, detallando la topología del modelo Artix 7 utilizado en esta tesis y destacando las propiedades de mayor relevancia para el diseño de PUF.

2.1. Teoría de la información

La teoría de la información es una disciplina formal que estudia algunos aspectos fundamentales de la teoría (más general) de la comunicación, a saber: el límite en la compresión de datos generados por una fuente de información (*i.e.*, enviar una cierta información empleando para ello el menor número de símbolos posible), y la velocidad máxima (símbolos por segundo) a la cual un cierto canal puede transmitir información. Como ya se mencionó en el capítulo anterior, el primer hito de la teoría de la información —hasta el punto de que puede considerarse el trabajo fundacional de esta disciplina— fue la publicación, en 1948, del trabajo *A mathematical theory of communication* [7] por parte del matemático e ingeniero estadounidense Claude Shannon. Entre las aportaciones esenciales de esta publicación destacan la identificación de la entropía de una fuente de información (H) como la

magnitud que determina su razón de compresión máxima, así como la "capacidad de un canal de comunicación" (C), para medir la tasa de transmisión de un canal. Shannon fue capaz de relacionar ambas cantidades en su "teorema de la codificación no-ruidosa", donde demuestra que la velocidad máxima a la que se puede transmitir la información generada por una cierta fuente a través de un determinado canal está dada por el cociente C/H. No obstante, la principal contribución de Shannon en este trabajo fue la presentación y demostración del "teorema de la codificación ruidosa", donde se establece que, dado un canal de comunicación el cual introduce errores en los mensajes transmitidos (*i.e.*, que elimina, añade o cambia algunos símbolos), y siempre que la cantidad de información media por unidad de tiempo generada por una fuente de información no supere la capacidad del canal, entonces se puede encontrar una codificación de los mensajes tal que la probabilidad de error sea arbitrariamente pequeña.

De forma colateral, la teoría de Shannon tuvo un impacto sustancial en otras disciplinas técnicas, en particular en la criptología. Los conceptos de "cantidad de información" y "fuente de información" presentados en [7] proporcionan un punto de partida formal para la caracterización del nivel de seguridad proporcionado por una cierta solución criptográfica [70]-[72]. Así mismo, se demuestra una similitud formal en la operación de un criptógrafo que trata de enmascarar un mensaje variando algunos símbolos y un canal ruidoso que modifica el contenido de un mensaje de forma accidental. En esta sección proporcionamos un trasfondo de las nociones propias de la teoría de la información que son utilizadas a lo largo de esta tesis.

2.1.1. Contenido de información

Naturalmente, la "información" como objeto formal de una teoría de la comunicación es independiente de la semántica asociada a un mensaje. En su lugar, esta se refiere *grosso modo* al número de caracteres necesarios para identificar unívocamente un mensaje concreto de entre un repertorio de posibles mensajes.

A continuación, vamos a exponer los conceptos fundamentales de la teoría de la información necesarios en esta tesis, utilizando un formalismo matemático simple pero riguroso. Sea un multiconjunto[1] (repertorio) de mensajes $\mathbf{X} \equiv \{x_i^{n_i}\}$

[1] Un multiconjunto es una colección de elementos que, a diferencia de un conjunto, sí puede contener varios elementos iguales.

donde n_i representa el número de veces que se repite el mensaje x_i en el repertorio (multiplicidad de x_i) y el símbolo de equivalencia "$\equiv$" denota una definición; un alfabeto (*i.e.*, colección de símbolos) $\mathbf{A} \equiv \{a_i\}$ y un codificador que lleva mensajes de $\mathbf{X}$ a tuplas ordenadas de símbolos, $\mathrm{Encod} : \mathbf{X} \to (a_1, a_2, \dots) \equiv (a_i)$, $a_i \in \mathbf{A}$. Diremos que dos mensajes x, x' son iguales si ambos codifican para la misma tupla de símbolos:

$$x = x' \iff \mathrm{Encod}(x) = \mathrm{Encod}(x') \tag{2.1}$$

Así mismo, definimos la suma de mensajes $x + x'$ como el mensaje x'' tal que su codificación es la yuxtaposición de las tuplas de los sumandos:

$$+ : \mathbf{X} \times \mathbf{X}' \to \mathbf{X}'' \tag{2.2}$$

$$x, x' \mapsto x'' \tag{2.3}$$

$$\mathrm{Encod}(x + x') = (\mathrm{Encod}(x), \mathrm{Encod}(x')) \tag{2.4}$$

donde el símbolo "$\times$" representa el producto cartesiano de espacios. Notar que esta operación es no-conmutativa. También puede definirse la suma de dos repertorios de la forma usual en teoría de multiconjuntos como el repertorio que contiene todos los mensajes de sus sumandos, con la suma de sus multiplicidades en caso de que ambos sumandos compartan un cierto mensaje,

$$\begin{aligned} \mathbf{X} + \mathbf{X}' \equiv & \left\{ x_i^{n_i + n'_i} | x_i \in \mathbf{X} \,\mathrm{AND}\, x_i \in \mathbf{X}' \right\} \\ & \cup \left\{ x_i^{n_i} | (x_i \in \mathbf{X} \,\mathrm{XOR}\, x_i \in \mathbf{X}') \right\} \end{aligned} \tag{2.5}$$

donde AND ("y" lógico) representa que ambos sumandos contienen el mensaje x simultáneamente, y XOR ("o exclusivo" lógico) representa que el mensaje x es contenido por uno de los sumandos exclusivamente. Finalmente, dado que la semántica de los mensajes es irrelevante, diremos que dos repositorios son iguales si son isomorfos, *i.e.*, si existe una relación 1:1 entre ellos (o equivalentemente, tienen el mismo número de elementos [73]):

$$\mathbf{X} = \mathbf{X}' \iff |\mathbf{X}| = |\mathbf{X}'| \tag{2.6}$$

donde la cardinalidad del multiconjunto se define como la suma de las multiplicidades de cada uno de sus elementos, $|\mathbf{X}| \equiv \sum_i n_i$.

Sea un repertorio $\mathbf{\Omega}$ y una partición $\{\mathbf{X}_i\}$ de multiconjuntos tales que cada uno sólo contiene el mensaje x_i con su correspondiente multiplicidad n_i, $\mathbf{X}_i = \{x_i^{n_i}\}$, de tal forma que el tamaño de la i-ésima parte es $|\mathbf{X}_i| = n_i$. Por (2.5) se puede escribir el repertorio completo como $\mathbf{\Omega} = \sum_i \mathbf{X}_i$ y su tamaño $|\mathbf{\Omega}| = \sum_i |\mathbf{X}_i|$. Se postula que

existe una medida de la cantidad de información, I_{x_i}, asociable a cada mensaje $x_i \in \mathbf{X}_i \subseteq \mathbf{\Omega}$, que es definida positiva, $I_{x_i} \geq 0$, cuya dependencia de la forma más general posible será $I_{x_i} = I_{x_i}(x_i, \mathbf{X}_i, \mathbf{\Omega})$. Dado que todos los mensajes de un mismo repertorio $\mathbf{X}_i$ son iguales, puede suprimirse esta dependencia de cada mensaje concreto y escribir $I_{x_i} \equiv I_{\mathbf{X}_i} = I_{\mathbf{X}_i}(\mathbf{X}_i, \mathbf{\Omega})$. Así mismo, postulamos que la cantidad de información de dos repertorios iguales será igual, $\mathbf{X} = \mathbf{X}' \Rightarrow I_{\mathbf{X}} = I_{\mathbf{X}'}$, y dado que por (2.6) dos repertorios son iguales si tienen el mismo tamaño, $|\mathbf{X}| = |\mathbf{X}'|$, la dependencia de la función $I_{\mathbf{X}_i}$ no podrá ser de otra forma más que $I_{\mathbf{X}_i} = I_{\mathbf{X}_i}(|\mathbf{X}_i|, \mathbf{\Omega})$. El mismo argumento puede aplicarse a la medida de información de dos repertorios $\mathbf{\Omega}$ iguales, $\mathbf{\Omega} = \mathbf{\Omega}' \Rightarrow I_{\mathbf{\Omega}} = I_{\mathbf{\Omega}'}$, y escribir $I_{\mathbf{X}_i} = I_{\mathbf{X}_i}(|\mathbf{X}_i|, |\mathbf{\Omega}|)$. Consideremos ahora la suma de dos repertorios iguales, $2\mathbf{\Omega} \equiv \mathbf{\Omega} + \mathbf{\Omega}$ (o más en general, definamos $n\mathbf{\Omega} \equiv \sum_i^n \mathbf{\Omega}$). Es razonable suponer que la cantidad de información contenida en dos repertorios idénticos (o en general, n repertorios idénticos) no es mayor que la contenida en uno solo de tales repertorios. Por lo tanto, postulamos una última condición para la cantidad de información, $I_{\mathbf{X}_i} = I_{n\mathbf{X}_i}$. Dado que los tamaños de los repertorios sí escalan, $I_{n\mathbf{X}_i} = I_{n\mathbf{X}_i}(|n\mathbf{X}_i|, |n\mathbf{\Omega}|) = I_{n\mathbf{X}_i}(n|\mathbf{X}_i|, n|\mathbf{\Omega}|)$, del postulado anterior se deduce:

$$I_{\mathbf{X}_i}\left(|\mathbf{X}_i|, |\mathbf{\Omega}|\right) = I_{\mathbf{X}_i}\left(n|\mathbf{X}_i|, n|\mathbf{\Omega}|\right) \tag{2.7}$$

Para un repertorio $\mathbf{\Omega}$, la función $I_{\mathbf{X}_i}$ es monótonamente creciente[2] con $|\mathbf{X}_i|$, de modo que la única forma de satisfacer (2.7) es mediante la igualdad de sus argumentos:

$$\left(|\mathbf{X}_i|, |\mathbf{\Omega}|\right) = \left(n|\mathbf{X}_i|, n|\mathbf{\Omega}|\right) \tag{2.8}$$

Esta es precisamente la definición del conjunto de los números racionales, *i.e.*, el conjunto de las clases de equivalencia dadas por la relación $(m, s) = (nm, ns)$, $n, m, s \in \mathbb{Z}$, de modo que podemos escribir la dependencia de la función I como:

$$I_{\mathbf{X}_i} = I_{\mathbf{X}_i}\left(\frac{|\mathbf{X}_i|}{|\mathbf{\Omega}|}\right) \tag{2.9}$$

Dada una fuente de información caracterizada por una variable aleatoria X que extrae N mensajes distribuidos uniformemente[3] del repertorio $\mathbf{\Omega}$, la proporción de mensajes provenientes de $\mathbf{X}_i$ será precisamente $N|\mathbf{X}_i|/|\mathbf{\Omega}|$, *i.e.*, la probabilidad (Prob) de extraer un mensaje de $\mathbf{X}_i$ será: $p_i \equiv \mathrm{Prob}(\mathsf{X} = x_i \in \mathbf{X}_i) = |\mathbf{X}_i|/|\mathbf{\Omega}|$. De este modo, expresamos (2.9), que depende de un único grado de libertad, como:

$$I_{\mathbf{X}_i} = I(p_i) \tag{2.10}$$

[2] Dado que el tamaño de la suma de repertorios es monótonamente creciente, $|\mathbf{X} + \mathbf{X}'| \geq |\mathbf{X}|$.

[3] Notar que esto permite caracterizar una fuente de mensajes con toda generalidad, ya que no hemos impuesto ninguna condición sobre los tamaños $|\mathbf{X}_i|$. De este modo, las distintas probabilidades de extraer uno u otro mensaje se puede capturar ajustando los tamaños de cada parte $\mathbf{X}_i$.

La forma funcional explícita de esta función puede deducirse imponiendo que sea continua en p_i y lineal respecto de la suma de mensajes definida en (2.2), $I_{x_i+x_j} = I_{x_i} + I_{x_j}$. Dado que la probabilidad de obtener conjuntamente x_i y x_j será $p_{ij} = p_i p_j$, esta propiedad se traduce en:

$$I(p_i p_j) = I(p_i) + I(p_j) \tag{2.11}$$

Utilizando que las cantidades p son definidas positivas hacemos el siguiente cambio de variables: $z \equiv \log_\alpha p_i$, $t \equiv \log_\alpha p_j$ donde α es una constante arbitraria. De este modo $p_i = \alpha^z$, $p_j = \alpha^t$, e introduciendo esto en (2.11) tenemos:

$$I\left(\alpha^{z+t}\right) = I\left(\alpha^z\right) + I\left(\alpha^t\right) \tag{2.12}$$

Definimos ahora la función $f(s) \equiv I\left(\alpha^s\right)$, de modo que reescribimos (2.12) como:

$$f(z+t) = f(z) + f(t) \tag{2.13}$$

Esta es la conocida ecuación funcional de Cauchy [74], cuya solución analítica es:

$$f(s) = \lambda s, \ \ s \in \mathbb{R} \tag{2.14}$$

lo cual nos permite escribir la función I como:

$$I\left(\alpha^z\right) = \lambda z \tag{2.15}$$

Y deshaciendo el cambio de variable $z = \log_m p_i$, obtenemos:

$$I\left(p_i\right) = \lambda \log_\alpha p_i \tag{2.16}$$

que depende de dos constantes α y λ. En teoría de la información se fijan por convenio estas constantes $\lambda = -1$ y $\alpha = 2$, de modo que el contenido de información sea positivo y esté dado en número de bits:

$$I\left(p\right) = -\log_2 p \tag{2.17}$$

Dado que en general "p" será la distribución de probabilidad de una variable aleatoria X, $p_x \equiv \mathrm{Prob}(\mathsf{X} = x)$, nos permitiremos el siguiente abuso del lenguaje al representar la cantidad de información, $I(x) \equiv I(p_x)$. A partir de esta cantidad podemos definir otras nociones:

$$I(x, y) = -\log_2 p_{xy} \tag{2.18}$$

donde $p_{xy} \equiv \mathrm{Prob}(\mathsf{X} = x, \mathsf{Y} = y)$ es la probabilidad conjunta de que las variables aleatorias X e Y tomen simultáneamente los valores x, y respectivamente.Dado que podemos escribir la probabilidad marginal $p_x = \sum_y p_{xy}$ y que las cantidades p son definidas positivas, se tiene que $p_{xy} \leq p_x$. Como $I(p)$ decrece monótonamente con p se puede escribir:

$$I(x, y) \geq I(x) \tag{2.19}$$

cumpliéndose la igualdad sólo si ambas variables son independientes. Además, dado que la probabilidad del evento conjunto $(\mathsf{X}_1 = x_1, \cdots, \mathsf{X}_n = x_n)$ para n variables aleatorias es invariante respecto de una permutación cualquiera σ de los índices[4], *i.e.*, $p_{[1,\cdots,n]} = p_{[\sigma(1),\cdots,\sigma(n)]}$, se tiene para la cantidad de información:

$$I(x_1, \cdots, x_n) = I(x_{\sigma(1)}, \cdots, x_{\sigma(n)}) \;\; \forall \sigma \equiv \text{permutación}\,(1, \cdots, n) \tag{2.20}$$

Análogamente, la asociatividad de la intersección de conjuntos es heredada por la cantidad de información:

$$I[(x, y), z] = I[x, (y, z)] \tag{2.21}$$

Dada la probabilidad condicional $p_{x|y} \equiv \mathrm{Prob}(\mathsf{X} = x|Y = y)$ (*i.e.*, la probabilidad de que la variable X tome el valor x dado que la variable Y toma el valor y), se define la cantidad de información condicional:

$$I(x|y) = -\log_2 p_{x|y} \tag{2.22}$$

Esta magnitud mide la cantidad de información de x que es revelada por el conocimiento de y (o equivalentemente, la incertidumbre que queda a x una vez se conoce y). Dado que $p_{x|y}$ es definida positiva, tendremos que

$$I(x|y) \geq 0 \tag{2.23}$$

con la igualdad cuando $y = x$, dado que $p_{x|x} = 1$. Aplicando el teorema de Bayes sobre la probabilidad condicional, $p_{x|y}p_y = p_{y|x}p_x$, se deduce:

$$I(x|y) = I(y|x) + I(x) - I(y) \tag{2.24}$$

[4]Dada la intersección de los eventos $(X_i = x_i)$, $\bigcap_{i=1}^{n} (\mathsf{X}_i = x_i) = \bigcap_{i=1}^{n} \left(\mathsf{X}_{\sigma(i)} = x_{\sigma(i)}\right)$, con $\sigma(i)$ el i-ésimo elemento de una permutación cualquiera de los índices $(1, \cdots, n)$.

Por otro lado, las cantidades de información conjunta y condicional se pueden relacionar aplicando la definición de probabilidad conjunta como probabilidad de la intersección, $p_{xy} = p_{x|y}p_y$:

$$I(x, y) = I(x|y) + I(y) \tag{2.25}$$

Esta última propiedad puede extenderse al caso en el que hay dependencia condicional de una tercera variable $\mathsf{Z} = z$. Utilizando (2.21) se tiene $I[(x, z), y] = I[x, (z, y)]$, y descomponiendo cada término como en (2.25):

$$\begin{aligned} I[(x, z), y] &= I[x, (z, y)] \\ I(x, z|y) + I(y) &= I(x|z, y) + I(z, y) \\ I(x, z|y) + I(y) &= I(x|z, y) + I(z|y) + I(y) \\ I(x, z|y) &= I(x|z, y) + I(z|y) \end{aligned} \tag{2.26}$$

De un modo parecido puede extenderse la independencia respecto de permutaciones de los argumentos para el caso de variables condicionales; empleando (2.20) y (2.25):

$$\begin{aligned} I(x, z|y) &= I[(\cancel{x, z}\overset{(z,x)}{\rightarrow}), y] - I(y) \\ &= I(z, x|y) \end{aligned} \tag{2.27}$$

$$\begin{aligned} I(x|z, y) &= I[x, (\cancel{z, y}\overset{(y,z)}{\rightarrow})] - I(\cancel{z, y}\overset{(y,z)}{\rightarrow}) \\ &= I(x|y, z) \end{aligned} \tag{2.28}$$

Se puede deducir otra propiedad interesante de la cantidad de información condicional reescribiendo (2.24) como $I(x|y) - I(x) = I(y|x) - I(y)$. Si llamamos a esta cantidad "α", *i.e.*, $\alpha \equiv I(x|y) - I(x) = I(y|x) - I(y)$, utilizando (2.24) tenemos $I(x|y) = I(x) - \alpha$. La posibilidad $\alpha > I(x)$ conduce al absurdo $I(x|y) < 0$, en contradicción con (2.23), de modo que sólo es posible $\alpha \geq 0$, que podemos escribir como la propiedad:

$$I(x|y) \leq I(x) \tag{2.29}$$

con la igualdad cuando $p_{x|y} = p_x$, *i.e.*, cuando ambas variables son independientes.

Finalmente, podemos relacionar las cantidades de información conjunta y condicional: utilizando (2.19) escribimos $I(x, y, z) \geq I(x, y)$, y como la cantidad

de información es invariante respecto de permutaciones en los argumentos (2.20) se tiene $I(x,z,y) \geq I(x,y)$; ahora sustraemos una cantidad $I(y)$ a ambos lados de la igualdad, $I(x,z,y) - I(y) \geq I(x,y) - I(y)$, y utilizando (2.25) escribimos esto como:

$$I(x,z|y) \geq I(x|y) \tag{2.30}$$

A continuación, utilizaremos estos resultados a propósito de la cantidad de información para caracterizar las fuentes de información de manera conjunta, utilizando la función entropía, H, tal y como fue propuesta originalmente por Shannon en [7]. Esta cantidad será a la postre una métrica fundamental para evaluar la calidad criptográfica de un sistema de seguridad de la información.

2.1.2. Entropía

Dado un repertorio $\mathbf{X}$ de mensajes $x \in \mathbf{X}$, se define la entropía H de la variable aleatoria X que extrae el mensaje x con probabilidad $p_x \equiv \mathrm{Prob}(X = x)$ como la cantidad de información promedio [75]:

$$H(\mathsf{X}) \equiv \overline{I(x)} = \sum_x p_x\, I(x) = -\sum_x p_x \log_2 p_x \tag{2.31}$$

Esta cantidad está acotada inferiormente por 0, que tiene lugar cuando la variable X se distribuye como una delta de Kronecker[5], $p_x = \delta_{x,x_0}$, y puede demostrarse (apéndice A) que la función entropía es máxima cuando la distribución de probabilidad es uniforme, $p_x = \text{cte}$. De forma análoga a (2.31) pueden definirse otras cantidades como promedios de las distintas magnitudes de cantidad de información:

$$H(\mathsf{X},\mathsf{Y}) \equiv \overline{I(x,y)} = \sum_{xy} p_{xy}\, I(x,y) = -\sum_{xy} p_{xy} \log_2 p_{xy} \tag{2.32}$$

es la entropía conjunta para la distribución de probabilidad $p_{xy} \equiv p\,(\mathsf{X} = x, \mathsf{Y} = y)$ asociada a dos variables X, Y que extraen mensajes de dos reservorios $\mathbf{X}$, $\mathbf{Y}$. Utilizando esta definición podemos escribir:

$$\begin{aligned} H(\mathsf{X},\mathsf{Y}) - H(\mathsf{X}) - H(\mathsf{Y}) &= -\sum_{xy} p_{xy} \log_2 p_{xy} + \sum_x p_x \log_2 p_x + \sum_y p_y \log_2 p_y \\ &= -\sum_{xy} p_{xy} \log_2 p_{xy} + \sum_{xy} p_{xy} \log_2 p_x + \sum_{xy} p_{xy} \log_2 p_y \end{aligned}$$

[5]Definida como: $\delta_{ij} = 1$ si $i = j$, 0 en otro caso.

$$
\begin{aligned}
&= -\sum_{xy} p_{xy} \log_2 p_{xy} + \sum_{xy} p_{xy} \left[\log_2 p_x + \log_2 p_y\right] \\
&= -\sum_{xy} p_{xy} \log_2 p_{xy} + \sum_{xy} p_{xy} \log_2 p_x p_y \\
&= \sum_{xy} p_{xy} \log_2 \frac{p_x p_y}{p_{xy}} \equiv \overline{\log_2 \alpha}
\end{aligned} \tag{2.33}
$$

donde se ha utilizado $p_i = \sum_{ij} p_{ij}$ en el segundo paso y $\alpha = \alpha(x, y) \equiv p_x p_y / p_{xy}$. Dado que el logaritmo es una función cóncava aplicamos la desigualdad de Jensen[6] y escribimos $\overline{\log_2 \alpha} \leq \log_2 \overline{\alpha}$; el promedio de α será:

$$
\overline{\alpha} = \sum_{xy} p_{xy} \frac{p_x p_y}{p_{xy}} = \sum_{xy} p_x p_y = 1 \tag{2.34}
$$

de forma que $\log_2 \overline{\alpha} = 0 \Rightarrow \overline{\log_2 \alpha} \leq 0$. Recuperando este resultado en el desarrollo (2.33) se tiene $H(\mathsf{X}, \mathsf{Y}) - H(\mathsf{X}) - H(\mathsf{Y}) \leq 0$, que escribimos en forma de la propiedad:

$$
H(\mathsf{X}, \mathsf{Y}) \leq H(\mathsf{X}) + H(\mathsf{Y}) \tag{2.35}
$$

con igualdad cuando ambas variables son independientes, $p_{xy} = p_x p_y$. Esta noción permite definir la tasa de entropía, h, para un proceso estocástico $\vec{\mathsf{X}} = (\mathsf{X}_1, \cdots, \mathsf{X}_n)$ (entendido este como una sucesión de variables aleatorias), a saber:

$$
h\left(\vec{\mathsf{X}}\right) \equiv \lim_{n \to \infty} \frac{H\left(\mathsf{X}_1, \cdots, \mathsf{X}_n\right)}{n} \tag{2.36}
$$

Tomando el promedio en (2.20) se deduce la invariancia de la entropía respecto de una permutación de sus argumentos, $\overline{I(x_1, \cdots, x_n)} = \overline{I(x_{\sigma(1)}, \cdots, x_{\sigma(n)})}$:

$$
H(\mathsf{X}_1, \cdots \mathsf{X}_n) = H(\mathsf{X}_{\sigma(1)}, \cdots \mathsf{X}_{\sigma(n)}) \;\; \forall \sigma \equiv \text{permutación}(1, \cdots, n) \tag{2.37}
$$

y por (2.21) se tiene la asociatividad de la entropía:

$$
H((\mathsf{X}, \mathsf{Y}), \mathsf{Z}) = H(\mathsf{X}, (\mathsf{Y}, \mathsf{Z})) \tag{2.38}
$$

Así mismo, se define la entropía condicional como el promedio de (2.22),

$$
H\left(\mathsf{X}|\mathsf{Y}\right) \equiv \overline{I(x|y)} = \sum_{xy} p_{xy}\, I(x|y) = -\sum_{xy} p_{x|y}\, p_y \log_2 p_{x|y} \tag{2.39}
$$

[6] Una función cóncava se puede caracterizar por el hecho de que para cualquier punto x_0 del dominio, el grafo de la curva está por debajo de la recta tangente a x_0, *i.e.*, $f(x) \leq r(x)\ \forall x$, donde $r(x) = (x - x_0)m + n$, $m = df/dx|_{x_0}$, $n = f(x_0)$. En particular esto será cierto para $x_0 = \overline{x}$, $f(x) \leq r(x) = (x - \overline{x})m + n$. Utilizando que el promedio es una operación lineal y por tanto preserva la desigualdad (o, en general, cualquier relación de orden: dado $a \lesseqgtr b$ se tiene $a - b \lesseqgtr 0$; tomando promedios $\overline{a - b} = \overline{a} - \overline{b} \lesseqgtr \overline{0} = 0 \Rightarrow \overline{a} \lesseqgtr \overline{b}$) escribimos $\overline{f(x)} \leq \overline{r(x)}$, donde $\overline{r(x)} = (\overline{x} - \overline{x})m + n = n = f(\overline{x})$; de modo que para una función cóncava $f(x)$ se tiene $\overline{f(x)} \leq f(\overline{x})$

donde se ha utilizado la definición de probabilidad condicional, $p_{x|y} = p_{xy}/p_y$. Si fijamos la variable y al valor y' (*i.e.*, si asumimos una distribución de probabilidad para Y en la forma de "delta de Kronecker", $p_y = \delta_{y,y'}$) la probabilidad condicional queda:

$$\begin{aligned} H\left(\mathsf{X}|\mathsf{Y}=y'\right) &= -\sum_{xy} p_{x|y}\,\delta_{y,y'}\log_2 p_{x|y} \\ &= -\sum_{x} p_{x|y'}\log_2 p_{x|y'} \end{aligned} \tag{2.40}$$

De manera que (2.39) puede escribirse como el promedio de las entropías condicionales parciales:

$$\begin{aligned} H(\mathsf{X}|\mathsf{Y}) &= -\sum_{xy} p_{x|y}\,p_y\log_2 p_{x|y} \\ &= \sum_{y} p_y\left[-\sum_{x} p_{x|y}\,\log_2 p_{x|y}\right] \\ &= \sum_{y} p_y\,H(\mathsf{X}|\mathsf{Y}=y) \end{aligned} \tag{2.41}$$

De la relación entre la entropía H y la cantidad de información I, y dado que el promedio preserva las relaciones de orden, se obtienen automáticamente las propiedades:

$$H\left(\mathsf{X}|\mathsf{Y}\right) \geq 0 \tag{2.42}$$
$$H(\mathsf{X}|\mathsf{Y}) = H(\mathsf{Y}|\mathsf{X}) + H(\mathsf{X}) - H(\mathsf{Y}) \tag{2.43}$$
$$H(\mathsf{X},\mathsf{Y}) = H(\mathsf{X}|\mathsf{Y}) + H(\mathsf{Y}) \tag{2.44}$$
$$H(\mathsf{X},\mathsf{Z}|\mathsf{Y}) = H(\mathsf{X}|\mathsf{Z},\mathsf{Y}) + H(\mathsf{Z}|\mathsf{Y}) \tag{2.45}$$
$$H(\mathsf{X},\mathsf{Z}|\mathsf{Y}) = H(\mathsf{Z},\mathsf{X}|\mathsf{Y}) \tag{2.46}$$
$$H(\mathsf{X}|\mathsf{Z},\mathsf{Y}) = H(\mathsf{X}|\mathsf{Y},\mathsf{Z}) \tag{2.47}$$
$$H\left(\mathsf{X}|\mathsf{Y}\right) \leq H\left(\mathsf{X}\right) \tag{2.48}$$

Finalmente, tomando promedios en (2.30) se tiene:

$$H(\mathsf{X},\mathsf{Z}|\mathsf{Y}) \leq H(\mathsf{X}|\mathsf{Y}) \tag{2.49}$$

que refleja la noción intuitiva de que eliminar un dato conocido *a priori* (a saber, el valor unívoco de la variable Z) incrementa la incertidumbre respecto del resultado de una medida.

La función entropía constituye una medida estándar para evaluar la calidad de los sistemas de información [76], así como una precursora de otras métricas ampliamente utilizadas [77]-[79]. En relación a la tecnología de las funciones no-clonables físicamente, la entropía y otras cantidades derivadas permiten evaluar sus propiedades de seguridad abstractas más allá de la caracterización física de su solución de implementación, lo cual permite comparar la idoneidad de una PUF como primitiva criptográfica en el contexto de cada aplicación concreta [80], [81].

2.2. Seguridad de la información

Tradicionalmente, la seguridad de la información se ha definido como la propiedad de un fragmento de información en el cual se garantizan las siguientes características a lo largo de todo el ciclo vital de la información, *i.e.*, durante la generación, almacenamiento y posible transmisión de la información [82]:

- **Confidencialidad**, la información es inaccesible o ininteligible para los individuos o sistemas que carecen de la autorización para ello.

- **Integridad**, la información es inalterable sin los permisos oportunos. Esta propiedad implica la capacidad de detectar dichas alteraciones en la información.

- **No-repudio**, tanto la emisión como la recepción de un mensaje pueden probarse fehacientemente.

- **Autenticidad**, un participante en un intercambio de información puede demostrar su identidad (autenticidad de la entidad), y se puede conocer el autor de un mensaje (autenticidad de la información). Este concepto no debe confundirse con la identificación, que es el proceso de asignar una etiqueta a una instancia participante en una comunicación con el fin de distinguirlo de otros participantes (*e.g.*, un nombre de pila sirve para identificar a un individuo tanto como un mote, sin embargo, una huella dactilar autentica a un individuo).

La criptografía es el conjunto de técnicas dirigidas a lograr los objetivos enumerados arriba. Por otra parte, el criptoanálisis estudia las vulnerabilidades de dichas técnicas y su posible subversión; ambas disciplinas integran el campo de la criptología.

2.2.1. Minentropía

La minentropía (H^{m}) es una cantidad relacionada con la entropía de Shannon (sección 2.1.2), definida como el logaritmo de la probabilidad máxima de una variable aleatoria X (*i.e.*, del evento $\mathsf{X} = x$ más probable):

$$H^{\mathrm{m}}(\mathsf{X}) \equiv -\log_2 \left[\max_x (p_x)\right] \tag{2.50}$$

donde $p_x \equiv \mathrm{Prob}(\mathsf{X} = x)$. El NIST define la minentropía como "el mayor valor m de una variable aleatoria X tal que cada observación de X proporciona al menos m bits de información, *i.e.*, la minentropía de X es la cota mínima para el contenido de información de las observaciones potenciales de X". De este modo, la minentropía puede interpretarse como una medida de la impredecibilidad mínima de una variable aleatoria [83]. Esta magnitud cumple:

$$H^{\mathrm{m}}(\mathsf{X}) \leq H(\mathsf{X}) \tag{2.51}$$

con la igualdad si y sólo si la variable X se distribuye uniformemente. La primera parte de este lema es fácil de probar aplicando la desigualdad de Jensen sobre la entropía para obtener la relación $H = -\overline{(\log_2 p)} \geq -\log_2 \overline{p}$, que junto con $\overline{p} \leq p^{\max}$ conduce automáticamente a $H \geq -\log_2 p^{\max} = H^{\mathrm{m}}$. En cuanto a la igualdad para una distribución uniforme, resulta evidente que en ese caso $p_x = p\,\forall x \Rightarrow p^{\max} = p$, y por lo tanto se tiene: $H = -\overline{\log_2 p} = -\log_2 (p^{\max}) = H^{\mathrm{m}}$. Para sistemas conjuntos de variables aleatorias (X, Y) no existe una versión equivalente de la regla de la cadena (2.44) aplicada a la minentropía, sin embargo se puede demostrar:

$$H^{\mathrm{m}}(\mathsf{X}, \mathsf{Y}) \geq H^{\mathrm{m}}(\mathsf{X}) \tag{2.52}$$

lo cual es equivalente al lema $p_{x',y'} \leq p_{x^\star}$, donde $x^\star$ es el valor más probable de la variable X, $p_{x^\star} \equiv \mathrm{Prob}(\mathsf{X} = x^\star) \geq p_x\,\forall x$, y la dupla x', y' es la pareja de valores cuya probabilidad conjunta es máxima, $p_{x',y'} \geq p_{xy}\,\forall x, y$. Este lema puede demostrarse utilizando la hipótesis absurda $p_{x',y'} > p_{x^\star}$: dado que para la probabilidad marginal $p_{x'} = \sum_y p_{x',y}$ se tiene $p_{x'} \geq p_{x',y'}$, aplicando la hipótesis absurda se llega a $p_{x'} \geq p_{x',y'} > p_{x^\star}$, lo cual contradice la definición de $x^\star$.

Estimación de la minentropía en un conjunto de medidas experimentales

Dada una variable aleatoria X que extrae el elemento $x_i \in \mathbf{X}$ con una distribución de probabilidad $p_i \equiv \mathrm{Prob}(\mathsf{X} = x_i)$. Sin pérdida de generalidad, asignamos los índices de los elementos pertenecientes a $\mathbf{X}$ en orden decreciente de probabilidad, $p_0 > p_1 > \ldots > p_{|\mathbf{X}|}$. Definamos ahora la variable aleatoria n_i como el número de veces que se repite la cantidad x_i dado que extraemos N valores aleatorios. Aproximaremos la minentropía (*i.e.*, la máxima probabilidad) a partir de una sucesión de N medidas como $\tilde{p}^{\max} \equiv \tilde{n}^{\max}/N$, donde $\tilde{n}^{\max} \equiv \max\{\tilde{n}_i\}$ y la tilde "$\sim$" representa la estimación de una magnitud como resultado de una medida experimental. Por otra parte, podemos escribir la distribución de las cantidades $\mathsf{n}_i = n_i$ como sigue:

- $n_i = 0$ implica que el valor $\mathsf{X} = x_i$ no aparece en ninguno de los N números extraídos. Dado que la probabilidad de no obtener x_i es $(1-p_i)$, y que sucesivas extracciones de X son independientes, se tiene $\mathrm{Prob}(n_i = 0) = (1 - p_i)^N$.

- $n_i = 1$ implica que uno y sólo uno de los N valores extraídos de la variable X coincidan con $\mathsf{X} = x_i$: esto es, la suma de la probabilidad[7] de que sólo el primero, sólo el segundo,..., sólo el N-ésimo valor extraído de X coincida con x_i, $\mathrm{Prob}(n_i = 1) = Np_i(1 - p_i)^{N-1}$.

- $n_i = 2$ implica que una pareja y sólo una pareja extraída de X coincide con x_i. El número de posibles parejas diferentes en los N elementos extraídos es el número combinatorio $\binom{N}{2}$, de modo que $\mathrm{Prob}(n_i = 2) = \binom{N}{2} p_i^2 (1 - p_i)^{N-2}$.

Esta sucesión se puede iterar para cualquier $0 \leq k \leq N$, obteniendo:

$$\mathrm{Prob}(\mathsf{n}_i = k) = \binom{N}{k} p_i^k \, (1 - p_i)^{N-k} = \underset{N,p_i}{\mathrm{Bin}}(k) \tag{2.53}$$

donde Bin_{N,p_i} es la distribución aleatoria binomial de N número de ensayos y sesgo p. Un cálculo exacto de la distribución de probabilidad para la variable aleatoria $\mathsf{n}^{\max} \equiv \max\{n_i\}$ requiere calcular la probabilidad para cada i-ésimo valor de X de que no haya en la sucesión de valores extraídos ningún $n_j > n_i$, $j \neq i$. Sin embargo, se puede argumentar que para valores $N \gg 1$ esta magnitud estará dominada por

[7] Notar que los eventos "sólo el primer elemento vale x_i", "sólo el segundo elemento vale x_i", etc., son disjuntos y, por tanto, la probabilidad de la unión es la suma aritmética de probabilidades.

la probabilidad máxima de la distribución, p_0: la probabilidad conjunta de obtener respectivamente $\mathsf{n}_0 = k$ y $\mathsf{n}_i = k'$, $i > 0$, será:

$$\text{Prob}(\mathsf{n}_0 = k, \mathsf{n}_i = k') = \underset{N,p_0}{\text{Bin}}(k)\, \underset{N,p_i}{\text{Bin}}(k') \tag{2.54}$$

y la probabilidad de que $\mathsf{n}_i > \mathsf{n}_0$ será la probabilidad acumulada,

$$\text{Prob}(\mathsf{n}_i > \mathsf{n}_0) = \sum_{k=0}^{N} \sum_{k'=k+1}^{N} \underset{N,p_0}{\text{Bin}}(k)\, \underset{N,p_i}{\text{Bin}}(k') \tag{2.55}$$

Utilizando que $N \gg 1$ podemos aproximar las distribuciones binomiales por curvas normales,

$$\underset{N,p}{\text{Bin}}(k) \to \underset{\mu,\sigma}{\text{Norm}}(k) = \frac{1}{\sqrt{2\pi}\sigma}\, e^{-\frac{1}{2}\frac{(k-\mu)^2}{\sigma^2}} \tag{2.56}$$

con $\mu = Np$, $\sigma^2 = Np(1-p)$, y sustituir las sumas en (2.55) por integrales [84]:

$$\begin{aligned} \text{Prob}(n_i > n_0) &\approx \int_{-\infty}^{\infty} dk \int_{k}^{\infty} dk'\, \underset{\mu_0,\sigma_0}{\text{Norm}}(k)\, \underset{\mu_i,\sigma_i}{\text{Norm}}(k') \\ &= \frac{1}{2}\left[\text{erf}\left(\frac{\sqrt{2}}{2}\frac{\mu_i - \mu_0}{\sqrt{\sigma_i^2 + \sigma_0^2}}\right) + 1\right] \end{aligned} \tag{2.57}$$

con $\mu_0 = Np_0$, $\sigma_0^2 = Np_0(1-p_0)$, $\mu_i = Np_i$ y $\sigma_i^2 = Np_i(1-p_i)$, y donde se ha utilizado la tabla de integrales de Ng y Geller, fórmula 13, sección 4.3 [85] con el cambio de variables $x = \sqrt{2}(\mu_i - k)/(2\sigma_i)$, $dx = -\sqrt{2}/(2\sigma_i)\, dk$, $a = -\sigma_i/\sigma_0$, y $b = (\mu_i - \mu_0)/(\sqrt{2}\sigma_0)$. Sustituyendo las expresiones para los promedios y varianzas se tiene:

$$\begin{aligned} \text{Prob}(\mathsf{n}_i > \mathsf{n}_0) &\approx \frac{1}{2}\left[\text{erf}\left(\frac{\sqrt{2}}{2}\frac{Np_i - Np_0}{\sqrt{Np_i(1-p_i) + Np_0(1-p_0)}}\right) + 1\right] \\ &= \frac{1}{2}\left[\text{erf}\left(-\frac{\sqrt{2|p_i - p_0|}}{2\sqrt{p_i(1-p_i) + p_0(1-p_0)}}\sqrt{N}\right) + 1\right] \\ &= \frac{1}{2}\left[\text{erf}\left(-|\alpha|\sqrt{N}\right) + 1\right] \\ &= \frac{1}{2}\left[1 - \text{erf}\left(|\alpha|\sqrt{N}\right)\right] \xrightarrow[N\gg 1]{} 0 \end{aligned} \tag{2.58}$$

siendo $\alpha \in \mathbb{R}$ una constante respecto de N. Así, la probabilidad de que $\mathsf{n}_i > \mathsf{n}_0$ se puede hacer arbitrariamente pequeña aumentando el número de valores extraídos

N. Dado que este argumento es aplicable a cualquier índice $i > 0$, podemos tomar la aproximación $n^{\max} \approx n_0$ y escribir:

$$\text{Prob}(\mathsf{n}^{\max} = k) \approx \binom{N}{k} p_0^k \left(1 - p_0\right)^{N-k} = \underset{N,p_0}{\text{Bin}}(k) \tag{2.59}$$

que es la distribución de probabilidad de una variable $\mathsf{n}^{\max}$ con probabilidad $p^{\max} \approx p_0$. De este modo, dada una sucesión de N valores extraídos de una distribución desconocida $p_x = \text{Prob}(X = x)$, estimaremos la cantidad $\tilde{n}^{\max}$ como el máximo número de repeticiones, $\tilde{n}^{\max} = \max\{\tilde{n}_i\}$, y utilizando el valor medio de la binomial dada en (2.59) escribimos:

$$\tilde{n}^{\max} \approx \overline{n}^{\max} = Np^{\max} \Longrightarrow \tilde{p}^{\max} \approx \tilde{n}^{\max}/N \tag{2.60}$$

Así mismo, la desviación estándar de la probabilidad estimada será:

$$\sigma\left(\tilde{p}^{\max}\right) \approx \sigma\left(\overline{n}^{\max}\right)/N \tag{2.61}$$

Y aplicando el estimador no-sesgado de la desviación del promedio de la binomial (2.59), $\sigma(\overline{n}^{\max}) = \sigma(n^{\max})/\sqrt{N-1}$, así como la estimación $\tilde{p}^{\max}$ dada en (2.60), podemos escribir:

$$\tilde{p}^{\max} = \frac{\tilde{n}^{\max}}{N} \pm c\,\frac{1}{N-1}\sqrt{\frac{\tilde{n}^{\max}}{N}\left(1 - \frac{\tilde{n}^{\max}}{N}\right)} \tag{2.62}$$

donde el parámetro c es un factor de cobertura que permite ajustar el margen de error a una cierta significancia[8] α. Dado (2.59) calcularemos la minentropía como:

$$H^{\text{m}} = -\log_2\left[\frac{\tilde{n}^{\max}}{N} + c\,\frac{1}{N-1}\sqrt{\frac{\tilde{n}^{\max}}{N}\left(1 - \frac{\tilde{n}^{\max}}{N}\right)}\right] \tag{2.63}$$

Para el caso $N \gg 1$ donde la distribución binomial se aproxima a una curva gaussiana, el factor c puede tomarse como el factor de cobertura normal para una significancia α:

$$\begin{aligned} \alpha = 0{,}68 &\Longleftrightarrow c = 1 \\ \alpha = 0{,}95 &\Longleftrightarrow c = 1{,}96 \\ \alpha = 0{,}99 &\Longleftrightarrow c = 2{,}576 \end{aligned} \tag{2.64}$$

[8] Esto es, que la probabilidad de que el valor estimado $\tilde{n}^{\max}$ se aleje del valor real $n^{\max}$ más que $c\sigma(n^{\max})$ sea inferior a $\alpha = 1 - \sum_{k=0}^{c\sigma(n^{\max})} \text{Bin}_{N,p^{\max}}(k)$.

2.2.2. Criptología

El uso de mensajes cifrados y su decodificación ha sido una materia ensayada por la civilización humana desde la antigüedad como ya comentamos en la sección 1.1. Sin embargo, no fue hasta 1949, con la experiencia del éxito vivido por la comunidad criptográfica inglesa durante la II Guerra Mundial al formalizar la —hasta entonces— "intuición criptológica" con el destacado trabajo de matemáticos como Marian Rejewski, Alan Turing o Gordon Welchman (sección 1.1), así como la publicación por parte de C. E. Shannon del trabajo *Communication Theory of Secrecy Systems* [6], que se puede hablar del nacimiento de la criptología como una disciplina científica. A pesar de esto, la consolidación de esta nueva disciplina como una ciencia formal y de indudable interés práctico habría de esperar a la publicación, en 1976, del trabajo *New Directions in Cryptography* de Diffie y Hellman, quienes demostraron que el intercambio de claves secretas podía realizarse de forma segura, fundando así la "criptografía de clave pública" o "asimétrica" [8].

Criptosistemas y seguridad

Un "criptosistema" es un sistema para la transmisión de información segura, en el sentido descrito al comienzo de la sección 2.2. Este comprende una red en la que participan varias instancias (nodos) las cuales son capaces de almacenar y procesar información segura, así como un protocolo criptográfico, que es una sucesión algorítmica de pasos a ejecutar por dos o más instancias para lograr un objetivo específico de seguridad de la información. Las funciones que han de ser aplicadas localmente por cada instancia como requerimientos del protocolo se denominan "primitivas criptográficas" (*e.g.*, cifradores, generadores de números aleatorios, etcétera). Además, un criptosistema incluye una o varias instancias maliciosas denominadas "adversarios" o "atacantes", cuyo objetivo es comprometer las propiedades de seguridad del sistema. Este esquema, ilustrado con la figura 2.1 de forma general, permite evaluar sistemáticamente las propiedades de seguridad de un sistema criptográfico de acuerdo con las siguientes características: "modelo de ataque", "ataque exitoso", y "nivel de seguridad". A continuación se describen cada uno de estos aspectos que caracterizan la seguridad de un sistema:

- **Modelo de ataque**: información del sistema que se considera accesible para un adversario. De forma general, esta debe ser consistente con el principio de

Fig. 2.1.: Esquema general de un criptosistema: en negro las posibles vías de comunicación de Aurora a Blas, en rojo de Blas a Aurora; ambas susceptibles de ser interceptadas o modificadas por un adversario.

Kerchoffs[9], el cual establece que todos los aspectos operacionales de un criptosistema son conocidos por los adversarios, incluyendo detalles prácticos de la implementación de las primitivas criptográficas, así como las distribuciones de probabilidad de cualesquiera procesos aleatorios. Así, el único elemento desconocido para un adversario y sobre el cual se apoya la seguridad completa del criptosistema debe ser una información secreta (típicamente un valor numérico) que se denomina "clave secreta". Los modelos de ataque fundamentales, ordenados en sentido creciente en cuanto a su capacidad para amenazar un sistema de comunicación, son:

- Ataque de texto cifrado (*Known-ciphertext Attack*, KCA): un adversario dispone de acceso a algunos mensajes cifrados aleatorios sin conocer completamente el texto plano (sin cifrar) correspondiente a los mismos. A menudo se considera el modelo de ataque más débil.

- Ataque de texto plano (*Known-plaintext Attack*, KPA): un adversario dispone de acceso a algunos mensajes aleatorios en forma de texto plano, así como a sus correspondientes textos cifrados.

- Ataque de texto dirigido (*Chosen-plaintext Attack*, CPA): un adversario puede acceder a uno o varios mensajes planos cualesquiera y obtener sus correspondientes textos cifrados a través de un "oráculo de cifrado",

[9] Criptógrafo holandés que propuso, en 1883, una lista de las características que debía poseer un buen sistema criptográfico. Muchas de estas características ideales están obsoletas hoy (*e.g.*, se pide que el criptograma sea "transmisible por telégrafo", o que la clave sea "fácilmente memorizable sin anotarla y fácil de cambiar"). Sin embargo, demostró una notable clarividencia al postular que "el conocimiento público de los detalles funcionales del criptosistema no debe comprometer la seguridad del cifrado", y proporcionó una incipiente distinción entre la seguridad teórica y práctica al exigir que un sistema fuera "si no teóricamente, prácticamente seguro" [86].

esto es, un objeto formal (por ejemplo, algún defecto del protocolo de seguridad) que permite al atacante cifrar mensajes arbitrarios sin necesidad de conocer la clave secreta.

– Ataque de texto cifrado dirigido (*Chosen-ciphertext Attack*, CCA): un adversario puede elegir uno o varios mensajes cifrados cualesquiera y obtener sus correspondientes textos planos utilizando un "oráculo de descifrado", que permite al atacante recuperar mensajes planos a partir de texto cifrado sin necesidad de conocer la clave.

A pesar de esta clasificación, el modelo de adversario en una aplicación real puede ser una combinación de las capacidades enumeradas arriba, o una forma parcial de las mismas. Por ejemplo, se podría considerar un atacante KCA que es capaz de acceder únicamente a una parte del criptograma, o bien que conozca *a priori* algunos bits de la clave, etc. En el caso particular de sistemas distribuidos como el IoT, también pueden concebirse adversarios con capacidades respecto de la implementación física de un cierto criptosistema. Los principales modelos de ataque de este género son los ataques de canal paralelo (sección 1.1.2). Estos se pueden clasificar atendiendo al fenómeno físico responsable de la filtración de información, entre los cuales destacan:

– Análisis de emisión electromagnética: consiste en la detección de las emisiones electromagnéticas de un dispositivo mientras ejecuta una operación criptográfica para correlacionar esta con los estados internos del mismo, buscando revelar la clave secreta. A pesar de que el fenómeno de emisión es inherente a la presencia de cualquier corriente eléctrica, existen algunas contramedidas, como por ejemplo reducir la frecuencia de operación y, en su caso, filtrar las componentes de alta frecuencia del dispositivo a fin de reducir la cantidad de energía emanada, o utilizar equipamiento apantallado eléctricamente. A pesar de que este fenómeno era conocido por la inteligencia estadounidense desde finales de la II Guerra Mundial, su difusión pública se debe al trabajo del investigador holandés Wim van Eck, quien en 1985 logró capturar y reproducir en un monitor la imagen mostrada en una pantalla remota mediante el análisis cuidadoso de la radiación emitida por cada píxel de la pantalla [87]. En el año 2013, Merli *et al.* mostraron la aplicación de esta técnica para extraer las frecuencias características de oscilación de una PUF basada en osciladores de anillo [88] (sección 4).

– Análisis de retardo: por diseño, el procesamiento de entradas diferentes puede implicar tiempos de computación diferentes, especialmente en sistemas optimizados para reducir el tiempo de procesamiento total. Por ejemplo, una interfaz segura que solicite y verifique la contraseña ingresada por un usuario puede detenerse cuando encuentra el primer carácter erróneo, a fin de optimizar el tiempo medio de cálculo. Sin embargo, en este ejemplo, un adversario capaz de monitorizar el tiempo dedicado por la interfaz a computar la contraseña podría descartar caracteres erróneos de forma sistemática, reduciendo la complejidad de encontrar la contraseña desde el caso idealmente exponencial a meramente lineal, $\mathcal{O}(M^n) \longrightarrow \mathcal{O}(Mn)$, donde M es el número de símbolos del alfabeto en el cual está escrita la contraseña, n es el número de caracteres de la contraseña y "$\mathcal{O}$" es la notación asintótica de Landau. Este tipo de ataques fue propuesto por el estadounidense Paul Kocher en 1996, quien lo aplicó con éxito a los esquemas de intercambio de claves de Diffie-Hellman y RSA [89].

– Análisis de consumo de potencia: de forma similar al caso anterior, las curvas de consumo de potencia de un dispositivo mientras ejecuta un algoritmo criptográfico serán en general diferentes en función de los datos de entrada a procesar. Un análisis de potencia busca correlacionar estos perfiles de consumo con los datos ingresados en el dispositivo, en particular el valor de una clave secreta. De nuevo, los trabajos pioneros en esta técnica se deben a Kocher *et al.*, que en 1999 la aplicaron con éxito al cifrador DES [90]. En 2010, Karakoyunlu *et al.* aplicaron esta técnica al software de posprocesado de una función no-clonable físicamente [91], y posteriormente en 2014 Becker y Kumar fueron capaces de crear un modelo de PUF de árbitro midiendo cuidadosamente su consumo de potencia [92].

Además, existen otros modelos de ataques físicos activos que actúan sobre el dispositivo, con el objetivo de inducir un comportamiento extraño o forzarlos activamente a filtrar información. Los principales modelos de esta clase de ataque son [93]:

– Ataque de inyección de fallos (*Fault Injection Attack*, FIA). Esta técnica trata de vulnerar un dispositivo físico criptográfico mediante la exposición de este a un estímulo que excede los límites de operación para los cuales fue diseñado, típicamente temperaturas extremas, pulsos de tensión

eléctrica o exposición a radiación (incluyendo haces láser y ultravioleta). Esto tiene por objeto que el mal funcionamiento de los componentes electrónicos filtre información secreta, o bien desactivar contramedidas dispuestas para otro tipo de acceso ilegítimo, por ejemplo medidas de protección frente a ataques de canal paralelo [94].

  - Ataques invasivos sobre el chip, esto es, acceder físicamente al sustrato de silicio para modificar la estructura del circuito integrado tal y como se diseñó originalmente. Esta clase de ataques requieren de equipamiento especializado y son extremadamente costosos de realizar. Dado que el comportamiento de una función no-clonable físicamente depende de su microestructura física, estas resultan enormemente sensibles a la manipulación, por lo cual se puede considerar que la tecnología PUF es robusta frente a este tipo de ataques ya que son fácilmente detectables. No obstante, la electrónica de posprocesado sí puede ser vulnerable a la intrusión. Para mitigar esta, Gassend *et al.* propusieron en 2008 una estrategia de diseño en la cual la lógica de posprocesado se encuentra físicamente envuelta por los elementos físicos que forma parte de la PUF, dificultando el acceso por parte de un adversario a una sección del circuito concreta sin dejar evidencias de manipulación [95].

- **Ataque exitoso**: un criptosistema se puede considerar destruido si el adversario es capaz de acceder a la clave secreta. Sin embargo, en función de los objetivos de seguridad perseguidos, el sistema se puede ver comprometido bajo circunstancias más débiles y asequibles. En general, se produce una ataque exitoso cuando se logra violar la seguridad de un sistema, de manera que alcanza sus objetivos previstos. Estos objetivos pueden variar y podrían incluir:

  - Acceso no autorizado: el atacante obtiene acceso a sistemas, redes o datos sin permiso. Esto podría implicar robo de información confidencial o datos sensibles.

  - Interrupción del servicio: el atacante logra interrumpir o deshabilitar los servicios normales de una red o sistema, lo que podría resultar en pérdida de disponibilidad.

  - Manipulación de datos: el atacante altera, corrompe o destruye datos, afectando la integridad de la información almacenada.

  - Elevación de privilegios: el atacante logra obtener privilegios más altos de los que le fueron asignados inicialmente, lo que le otorga un mayor control sobre el sistema.

  - Inyección de código: el atacante introduce código malicioso en un sistema con el objetivo de ejecutar comandos no autorizados.

  - Suplantación de identidad (*phishing*): el atacante engaña a usuarios para obtener credenciales de acceso, lo que le permite acceder a sistemas o datos confidenciales.

- **Nivel de seguridad**: capacidades (potencia de cálculo y tiempo) de que dispone un adversario para atacar el sistema. En general, se pueden distinguir tres niveles de seguridad en función de si un criptosistema es robusto frente a un adversario dotado de las siguientes capacidades:

  - Seguridad computacional: un sistema es "computacionalmente seguro" cuando un adversario con unos recursos (tiempo y potencia de cálculo) determinados no es capaz de lograr un ataque exitoso (dado un cierto modelo de ataque).

  - Seguridad probable: un criptosistema se dice "probablemente seguro" cuando se demuestra matemáticamente que lograr un cierto ataque equivale a resolver un determinado problema (*e.g.*, el cálculo del logaritmo discreto en un cuerpo finito), el cual es considerado irresoluble en la práctica.

  - Seguridad incondicional: un sistema es "incondicionalmente seguro" cuando un adversario con recursos ilimitados no es capaz de lograr un ataque exitoso (dado un cierto modelo de ataque).

De acuerdo con esto, un criptosistema se puede calificar como "seguro" en el sentido de que un adversario con unas ciertas capacidades no logrará un ataque exitoso dado un cierto modelo de ataque.

Protocolos criptográficos

Un protocolo criptográfico contribuye a la superficie de ataque de un criptosistema, y un adversario puede abusar de los defectos de diseño de un protocolo para comprometer un sistema de seguridad sin necesidad de atacar directamente las primitivas criptográficas empleadas por las instancias participantes. Por ejemplo, un sistema de autenticación basado en el uso de contraseñas donde el verificador no exige su renovación periódica, es susceptible de ser explotado por un adversario que logre interceptar la contraseña correspondiente a una instancia legítima, pudiendo posteriormente suplantar a esta frente al verificador. En este caso, el adversario habrá obtenido acceso a los recursos reservados a un cierto nodo en una red de comunicaciones sin necesidad de vulnerar el algoritmo de autenticación implementado en el sistema. Dado un modelo de adversario y un tamaño de la red, un protocolo se puede describir formalmente y analizar desde el punto de vista de la seguridad de forma automática por *software* de cálculo simbólico, esto es, un programa que ensaye todas las combinaciones posibles de símbolos de acuerdo con una lista de reglas formales, descartando —o confirmando— la existencia de ataques, *i.e.*, sucesiones de pasos, legítimos desde el punto de vista del protocolo, que sin embargo llevan el sistema a un estado definido como "comprometido" [96], [97]. En este trabajo estudiamos las funciones no-clonables físicamente aplicables a IoT (*i.e.*, compactas y eficientes en términos de seguridad/consumo) únicamente como primitivas criptográficas, suponiendo que son implementadas en protocolos criptográficos reconocidos como seguros.

Criptografía de clave secreta (simétrica)

Dados un espacio de mensajes $\mathbf{X}$, criptogramas (o mensajes cifrados) $\mathbf{Y}$ y claves $\mathbf{K}$, se dice que un par de funciones cifrador/descifrador $\mathrm{Encod}_s(x) = y$, $\mathrm{Decod}_s(y) = x$ con $\mathrm{Decod}_s = \mathrm{Encod}_s^{-1}$, $x \in \mathbf{X}$, $y \in \mathbf{Y}$, $s \in \mathbf{K}$ constituye un esquema criptográfico simétrico (o de clave secreta) si la clave s debe ser mantenida en secreto para cualquier agente ajeno al proceso comunicativo a fin de preservar la confidencialidad del mensaje. En la figura 2.2 se representa la aplicación de un sistema de cifrado utilizando un esquema de clave secreta, en el cual el emisor envía un mensaje cifrado al receptor, quien puede recuperar el contenido original utilizando la misma clave s utilizada por el emisor para cifrar el mensaje. Un sistema así se caracteriza por la necesidad de un canal seguro, el cual permite que emisor y receptor compartan una misma clave secreta. Uno de los principales inconvenientes

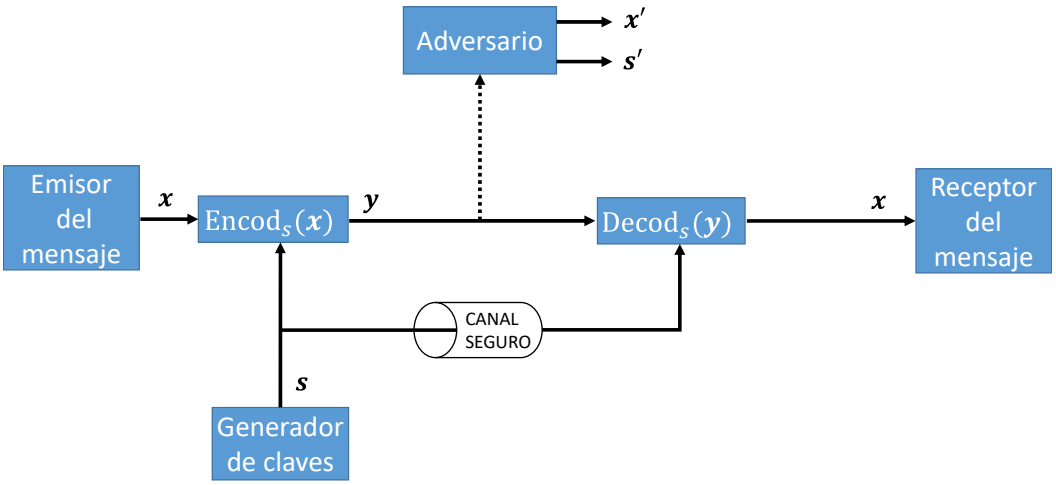


Fig. 2.2.: Representación esquemática de una comunicación secreta cifrada con un método de clave secreta. Un adversario es capaz de "romper" el criptosistema si logra recuperar la clave secreta ($s' = s$) y el mensaje plano ($x' = x$).

de esta solución radica en encontrar un método eficiente para que todas las instancias implicadas compartan una misma clave secreta ("distribución segura de claves") [86].

Es posible estudiar las condiciones bajo las cuales un cierto criptosistema es incondicionalmente seguro de forma general: sea la variable aleatoria X que selecciona mensajes en el espacio $\mathbf{X}$ y un adversario modelado mediante una variable aleatoria Y que selecciona criptogramas candidatos en $\mathbf{Y}$, la condición de cifrado perfecto se define como la independencia estadística entre el mensaje plano (sin cifrar) y el criptograma:

$$\text{Prob}(\mathsf{X} = x|\mathsf{Y} = y) = \text{Prob}(\mathsf{X} = x)\ \forall x \in \mathbf{X},\ y \in \mathbf{Y} \tag{2.65}$$

donde $y = \text{Encod}_s(x)$ es el criptograma que codifica para el mensaje x, esto es, el conocimiento del texto cifrado no proporciona ninguna información sobre el texto original. Esta condición puede expresarse también en términos de la entropía asociadas a cada variable, $H(\mathsf{X}|\mathsf{Y}) = H(\mathsf{X})$. Si la clave secreta es generada mediante una variable aleatoria S, para cualquier sistema criptográfico simétrico se tiene $H(\mathsf{X}|\mathsf{Y}) \leq H(\mathsf{S})$ [98], e imponiendo la condición de cifrado perfecto:

$$H(\mathsf{S}) \geq H(\mathsf{X}) \tag{2.66}$$

Esta desigualdad se denomina "cota fundamental de Shannon para el cifrado perfecto", y constituye una condición necesaria, si bien insuficiente, del cifrado perfecto [99].

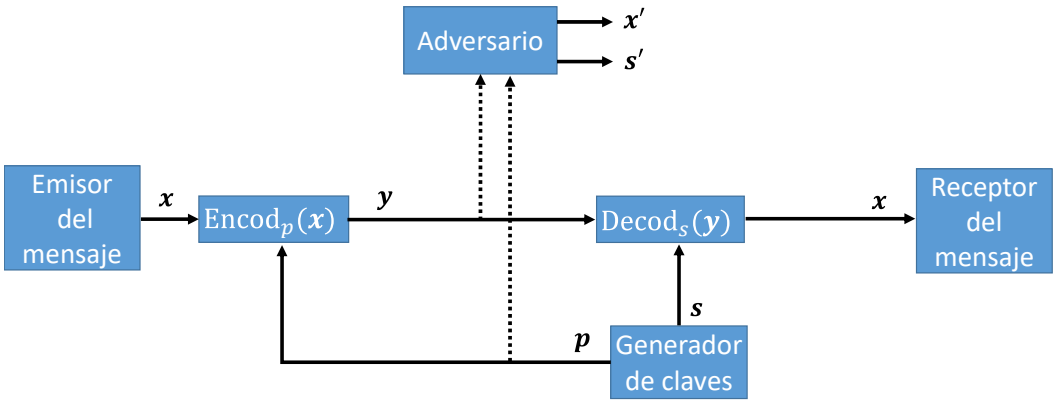


Fig. 2.3.: Representación esquemática de una comunicación secreta cifrada con un método de clave pública. Este sistema se considera comprometido si un adversario logra recuperar la clave secreta ($s' = s$) y el mensaje plano ($x' = x$).

Criptografía de clave pública (asimétrica)

Un esquema de clave pública está constituido por un par de funciones Encod_p, Decod_s (o recíprocamente, Encod_s, Decod_p), $p, s \in \mathbf{K}$ donde p constituye una clave pública y s la clave secreta (o privada) del sistema, con la propiedad de que el conocimiento de la clave pública proporciona poca o ninguna información sobre la clave secreta. Esta noción se circunscribe a un nivel de seguridad, *i.e.*, a las capacidades de un adversario, de tal manera que si este es modelado por una variable aleatoria K definida en el espacio de claves $\mathbf{K}$, un esquema de criptografía asimétrico debe cumplir $\mathrm{Prob}(\mathsf{K} = s|p) \approx \mathrm{Prob}(\mathsf{K} = s) \approx 0$. En la figura 2.3 se representa un sistema de cifrado de clave pública. Este esquema permite la comunicación secreta sin compartir una clave y, por lo tanto, sin necesidad de un canal seguro. En su lugar, una instancia receptora distribuye su clave pública p en un medio potencialmente inseguro, de forma que cualquier nodo de la red puede utilizar la función Encod_p para enviar un mensaje cifrado, el cual sólo puede ser decodificado por el receptor utilizando su clave secreta s y la función Decod_s [86].

Resulta notable que la criptografía asimétrica sea conceptualmente análoga al funcionamiento de un candado físico, donde la operación de hacer una información segura (cerrar el candado) es accesible para cualquier nodo del sistema, mientras que sólo el poseedor de la llave adecuada puede acceder a dicha información (abrir el candado).

Autenticación de entidades

La autenticación es un objetivo de seguridad de la información en el que una instancia solicitante A es capaz de presentar a otro participante verificador B una cierta información tal que B es capaz de: (i) conocer la identidad de A (en el sentido de que es capaz de identificarlo, *i.e.*, "etiquetarlo" de forma unívoca respecto de otros participantes potenciales en la comunicación), y (ii) que A ha participado activamente en la generación de dicha información. El objetivo parcial en el que el requisito (ii) no se cumple se denomina "autenticación de la información" [86], [100]. En general, se pueden distinguir dos esquemas de identificación de entidades:

- Autenticación débil: basada en que el solicitante y el verificador comparten un secreto (*i.e.*, una contraseña[10]) que se mantiene invariante en el tiempo, de modo que el solicitante puede autenticarse presentando al verificador el conocimiento de dicha contraseña. Este esquema tiene inconvenientes evidentes a propósito de la manera en que se comunica la información secreta, siendo vulnerable a ataques sencillos sobre el protocolo de comunicación como ataques de repetición o ataques de diccionario.

- Autenticación fuerte: la instancia solicitante demuestra el conocimiento de un secreto compartido con el verificador sin enviar este a través del canal de comunicación. En su lugar, el verificador envía un "reto" a la instancia solicitante, la cual procesa este utilizando la información secreta de que dispone para defender su identidad, y reenvía una "respuesta" al verificador. La instancia verificadora puede comprobar que la respuesta es correcta dada la identidad reclamada por el solicitante procesando él mismo el reto mediante la información secreta del solicitante (que, tal y como se ha indicado, es compartida). Debido a que la única información transmitida entre ambas instancias a través del canal de comunicación es un par reto-respuesta, la autenticación fuerte es referida también "autenticación CRP" (*Challenge-Response pair*, CRP). Según el esquema criptográfico utilizado se puede distinguir:

  - Autenticación CRP basada en criptografía de clave secreta: la instancia solicitante construye la respuesta encriptando el reto enviado por el

[10] Utilizamos "contraseña" como un caso particular de "clave"; se pueden distinguir de manera informal aduciendo que una contraseña es una clave adaptada a su uso directo por seres humanos (*i.e.*, relativamente estable en el tiempo y manejable en cuanto al número de caracteres).

verificador, el cual comprobará que el mensaje cifrado se corresponde con la clave secreta poseída únicamente por el solicitante.

- Autenticación CRP basada en criptografía de clave pública: la autenticación se basa en que el solicitante demuestra estar en posesión de una cierta clave secreta (s), dado que el verificador conoce la clave pública (p) asociada. De nuevo, se pueden distinguir dos casos en función del esquema asimétrico utilizado: si la clave secreta se utiliza para descifrar un mensaje encriptado con la clave pública (Encod_p, Decod_s), el verificador genera un reto y lo envía cifrado al solicitante, quien lo devuelve descifrado demostrando estar en posesión del binomio (p, s) utilizado para la encriptación. Recíprocamente, se puede utilizar un esquema de firma digital (*i.e.*, la clave secreta es utilizada para cifrar y la pública para descifrar, Encod_s, Decod_p). Ahora el verificador envía un reto al solicitante, el cual lo reenvía cifrado utilizando la clave secreta. Finalmente, la autenticación será exitosa si el verificador decodifica correctamente la respuesta utilizando la clave pública asociada al solicitante.

Funciones *hash* y funciones MAC

Una función *hash* es una primitiva criptográfica que relaciona cada entrada (x) formada por una cadena arbitrariamente larga de bits con una salida (y) formada por una cadena de bits de longitud fija, $y = \mathrm{Hash}(x)$. Además, la función Hash debe tener la propiedad de ser computacionalmente difícil de invertir (*i.e.* de encontrar la preimagen x de un cierto valor y), y además la distribución de $\mathrm{Hash}(x)$ debe ser uniforme (*i.e.*, dada una cierta métrica, la probabilidad de que dos respuesta y_1, y_2 estén cerca no está condicionada por la cercanía de sus entradas x_1, x_2). Los principales usos de las funciones *hash* en protocolos criptográficos son:

- Almacenamiento seguro de claves: generalmente, los servicios de control de acceso basados en contraseñas no almacenan estas como texto plano, sino que conservan únicamente el resultado de aplicar una cierta función *hash* a cada contraseña, de forma que el proceso de verificación se realiza comparando el *hash* de la contraseña tal y como es ingresada por el usuario con la base de datos del servicio. De este modo, en caso de que los datos almacenados en el verificador sean filtrados a través de una brecha de seguridad, las contraseñas

de acceso permanecerán inaccesibles debido a la no-invertibilidad de la función *hash*.

- Integridad (o autenticidad) de la información: el receptor de un cierto mensaje x' puede verificar que este no ha sido modificado durante la transmisión respecto del mensaje original x enviado por el emisor, calculando la función *hash* $y' = \mathrm{Hash}(x')$. Para ello, el emisor típicamente envía el resultado de aplicar la función al mensaje original, $y = \mathrm{Hash}(x)$, junto con el propio mensaje, a fin de que el receptor pueda verificar $y' = y$. Este protocolo de autenticación de la información tiene una vulnerabilidad en el caso de que un adversario intercepte la dupla "mensaje-*hash*", lo que le permitiría remitir un mensaje diferente del original x'' junto con su respectivo valor $y'' = \mathrm{Hash}(x'')$. Una protección eficaz y conveniente frente a este ataque es el empleo de un código de autenticación de mensajes (*Message Authentication Code*, MAC). Una función MAC es una aplicación criptográfica de clave secreta que equivale a una función *hash*, la cual sólo pueda ser calculada sobre una entrada x si se dispone de una cierta clave secreta k, $y = \mathrm{MAC}(x, k)$. De este modo, si emisor y receptor comparten[11] una clave k que es desconocida para el adversario, este no será capaz de suplantar el par "mensaje-MAC".

Firma digital

La firma digital es una primitiva criptográfica cuyo objetivo es garantizar la autenticación de la información, su integridad, y no-repudio, de forma análoga a la firma manuscrita. En general, el proceso de firma digital por parte de una instancia "i" consiste en combinar alguna información secreta privada, s_i, con un mensaje x (que pertenece a un espacio de mensajes $\mathbf{X}$, $x \in \mathbf{X}$) para dar lugar a un objeto $f_i(x)$ denominado "firma digital", $f_i(x) = \mathrm{Firma}(x, s_i)$. En cualquier instante futuro, una instancia verificadora puede recuperar la dupla x, i aplicando una "función de verificación", $(x, i) = \mathrm{Verif}[f_i(x)]$, que es pública y no involucra ninguna información secreta; notar además que la función de verificación no revela el secreto s_i. De este modo la entidad verificadora pueda garantizar que la instancia i no reniega de la autoría del mensaje x (no-repudio), que una cierta instancia i' posee en efecto

[11] Naturalmente, es necesario que esta clave sea compartida a través de un método seguro, y existiendo tal se puede argumentar que el emisor podría simplemente comunicar al receptor el valor del *hash* que debe esperar para neutralizar el ataque. Sin embargo, esto requeriría de una comunicación segura cada vez que se envía un nuevo mensaje, mientras que el uso de un MAC permite que esta comunicación segura se de únicamente una vez.

la autoría del mensaje x, $i' = i$ (autenticación de la información), o bien que un mensaje x' es en efecto el mismo que fue firmado por i, $x' = x$ (integridad de la información). Típicamente, una firma digital se construye en un esquema de criptografía de clave pública, donde la instancia i consta de una clave secreta s_i, una clave pública p_i y un par de funciones Encod_{s_i}, Decod_{p_i}. De este modo, se define la función Firma como $f_i(x) = \mathrm{Firma}(x, s_i) \equiv \mathrm{Encod}_{s_i}(x)$; y la función de verificación como $(x, i) = \mathrm{Verif}[f_i(x)] = (\mathrm{Decod}_{p_i}[f_i(x)], \arg(p_i))$, donde la aplicación "arg" es simplemente la identidad $i = \arg(p_i)$.

2.3. Funciones no-clonables físicamente

Una función no-clonable físicamente (PUF) es una relación entre una serie de estímulos aplicados a un elemento físico (instancia PUF), y las respuestas de este, tal que se cumplen las siguientes propiedades formales:

- **Unicidad**, esta es la propiedad de que la relación estímulos-respuestas de dos instancias PUF concretas permanezcan distinguibles a lo largo del tiempo.

- **Reproducibilidad**, esta es la propiedad de que la relación estímulos-respuestas de una instancia PUF concreta permanezca invariable en el tiempo.

- **No-clonabilidad física**, esta es la propiedad de que la estructura física responsable de generar la funcionalidad estímulos-respuestas en cada instancia PUF sea imposible de replicar hasta el punto de comprometer la unicidad de la PUF.

Podemos formalizar esta noción de PUF a partir del concepto de "función física", entendiendo esta como una "función de transferencia" $\mathcal{F}$ que relaciona un espacio de estímulos físicos $\boldsymbol{\Xi}$, con un espacio de respuestas físicas $\boldsymbol{\Psi}$:

$$\begin{aligned} \mathcal{F} : \boldsymbol{\Lambda} \times \boldsymbol{\Xi} &\to \boldsymbol{\Psi} \\ (\lambda, \xi) &\mapsto \mathcal{F}(\lambda, \xi) = \Psi|\lambda \end{aligned} \tag{2.67}$$

donde $\lambda \in \boldsymbol{\Lambda}$ es un índice extraído de una variable aleatoria Λ con densidad de probabilidad $p_\lambda d\lambda \equiv \mathrm{Prob}(\lambda < \Lambda < \lambda + d\lambda)$, el cual representa la construcción de una instancia concreta de un dispositivo físico $\mathcal{F}$, y queda determinado durante el proceso de fabricación con la propiedad de que no existen dos instancias diferentes

tales que sus índices λ coincidan. Esto se puede formalizar identificando sin pérdida de generalidad el espacio $\mathbf{\Lambda} \equiv (0,1) \subset \mathbb{R}$, de tal manera que un valor concreto $\lambda_0 \in \mathbf{\Lambda}$ constituye un conjunto de medida nula, y por lo tanto $\mathrm{Prob}(\Lambda = \lambda_0) = 0$. La cantidad $\xi \in \mathbf{\Xi}$ representa un estímulo físico, y $\Psi|\lambda$ es una variable aleatoria que tiene asociada una densidad de probabilidad:

$$\begin{aligned} p_{\psi|\lambda}\, d\psi &\equiv \mathrm{Prob}(\psi < \Psi|\lambda < \psi + d\psi) \\ &= \mathrm{Prob}(\psi < \mathcal{F}(\lambda, \xi) < \psi + d\psi) \end{aligned} \tag{2.68}$$

con $\psi \in \mathbf{\Psi}$. Así, el objeto formal $\mathcal{F}(\lambda, \xi)$ será una variable aleatoria sobre el espacio $\mathbf{\Psi}$ que captura la posible variabilidad de las respuestas para una misma instancia y un mismo estímulo, y simboliza la información necesaria y suficiente que requiere un fabricante para producir una instancia $\mathcal{F}(\lambda = \lambda_0, \xi)$ de un cierto diseño. En este trabajo definimos las funciones no-clonables físicamente de manera general como funciones físicas dotadas de una interfaz digital de entrada/salida, utilizando las funciones "conversor digital-analógico" (*Digital-Analog Converter*, DAC):

$$\begin{aligned} \mathrm{DAC}: \mathbf{X} &\to \mathbf{\Xi} \\ \vec{x} &\mapsto \mathrm{DAC}(\vec{x}) = \xi \end{aligned} \tag{2.69}$$

que relaciona de forma inyectiva (*i.e.*, invertible) un espacio de palabras binarias $\mathbf{X}$ con un espacio de estímulos físicos, $\mathbf{\Xi}$, interpretables por una determinada función física. Si cada vector binario $\vec{x}$ está formado por M_b símbolos binarios, $\vec{x} = (x_1, \cdots, x_{M_b})$, el conjunto $\mathbf{X}$ será isomorfo al conjunto (*i.e.*, se puede identificar con) $\{0,1\}^{M_b}$. El objeto $\{0,1\}$ se puede convertir en el cuerpo finito de Galois de dos elementos (que llamaremos $\mathbb{B}$) dotándolo de una operación suma, a saber, la suma módulo 2 (dada por la operación lógica XOR) con elemento neutro "0", y de un producto interno dado por la operación lógica AND con elemento neutro "1". Ahora, $\mathbb{B}$ puede utilizarse como soporte para un espacio vectorial dado por $(\mathbf{X}, \mathbb{B}, \oplus, *)$, donde abusamos de los símbolos "$\oplus$" (XOR) para referirnos a la suma de vectores:

$$\vec{a} \oplus \vec{b} \equiv (a_1 \oplus b_1, \cdots, a_{M_b} \oplus b_{M_b}) = \vec{c} \in \mathbf{X} \tag{2.70}$$

y del símbolo "$*$" (AND) para referirnos al producto por un escalar:

$$a * \vec{b} \equiv (a * b_1, \cdots, a * b_{M_b}) = \vec{c} \in \mathbf{X} \tag{2.71}$$

De esta forma, $\mathbf{X}$ será un espacio vectorial finito de tamaño $|\mathbf{X}| = 2^{M_b}$. Finalmente, podemos dotar a este espacio de una métrica definiendo una distancia, esto es, una función dist:

$$\begin{aligned} \mathrm{dist}: \mathbf{X} \times \mathbf{X} &\to \mathbb{R} \\ \vec{a}, \vec{b} &\mapsto \mathrm{dist}\left(\vec{a}, \vec{b}\right) = c \end{aligned} \tag{2.72}$$

con las propiedades de que es definida positiva, nula si y sólo si $\vec{a}$ es igual a $\vec{b}$, y cumple la desigualdad triangular:

$$\mathrm{dist}\left(\vec{a}, \vec{b}\right) \geq 0 \; \forall \, \vec{a}, \vec{b} \tag{2.73}$$

$$\mathrm{dist}\left(\vec{a}, \vec{b}\right) = 0 \iff \vec{a} = \vec{b} \tag{2.74}$$

$$\mathrm{dist}\left(\vec{a}, \vec{b}\right) + \mathrm{dist}\left(\vec{b}, \vec{c}\right) \geq \mathrm{dist}\left(\vec{a}, \vec{c}\right) \tag{2.75}$$

La distancia euclídea estándar definida entre un par de vectores cumple estas propiedades, sin embargo, la noción de métrica más utilizada en el contexto de espacios vectoriales sobre un cuerpo binario es la "distancia de Hamming" (*Hamming Distance*, HD), definida como[12]:

$$\mathrm{HD}\left(\vec{a}, \vec{b}\right) = \sum_{i=1}^{M_b} a_i \oplus b_i \tag{2.76}$$

donde el sumatorio se refiere a la suma estándar de números enteros. Se puede comprobar que esta definición cumple los requisitos de una métrica descritos arriba: dado que las cantidades $a, b \in \{0, 1\}$ cumplen $a \oplus b \geq 0$, la suma en (2.76) es definida positiva. Además, $a \oplus b = 0 \iff a = b$, y dada la condición anterior se tiene que $\mathrm{HD}(\vec{a}, \vec{b}) = 0 \iff \vec{a} = \vec{b}$. Por último, en cuanto a la desigualdad triangular, se tiene que dados tres vectores $\vec{a}$, $\vec{b}$ y $\vec{c}$ podemos escribir:

$$\mathrm{HD}(\vec{a}, \vec{c}) - \mathrm{HD}(\vec{a}, \vec{b}) - \mathrm{HD}(\vec{b}, \vec{c}) = \sum_i \left(a_i \oplus c_i - a_i \oplus b_i - b_i \oplus c_i\right) \tag{2.77}$$

Los posibles valores de cada sumando pueden calcularse aplicando la tabla de verdad dada en 2.1. Dado que cada sumando es menor o igual que cero, la suma cumplirá $\sum_i \left(a_i \oplus c_i - a_i \oplus b_i - b_i \oplus c_i\right) \leq 0$ y, por lo tanto, queda demostrada la desigualdad triangular para la distancia Hamming.

[12] Estrictamente, esta cantidad se puede definir sobre un par de vectores de símbolos cualesquiera de igual longitud, y su valor es igual al número de símbolos diferentes entre ambos vectores, resolviendo para cada posición en el vector; por ejemplo, las palabras *RETOS* e *HILOS* difieren en el primer, segundo y tercer carácter, de modo que su distancia de Hamming es igual a 3. En cambio, *HILOS* e *HITOS* se diferencian sólo en el tercer carácter, de modo que su distancia de Hamming es igual a 1.

Tab. 2.1.: Todos los posibles resultados de los sumandos dados en (2.77).

a	b	c	$a \oplus c - a \oplus b - b \oplus c$	
0	0	0	$0 - 0 - 0 =$	0
0	0	1	$1 - 0 - 1 =$	0
0	1	0	$0 - 1 - 1 =$	-2
0	1	1	$1 - 1 - 0 =$	0
1	0	0	$1 - 1 - 0 =$	0
1	0	1	$0 - 1 - 1 =$	-2
1	1	0	$1 - 0 - 1 =$	0
1	1	1	$0 - 0 - 0 =$	0

Por otro lado, la función "conversor analógico-digital" (*Analog-Digital Converter*, ADC):

$$\begin{aligned} \mathrm{ADC}: \ & \boldsymbol{\Psi} \to \mathbf{Y} \\ & \vec{\psi} \mapsto \mathrm{ADC}\left(\vec{\psi}\right) = \vec{y} \end{aligned} \tag{2.78}$$

relaciona un espacio $\boldsymbol{\Psi}$ de respuestas físicas producidas por una determinada función física con un espacio de palabras binarias ($\mathbf{Y}$). A diferencia de la conversión digital a analógico, ahora la relación ADC será en general suprayectiva, capturando así la posibilidad de que el conversor sea incapaz de resolver entre varias medidas físicas si la variación en estas no es lo suficientemente grande, devolviendo en este caso una misma respuesta binaria. Si las respuestas binarias constan de N_b bits, podemos repetir la argumentación anterior y construir un espacio métrico de tamaño $|\mathbf{Y}| = 2^{N_b}$, $(\mathbf{Y}, \mathbb{B}, \oplus, *)$, junto con la distancia de Hamming definida en (2.76).

Esto nos permite definir formalmente una función no-clonable físicamente como una aplicación binaria aleatoria $\mathcal{P}$ que conecta un conjunto de desafíos $\mathbf{X}$ con un espacio de respuestas binarias $\mathbf{Y}$, y que representa la dispersión de las respuestas para una misma instancia PUF y un mismo reto:

$$\begin{aligned} \mathcal{P}: \ & \boldsymbol{\Lambda} \times \mathbf{X} \to \mathbf{Y} \\ & (\lambda, \vec{x}) \mapsto \mathcal{P}(\lambda, \vec{x}) \equiv \mathrm{ADC}\left(\mathcal{F}\left[\lambda, \mathrm{DAC}(\vec{x})\right]\right) = \mathsf{Y}|\lambda \end{aligned} \tag{2.79}$$

donde ahora el índice λ representa una construcción concreta (*i.e.*, una instancia) de un diseño PUF, $\vec{x} = (x_1, \cdots, x_{M_b}) \in \mathbf{X}$ representa un desafío (o reto) binario adecuado para ser enviado a través de un canal de comunicación digital o de ser almacenado en un dispositivo de memoria, y que se utilizará para evaluar el dispositivo PUF, y $\mathsf{Y}|\lambda$ es una variable aleatoria discreta que tiene asociada una distribución de probabilidad $p_{\vec{y}|\lambda} \equiv \mathrm{Prob}\left(\mathsf{Y}|\lambda = \vec{y}\right) = \mathrm{Prob}\left[\mathcal{P}(\lambda, \vec{x}) = \vec{y}\right]$, donde $\vec{y} = (y_1, \cdots, y_{N_b}) \in \mathbf{Y}$

representa la respuesta en forma de vector digital proporcionada por la instancia a un determinado reto. Esta cantidad se puede poner en relación con la densidad de probabilidad para una función física a través de la aplicación ADC:

$$p_{\vec{y}|\lambda} = \int_{\mathrm{ADC}^{-1}(\vec{y})} p_{\psi|\lambda}\, d\psi \tag{2.80}$$

donde la integral se realiza en el subespacio $\mathrm{ADC}^{-1}(\vec{y}) \subset \mathbf{\Psi}$ de respuestas físicas que codifican para una misma respuesta binaria $\vec{y}$. Finalmente, podemos escribir la probabilidad marginal de obtener una respuesta $\vec{y}$:

$$p_{\vec{y}} = \int_{\mathbf{\Lambda}} p_{\vec{y}|\lambda}\, p_\lambda\, d\lambda \tag{2.81}$$

Evaluar una función no-clonable físicamente es determinar un conjunto de respuestas de $\mathcal{P}$ tales que proporcionen información significativa sobre el comportamiento de la función [101], [102]. La formalización dada en (2.79) ofrece una manera clara de definir el conjunto de respuestas propio de una evaluación PUF. Dada la variable aleatoria $\mathcal{P}(\lambda, \vec{x})$, podemos definir una sucesión de N^{rep} medidas como un proceso estocástico:

$$\vec{\mathcal{P}}(\lambda, \vec{x}) \equiv [\mathcal{P}(\lambda, \vec{x})_1, \cdots, \mathcal{P}(\lambda, \vec{x})_{N^{\mathrm{rep}}}] \tag{2.82}$$

de modo que para la instancia λ_i, el reto $\vec{x}_j$ y la k-ésima repetición definimos la ijk-ésima respuesta de la PUF $\mathcal{P}$,

$$\vec{y}^{\mathcal{P}}_{ijk} \equiv \mathcal{P}(\lambda_i, \vec{x}_j)_k \tag{2.83}$$

y dado un conjunto de N^{inst} instancias $\{\lambda_i\}_{i=1}^{N^{\mathrm{inst}}} \subset \mathbf{\Lambda}$ y N^{retos} retos $\{\vec{x}_j\}_{j=1}^{N^{\mathrm{retos}}} \subset \mathbf{X}$, se define el conjunto de evaluación (o *experimento*) sobre la PUF $\mathcal{P}$, $\mathbf{Y}^{\mathcal{P}} \subset \mathbf{Y}$ como:

$$\mathbf{Y}^{\mathcal{P}}\left(N^{\mathrm{inst}}, N^{\mathrm{retos}}, N^{\mathrm{rep}}\right) \equiv \left\{\vec{y}^{\mathcal{P}}_{ijk}\right\}_{i=1,\ j=1,\ k=1}^{N^{\mathrm{inst}}, N^{\mathrm{retos}}, N^{\mathrm{rep}}} \tag{2.84}$$

2.3.1. Unicidad

Se dice que una PUF presenta una alta unicidad cuando, dada una cierta métrica $\mathrm{dist}(\vec{y_1}, \vec{y_2})$, $\vec{y_1}, \vec{y_2} \in \mathbf{Y}$ definida sobre el espacio de respuestas PUF, la distancia típica entre cualesquiera pares de respuestas generadas por instancias diferentes para a un mismo reto es grande, donde "grande" significa "de magnitud suficiente" como para

resolver entre instancias diferentes, y depende de la aplicación en que se empeñe el sistema PUF. En el contexto de la notación formal introducida en esta sección, esto se puede expresar como:

$$\mathrm{Prob}\left(\mathrm{dist}\left[\mathcal{P}\left(\lambda,\vec{x}\right),\mathcal{P}\left(\lambda',\vec{x}\right)\right]\gg 0\right)\approx 1\;\forall\,\vec{x},\lambda\neq\lambda' \tag{2.85}$$

La distancia de Hamming definida en (2.76) resulta una métrica conveniente porque la distribución de distancias de Hamming entre pares de vectores binarios aleatorios resulta sencilla de modelar: sea un par de variable aleatorias vectoriales de dimensión N_b, $\vec{\mathsf{A}}=(\mathsf{A}_1,\cdots,\mathsf{A}_{N_b})$, $\vec{\mathsf{B}}=(\mathsf{B}_1,\cdots,\mathsf{B}_{N_b})$ cuyos elementos A_i, B_i se distribuyen de manera independiente en $\mathbb{B}$, $\mathrm{Prob}(\mathsf{A}_i|\mathsf{A}_j)=\mathrm{Prob}(\mathsf{A}_i)$, $\forall i,j$ (ídem para cada B_i) y $\mathrm{Prob}(\mathsf{A}_i|\mathsf{B}_i)=\mathrm{Prob}(\mathsf{A}_i)\;\forall i$; denotaremos la probabilidad de que el i-ésimo bit de cada variable sea la unidad como $q_{a_i}\equiv\mathrm{Prob}(\mathsf{A}_i=1)$, $q_{b_i}\equiv\mathrm{Prob}(\mathsf{B}_i=1)$ y, en consecuencia, $\mathrm{Prob}(\mathsf{A}_i=0)=1-q_{a_i}$, $\mathrm{Prob}(\mathsf{B}_i=0)=1-q_{b_i}$. Si llamamos $q_{a_i}\equiv\mathrm{Prob}(\mathsf{A}_i=1)$, $q_{b_i}\equiv\mathrm{Prob}(\mathsf{B}_i=1)$ a las probabilidades de que el i-ésimo bit de cada vector valga 1 (*i.e.*, el sesgo de la distribución de bits), podemos utilizar la definición de la distancia de Hamming entre los vectores $\vec{a}$, $\vec{b}$ dada en (2.76), $\mathrm{HD}\left(\vec{a},\vec{b}\right)\equiv\sum_i a_i\oplus b_i$, para calcular la probabilidad p_i de que la i-ésima posición contribuya a la distancia de Hamming total con una unidad:

$$\begin{aligned}p_i\equiv\mathrm{Prob}\left(\mathsf{A}_i\oplus\mathsf{B}_i=1\right)&=\mathrm{Prob}\left(\mathsf{A}_i=1,\mathsf{B}_i=0\right)+\mathrm{Prob}\left(\mathsf{A}_i=0,\mathsf{B}_i=1\right)\\&=q_{a_i}\left(1-q_{b_i}\right)+\left(1-q_{a_i}\right)q_{b_i}\end{aligned} \tag{2.86}$$

donde se ha utilizado la independencia de las variables A_i, B_i entre sí para escribir la probabilidad del evento conjunto como un producto de probabilidades. Si además los bits de cada vector son idénticamente distribuidos, *i.e.*, $q_{a_i}=q_a$, $q_{b_i}=q_b$, $p_i=p$, la probabilidad de extraer un par de vectores concretos $\vec{a}$, $\vec{b}$ tales que su distancia de Hamming sea igual a m será:

$$\mathrm{Prob}\left[\mathrm{HD}(\vec{\mathsf{A}}=\vec{a},\vec{\mathsf{B}}=\vec{b})=m\right]=p^m\left(1-p\right)^{N_b-m} \tag{2.87}$$

Y dado que el número de vectores $(\vec{a}+\vec{b})$ de N_b bits tal que sus elementos suman k es el número combinatorio N_b sobre k, la probabilidad de extraer un par de vectores cualesquiera tales que su distancia de Hamming es igual a m será:

$$\begin{aligned}\mathrm{Prob}\left[\mathrm{HD}\left(\vec{A},\vec{B}\right)=m\right]&=\binom{N_b}{m}p^m(1-p)^{N_b-m}\\&=\underset{N_b,p}{\mathrm{Bin}}(m)\end{aligned} \tag{2.88}$$

que es la distribución binomial de parámetros N_b número de ensayos y sesgo $p = q_a\,(1 - q_b) + (1 - q_a)\,q_b$.

En virtud de esta métrica, se define la inter-distancia de Hamming, $\mathrm{HD}^{\mathrm{inter}}$, como la distancia de Hamming entre dos respuestas para dos instancias diferentes (i, i'), un mismo reto (j) y una misma repetición (k):

$$\mathrm{HD}^{\mathrm{inter}}_{ii'jk} \equiv \mathrm{HD}\left(\vec{y}^{\mathcal{P}}_{ijk},\, \vec{y}^{\mathcal{P}}_{i'jk}\right) \tag{2.89}$$

y la distribución de las inter-distancias sobre el experimento $\mathbf{Y}^{\mathcal{P}}$:

$$\mathbf{D}^{\mathrm{inter}}\left(\mathbf{Y}^{\mathcal{P}}\right) = \left\{\mathrm{HD}^{\mathrm{inter}}_{i,i',j,k}\right\}_{i=1,\;\; i'=i+1,\, j=1,\;\; k=1}^{N^{\mathrm{inst}},\, N^{\mathrm{inst}},\;\;\; N^{\mathrm{retos}},\, N^{\mathrm{rep}}} \tag{2.90}$$

Para una PUF ideal, el conocimiento del i-ésimo bit producido por una instancia debería aportar la menor cantidad posible de información respecto del resultado de la medida para cualesquiera otras instancias, *i.e.*, la incertidumbre (o aleatoriedad) de la distancia de Hamming para el i-ésimo bit será máxima (notar que para ambos casos extremos de una distancia $\mathrm{HD} = 0$, $\mathrm{HD} = 1$, uno de los bit está perfectamente determinado por el otro). La cantidad que mide la magnitud de esta aleatoriedad es la entropía de Shannon (sección 2.1.2). Utilizando la distribución de probabilidad q_i dada en (2.86) para la contribución a la distancia de Hamming del i-ésimo bit, escribimos su entropía h_i como:

$$h_i = -p_i \log_2 p_i - (1 - p_i) \log_2 (1 - p_i) \tag{2.91}$$

De forma que el valor p_i que maximiza esta función:

$$\begin{aligned} \frac{dh_i}{dp_i} &= -\log_2 p_i + \log_2 (1 - p_i) = 0 \\ &\Rightarrow p_i = \frac{1}{2} \end{aligned} \tag{2.92}$$

Este resultado permite modelar la inter-distancia de una PUF ideal cuyas respuestas constan de N_b bits como una distribución binomial (2.88) con parámetros $n = N_b$, $p = 1/2$,

$$\mathrm{Prob}\left(\mathrm{HD}^{\mathrm{inter}} = k\right) = \underset{N_b,1/2}{\mathrm{Bin}}(k) \tag{2.93}$$

cuya forma característica se ha representado en la figura 2.4. De esta manera, podemos caracterizar la unicidad de una propuesta PUF mediante el grado en el que se aproxima la distribución de inter-distancias de un experimento $\mathbf{Y}^{\mathcal{P}}$ a la distribución binomial de parámetros $n = N_b$, $p = 1/2$. La práctica habitual para

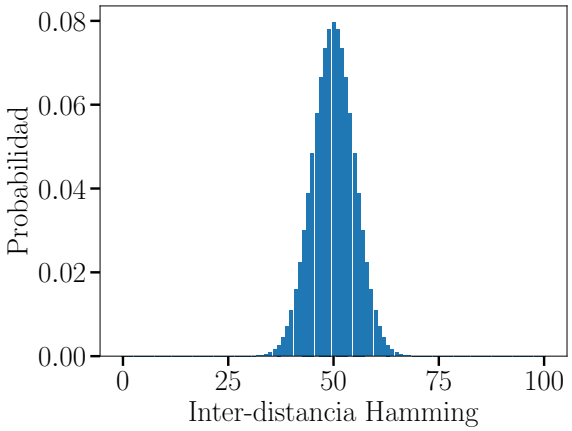


Fig. 2.4.: Distribución de probabilidad de las inter-distancias de Hamming para una PUF ideal.

proporcionar esta medida de unicidad es comparar el valor promedio de las inter-distancias distribuidas idealmente, $\mu^{\text{inter}} = N/2$, con la estimación experimental $\tilde{\mu}^{\text{inter}}$ definida como:

$$\tilde{\mu}^{\text{inter}} = \overline{\mathbf{D}^{\text{inter}}}\left(\mathbf{Y}^{\mathcal{P}}\right) = \frac{\sum_{i}^{N^{\text{inst}}} \sum_{i'=i+1}^{N^{\text{inst}}} \sum_{j}^{N^{\text{retos}}} \sum_{k}^{N^{\text{rep}}} \text{HD}\left(\vec{y}_{ijk}^{\mathcal{P}}, \vec{y}_{i'jk}^{\mathcal{P}}\right)}{N^{\text{inst}} N^{\text{retos}} N^{\text{rep}} (N^{\text{inst}} - 1)/2} \tag{2.94}$$

2.3.2. Reproducibilidad

Se dice que una PUF es reproducible (o "fiable") cuando una instancia expuesta a un estímulo $\vec{x}$ en diferentes instantes y, presumiblemente, diferentes condiciones ambientales, es capaz de devolver una misma respuesta. Dado que la naturaleza estocástica de una función no-clonable físicamente impide el cumplimiento estricto de esta condición, debe interpretarse como que las respuestas generadas por una misma instancia son lo suficientemente cercanas, dada una cierta métrica, como para permitir asimilarlas. Esta condición se puede expresar como:

$$\text{Prob}(\text{dist}[\mathcal{P}(\lambda, \vec{x})_k, \mathcal{P}(\lambda, \vec{x})_{k'}] \approx 0) \approx 1 \; \forall \vec{x}, \lambda, k, k' \tag{2.95}$$

Dado un experimento PUF $\mathbf{Y}^{\mathcal{P}} = \{\vec{y}_{ijk}^{\mathcal{P}}\}$ tal y como se ha descrito en (2.84), se define la intra-distancia de Hamming (HD^{intra}) como la distancia de Hamming

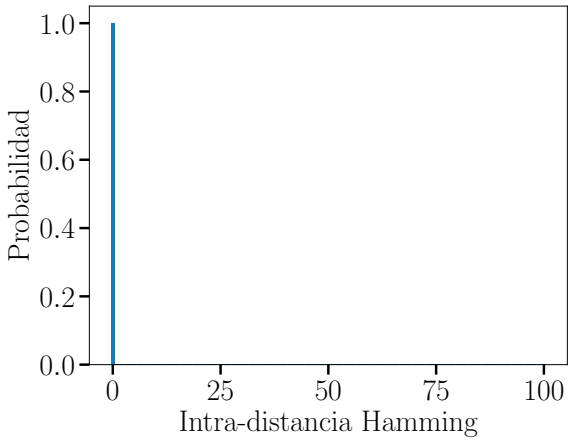


Fig. 2.5.: Distribución de probabilidad de las intra-distancias de Hamming para una PUF ideal.

entre dos respuestas para dos repeticiones (k, k') diferentes, un mismo reto (j) y una misma instancia (i):

$$\mathrm{HD}^{\text{intra}}_{ijkk'} \equiv \mathrm{HD}\left(\vec{y}^{\mathcal{P}}_{ijk},\, \vec{y}^{\mathcal{P}}_{ijk'}\right) \tag{2.96}$$

y la distribución sobre el experimento $\mathbf{Y}^{\mathcal{P}}$:

$$\mathbf{D}^{\text{intra}}\left(\mathbf{Y}^{\mathcal{P}}\right) \equiv \left\{\mathrm{HD}^{\text{intra}}_{ijkk'}\right\}_{i=1,\ j=1,\ k=1,\ k'=k+1}^{N^{\text{inst}},\, N^{\text{retos}},\, N^{\text{rep}},\, N^{\text{rep}}} \tag{2.97}$$

En una PUF ideal, las respuestas de una instancia para un determinado reto son perfectamente reproducibles (2.95), lo cual se puede modelar fácilmente de acuerdo con (2.88) como:

$$\mathrm{Prob}\left(\mathrm{HD}^{\text{intra}} = k\right) = \underset{n,0}{\mathrm{Bin}}(k) = \delta_{k,0} \tag{2.98}$$

donde "δ" representa la delta de Kronecker, *i.e.*, la distribución nula para todo $k \neq 0$, tal como se muestra en la figura 2.5. De forma análoga al caso de la unicidad, la reproducibilidad de una PUF se puede caracterizar mediante la bondad con la que se aproxima la distribución de los valores de un experimento $\mathbf{Y}^{\mathcal{P}}$ a la distribución binomial, y en particular la desviación del parámetro $\tilde{p}$ experimental con respecto del parámetro $p = 0$ ideal. Típicamente se compara el promedio de la distribución ideal, $\mu^{\text{intra}} = 0$, con el valor promedio de intra-distancias halladas experimentalmente $\tilde{\mu}^{\text{intra}}$, el cual se define como:

$$\tilde{\mu}^{\text{intra}} = \overline{\mathbf{D}^{\text{intra}}}\left(\mathbf{Y}^{\mathcal{P}}\right) = \frac{\sum_{i}^{N^{\text{inst}}} \sum_{j}^{N^{\text{retos}}} \sum_{k}^{N^{\text{rep}}} \sum_{k'=k+1}^{N^{\text{rep}}} \mathrm{HD}\left(\vec{y}^{\mathcal{P}}_{ijk},\, \vec{y}^{\mathcal{P}}_{ijk'}\right)}{N^{\text{inst}} N^{\text{retos}} N^{\text{rep}} (N^{\text{rep}} - 1)/2} \tag{2.99}$$

2.3.3. No-clonabilidad física e identificabilidad

Las propiedades de no-clonabilidad física e identificabilidad, junto con la capacidad de integrar una interfaz digital tal y como se representa en (2.79), son las características necesarias y suficientes (*i.e.*, definitorias) de un sistema PUF [29].

No-clonabilidad

Estrictamente, la no-clonabilidad es una propiedad inherente a cualquier construcción física, en el sentido de que cualquier proceso de fabricación comprenderá un cierto margen de error para cualesquiera parámetros físicos controlados explícita o implícitamente durante el proceso y, por lo tanto, para cualquier magnitud física existirá una escala de variaciones lo suficientemente pequeñas (*i.e.*, el error característico asociado del proceso de fabricación) en la cual las diferencias entre dos objetos físicos diseñados para ser idénticos resulten evidentes. Lo que distingue un dispositivo PUF de la variabilidad presentada por otros dispositivos es la exigencia de que estas discrepancias en el margen de error del proceso de fabricación sean medibles, permitiendo la comparación entre diversos dispositivos; típicamente, esta condición de mensurabilidad suele endurecerse exigiendo que el proceso de evaluación sea además fácil de llevar a cabo. Este concepto de "facilidad" es relativo a cada aplicación específica, a los recursos disponibles y al nivel de seguridad buscado, *e.g.*, existen propuestas PUF que requieren de un montaje experimental tal como la aplicación de láseres y un sistema de procesamiento de imagen -PUF óptica- que resultaría impracticable para una PUF electrónica embebida en un dispositivo IoT.

Un subconjunto de particular interés dentro de los dispositivos que cumplen estas propiedades son los sistemas electrónicos, dado que permiten su integración en ecosistemas mayores a los cuales aportan una raíz de la confianza[13] sobre la cual apoyar todas o algunas propiedades de seguridad informática. En particular, la autenticación, almacenamiento seguro de claves y generación de claves, como se discutirá más adelante en la sección 2.3.7.

[13] En un sistema informático que ejecuta aplicaciones criptográficas, la raíz de la confianza (*Root of Trust*, RoT) es la fuente de entropía primaria la cual es inaccesible física o lógicamente fuera del sistema informático que la explota; donde "inaccesible" significa que carece de interfaces externas, y no debe confundirse con "invulnerable".

En base a la formalización (2.79), la no-clonabilidad física se puede expresar como:

$$\mathcal{P}(\lambda, \vec{x}) = \mathcal{P}(\lambda', \vec{x}) \Longrightarrow \lambda = \lambda' \tag{2.100}$$

que incide en el hecho de que dos instancias PUF sólo pueden ser iguales si son efectivamente el mismo dispositivo. Sin embargo, esta propiedad no exige que el dispositivo no pueda ser clonado matemáticamente, *i.e.*, construido un modelo matemático que simule la funcionalidad de una PUF y ofrezca la respuesta correcta a cada reto. Esto es trivial en el caso de PUF débiles (sección 2.3.6), donde un adversario que acceda al conjunto de pares CRP (probablemente sólo uno) puede realizar un clon matemático sencillamente tabulando cada desafío con su correspondiente respuesta, comprometiendo así la instancia PUF. Diremos que una función Γ_λ,

$$\begin{aligned} \Gamma_\lambda :\ & \mathbf{X} \to \mathbf{Y} \\ & \vec{x} \mapsto \Gamma_\lambda(\boldsymbol{x}) = \vec{y} \end{aligned} \tag{2.101}$$

es un clon matemático de la λ-ésima instancia de una PUF $\mathcal{P}$ si cumple:

$$\mathrm{Prob}[\Gamma_\lambda(\vec{x}) = \vec{y}] = \mathrm{Prob}[\mathcal{P}(\lambda, \vec{x}) = \vec{y}]\ \forall\, \vec{x} \in \mathbf{X} \tag{2.102}$$

Recíprocamente, la instancia $\mathcal{P}(\lambda = \lambda_0, \vec{x})$ es no-clonable matemáticamente si no existe ninguna función Γ_{λ_0} que cumpla (2.102). Si $\mathcal{P}(\lambda = \lambda_0, \vec{x})$ es no-clonable matemáticamente $\forall \lambda_0$, diremos que la PUF $\mathcal{P}$ es verdaderamente no-clonable. La propiedad de verdadera no-clonabilidad es una característica deseable, sin embargo innecesaria, de una propuesta PUF. Por el contrario, la no-clonabilidad física dada en (2.100) así como la propiedad de identificabilidad que se introduce a continuación son las propiedades definitorias de una función no-clonable físicamente.

Identificabilidad

Dada la naturaleza difusa (*i.e.*, no perfectamente reproducible en el tiempo) de las funciones no-clonables físicamente, evaluar la medida en la que una PUF se desvía de las características ideales (2.3.1 y 2.3.2) resulta fundamental para delimitar su ámbito de aplicación. En un proceso de identificación basado en funciones no-clonables físicamente pueden reconocerse dos clases de errores: “falso rechazo”, cuando una instancia legítima que trata de autenticarse es calificada equivocadamente como fraudulenta, y de forma complementaria la “falsa aceptación”, en el que una instancia fraudulenta es reconocida como legítima por la instancia verificadora. La decisión de si dos evaluaciones PUF proceden de hecho de una misma instancia o

de instancias diferentes se articula a través de la noción de distancia que se haya definido entre respuestas, en particular de la distancia Hamming en el contexto de este trabajo. Intuitivamente, resulta claro que una PUF puede calificarse de "más" identificable cuanto más improbable es cometer un error de falso rechazo o de falsa aceptación. Dado que el rechazo/aceptación es una decisión tomada en virtud de una medida de distancia (de Hamming), podemos estimar la probabilidad de error a partir de las distribuciones de intra/inter-distancia. Sea un conjunto de N^{rep} respuestas PUF obtenidas en un experimento para una instancia λ_i y un reto $\vec{x}_j$, $\{\vec{y}^{\mathcal{P}}_{ijk}\}_{k=1}^{N^{\text{rep}}}$, definimos la función Enroll como la aplicación que extrae de este conjunto una respuesta "dorada" de referencia, que servirá como identificador de la PUF:

$$\vec{y}^{\mathcal{P}}_{ij,\text{ref}} \equiv \text{Enroll}\left(\{\vec{y}^{\mathcal{P}}_{ijk}\}_{k=1}^{N^{\text{rep}}}\right) \tag{2.103}$$

Formalmente, se define el "rechazo" como el evento de que la distancia entre esta respuesta de referencia y una evaluación circunstancial, de la misma u otra instancia, $\vec{y}^{\mathcal{P}}_{i'jk'}$, sea mayor que un cierto umbral de identificación, u^{id}. Cuando se requiera la autenticación del dispositivo, el agente verificador expondrá la instancia solicitante $\lambda_{i'}$ a un reto $\vec{x}_j$, y comparará la respuesta $\vec{y}^{\mathcal{P}}_{i'jk'}$ con la clave dorada $\vec{y}^{\mathcal{P}}_{ij,\text{ref}}$, la cual habrá sido recopilada por el verificador con antelación, probablemente en una fase del ciclo vital de la PUF previa a su despliegue en campo (sección 2.3.7). Si la distancia entre ambas es menor que el umbral de identificación, $\text{HD}(\vec{y}^{\mathcal{P}}_{i'jk'}, \vec{y}^{\mathcal{P}}_{ij,\text{ref}}) \leq u^{\text{id}}$, entonces la identificación se considera positiva, asumiendo $\lambda_{i'} = \lambda_i$. En caso contrario, el intento de autenticación será negativo, $\lambda_{i'} \neq \lambda_i$. Los posibles eventos que pueden tener lugar durante el proceso de autenticación son (se omiten los índices correspondientes al reto y las repeticiones j, k y k' de cada respuesta para aliviar la notación):

Evento	**Descripción**	**Prob. acumulada**
Verdadera aceptación (VA)	$\text{HD}\left(\vec{y}^{\mathcal{P}}_{i'}, \vec{y}^{\mathcal{P}}_{i,\text{ref}}\right) \leq u^{\text{id}}$, $\lambda_{i'} = \lambda_i$	$\text{Prob}\left(\text{HD}^{\text{intra}} \leq u^{\text{id}}\right)$
Verdadero rechazo (VR)	$\text{HD}\left(\vec{y}^{\mathcal{P}}_{i'}, \vec{y}^{\mathcal{P}}_{i,\text{ref}}\right) > u^{\text{id}}$, $\lambda_{i'} \neq \lambda_i$	$\text{Prob}\left(\text{HD}^{\text{inter}} > u^{\text{id}}\right)$
Falsa aceptación (FA)	$\text{HD}\left(\vec{y}^{\mathcal{P}}_{i'}, \vec{y}^{\mathcal{P}}_{i,\text{ref}}\right) \leq u^{\text{id}}$, $\lambda_{i'} \neq \lambda_i$	$\text{Prob}\left(\text{HD}^{\text{inter}} \leq u^{\text{id}}\right)$
Falso rechazo (FR)	$\text{HD}\left(\vec{y}^{\mathcal{P}}_{i'}, \vec{y}^{\mathcal{P}}_{i,\text{ref}}\right) > u^{\text{id}}$, $\lambda_{i'} = \lambda_i$	$\text{Prob}\left(\text{HD}^{\text{intra}} > u^{\text{id}}\right)$

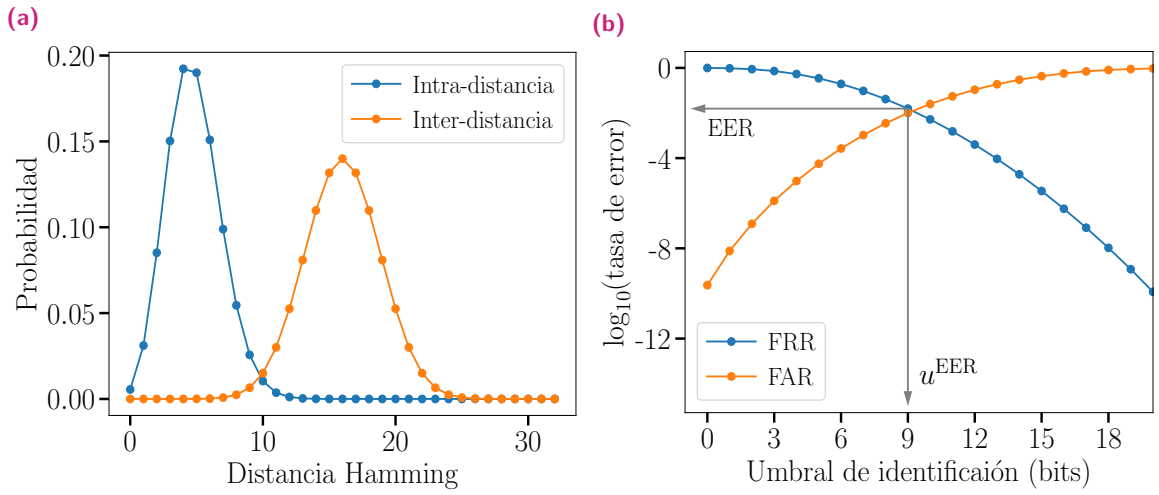


Fig. 2.6.: Ejemplo de análisis de identificabilidad: (a) ajustes binomiales a las distribuciones de intra/inter-distancias de Hamming; y (b) ejemplos de curvas FAR/FRR. En ambos casos se ha destacado la tasa EER.

A las probabilidades $\mathrm{Prob(FA)}$, $\mathrm{Prob(FR)}$ se les denomina respectivamente "tasa de falsa aceptación" (*False Acceptance Rate*, FAR) y "tasa de falso rechazo" (*False Rejection Rate*, FRR). Estas cantidades son funciones del umbral de identificación u^{id} y se definen:

$$\mathrm{FAR}(u^{\mathrm{id}}) = \sum_{k=0}^{u^{\mathrm{id}}} \mathrm{Prob}\Big(\mathrm{HD}^{\mathrm{inter}} = k\Big) = F_{\mathrm{HD}^{\mathrm{inter}}}(u^{\mathrm{id}}) \tag{2.104}$$

$$\mathrm{FRR}(u^{\mathrm{id}}) = \sum_{k=u^{\mathrm{id}}}^{N} \mathrm{Prob}\Big(\mathrm{HD}^{\mathrm{intra}} = k\Big) = 1 - F_{\mathrm{HD}^{\mathrm{intra}}}(u^{\mathrm{id}}) \tag{2.105}$$

donde $F_{\mathsf{K}}(x)$ es la acumulación de la distribución de probabilidad de la variable aleatoria K hasta el valor x, $F_{\mathsf{K}}(x) \equiv \sum_{k=0}^{x} \mathrm{Prob}(\mathsf{K} = k)$. En general, la elección del umbral de identificación depende de la aplicación concreta en que se esté empleando la PUF, de forma que pueden tomarse valores más conservadores (*i.e.*, umbrales menores) en aplicaciones donde resulte crítico evitar la falsa aceptación, aun a costa de cometer un mayor número de errores de falso rechazo. Recíprocamente, el umbral puede incrementarse en caso de que la denegación de un servicio a una instancia legítima resulta más lesivo que la suplantación por parte de una instancia ilegítima. En la figura 2.6a se muestran los ajustes típicos a la intra/inter-distancias de Hamming para un experimento PUF. En la figura 2.6b se muestran las curvas FAR y FRR tal y como se definen en (2.104) y (2.105) para las distribuciones de la figura 2.6a. El punto en el que se cortan ambas curvas es el umbral de identificación

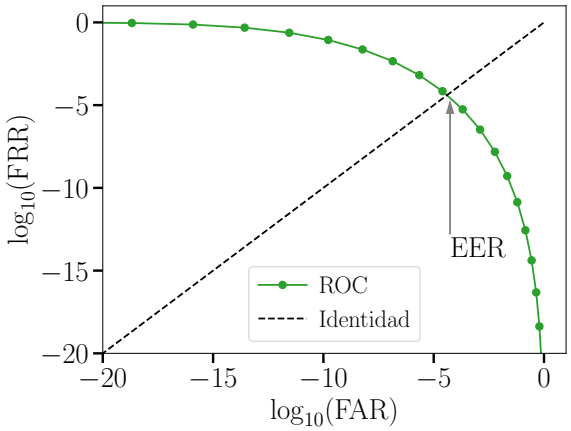


Fig. 2.7.: Ejemplo de curva ROC para una función no-clonable físicamente, donde se ha destacado la tasa EER.

correspondiente a la "tasa de igual error" (*Equal Error Rate*, EER), en el que las tasas de falso rechazo y falsa aceptación coinciden; denotaremos este umbral como u^{EER}.

Finalmente, el comportamiento completo de una PUF respecto de su identificabilidad se puede caracterizar mediante la curva característica operativa del receptor (*Receiver-Operating Characteristic*, ROC). Esta mide la tasa de falso rechazo en función de la tasa de falsa aceptación, $\mathrm{FRR}\,(\mathrm{FAR})$, y es una figura de mérito adecuada para comparar el rendimiento como sistema de identificación de PUF diferentes, incluso si su comportamiento se basa en fenómenos físicos distintos [103]. En la figura 2.7 se muestra un ejemplo de curva ROC; el punto en el que cortan la recta identidad y la curva ROC coincide con la cantidad EER.

2.3.4. Modelo cuasi-ideal de PUF

Las distribuciones de intra/inter-distancia Hamming en un experimento real serán, en general, diferentes de las curvas binomiales ideales dadas en (2.93) y (2.98). Sin embargo, si llamamos $\vec{y}_i^{\bigstar}$ a la respuesta binaria nominal de la i-ésima instancia, diremos que la PUF es cuasi-ideal si podemos construir la respuesta para una instancia i y una repetición k como $\vec{y}_{i,k} = \vec{y}_i^{\bigstar} \oplus \Delta\vec{y}_{i,k}$ con las condiciones:

1. Cada bit de la respuesta nominal está dado por una variable aleatoria binaria $\mathrm{Y}_i^{\bigstar}$ que se distribuye de manera idéntica e independiente para todos los bits de la respuesta, con probabilidad $q^{\mathrm{inst}} \equiv \mathrm{Prob}(\mathrm{Y}_i^{\bigstar} = 1)\,\forall i$.

2. Cada bit de la diferencia entre la respuesta nominal y real está dado por una variable aleatoria binaria $\Delta \mathsf{Y}_{i,k}$ que se distribuye de manera idéntica e independiente para todos los bits de la respuesta, con probabilidad $q^{\text{rep}} \equiv \text{Prob}(\Delta \mathsf{Y}_{i,k} = 1)\, \forall i, k$.

La intra-distancia de Hamming entre dos repeticiones k, k' de una misma instancia i será:

$$\begin{aligned} \text{HD}^{\text{intra}}_{i,k,k'} &= \sum y_{i,k} \oplus y_{i,k'} \\ &= \sum y_i^{\bigstar} \oplus \Delta y_{i,k} \oplus y_i^{\bigstar} \oplus \Delta y_{i,k'} \\ &= \sum \left(y_i^{\bigstar} \oplus y_i^{\bigstar} \right) \oplus \left(\Delta y_{i,k} \oplus \Delta y_{i,k'} \right) \\ &= \sum \Delta y_{i,k} \oplus \Delta y_{i,k'} \end{aligned} \tag{2.106}$$

donde se ha suprimido el índice de la suma para aliviar la notación, pero debe notarse que esta se extiende a cada bit de la respuesta binaria, y se ha utilizado la propiedad conmutativa y asociativa de la operación $\oplus$, así como $x \oplus x = 0$. Podemos utilizar el resultado (2.86) con $q_a = q_b = q^{\text{rep}}$ para escribir la probabilidad p^{intra} de que los bits $y_{i,k}$, $y_{i,k'}$ contribuyan a la distancia de Hamming:

$$p^{\text{intra}} \equiv \text{Prob}\left(\Delta \mathsf{Y}_{i,k} \oplus \Delta \mathsf{Y}_{i,k'} = 1 \right) = 2q^{\text{rep}}(1 - q^{\text{rep}}) \tag{2.107}$$

Lo cual nos permite escribir la distribución de probabilidad para la intra-distancia de Hamming como (2.88) una binomial de parámetros $n = N_b$, $p = p^{\text{intra}}$:

$$\text{Prob}\left(\text{HD}^{\text{intra}}_{i,k,k'} = m \right) = \underset{N_b,\, p^{\text{intra}}}{\text{Bin}}(m) \tag{2.108}$$

Por otra parte, para la distribución de inter-distancias se tiene:

$$\begin{aligned} HD^{\text{inter}}_{i,i',k} &= \sum y_{i,k} \oplus y_{i',k} \\ &= \sum y_i^{\bigstar} \oplus \Delta y_{i,k} \oplus y_{i'}^{\bigstar} \oplus \Delta y_{i',k} \\ &= \sum \left(y_i^{\bigstar} \oplus y_{i'}^{\bigstar} \right) \oplus \left(\Delta y_{i,k} \oplus \Delta y_{i',k} \right) \end{aligned} \tag{2.109}$$

donde ahora no se puede cancelar el primer término porque las cantidades $\vec{y}_i^{\star}$, $\vec{y}_{i'}^{\star}$ serán en general distintas entre sí. Sin embargo, podemos escribir la probabilidad de que los bits $y_{i,k}$, $y_{i',k}$ contribuyan a la distancia de Hamming:

$$\begin{aligned} p^{\text{inter}} &\equiv \text{Prob}\left[(\mathsf{Y}_i^{\star} \oplus \mathsf{Y}_{i'}^{\star}) \oplus (\Delta\mathsf{Y}_{i,k} \oplus \Delta\mathsf{Y}_{i',k}) = 1\right] \\ &= \text{Prob}(\mathsf{Y}_i^{\star} \oplus \mathsf{Y}_{i'}^{\star} = 1)\text{Prob}(\Delta\mathsf{Y}_{i,k} \oplus \Delta\mathsf{Y}_{i',k} = 0) \\ &+ \text{Prob}(\mathsf{Y}_i^{\star} \oplus \mathsf{Y}_{i'}^{\star} = 0)\text{Prob}(\Delta\mathsf{Y}_{i,k} \oplus \Delta\mathsf{Y}_{i',k} = 1) \\ &= \text{Prob}(\mathsf{Y}_i^{\star} \oplus \mathsf{Y}_{i'}^{\star} = 1)\left(1 - \text{Prob}(\Delta\mathsf{Y}_{i,k} \oplus \Delta\mathsf{Y}_{i',k} = 1)\right) \\ &+ \left[1 - \text{Prob}(\mathsf{Y}_i^{\star} \oplus \mathsf{Y}_{i'}^{\star} = 1)\right]\text{Prob}(\Delta\mathsf{Y}_{i,k} \oplus \Delta\mathsf{Y}_{i',k} = 1) \end{aligned} \tag{2.110}$$

Por (2.107) se tiene $\text{Prob}(\Delta\mathsf{Y}_{i,k} \oplus \Delta\mathsf{Y}_{i',k} = 1) = p^{\text{intra}}$. Así mismo, en el modelo cuasi-ideal, la variable $\mathsf{Y}_i^{\star}$ se distribuye uniformemente con probabilidad $q^{\text{inst}} = \text{Prob}(\mathsf{Y}_i^{\star} = 1)\,\forall i$ de que un bit cualquiera valga la unidad, de modo que de acuerdo con (2.86), cada bit de la variable $(\mathsf{Y}_i^{\star} \oplus \mathsf{Y}_{i'}^{\star})$ se distribuirá uniformemente con probabilidad:

$$\text{Prob}(\mathsf{Y}_i^{\star} \oplus \mathsf{Y}_{i'}^{\star} = 1) = 2q^{\text{inst}}(1 - q^{\text{inst}}) \tag{2.111}$$

Sustituyendo esta expresión en (2.110) se tiene:

$$p^{\text{inter}} = \left[2q^{\text{inst}}(1 - q^{\text{inst}})\right]\left(1 - 2p^{\text{intra}}\right) + p^{\text{intra}} \tag{2.112}$$

De forma que utilizando (2.87) podemos escribir la probabilidad de que la inter-distancia de Hamming entre los vectores $y_{i,k}$, $y_{i',k}$ (2.109) valga m:

$$\text{HD}_{i,i',k}^{\text{inter}} = (p^{\text{inter}})^m\,(1 - p^{\text{inter}})^{N_b - m} \tag{2.113}$$

y por (2.88) se tiene que la inter-distancia de Hamming se distribuirá como una binomial de parámetros $n = N_b$, $p = p^{\text{inter}}$:

$$\text{Prob}\left(\text{HD}^{\text{inter}} = m\right) = \underset{N_b, p^{\text{inter}}}{\text{Bin}}(m) \tag{2.114}$$

En la figura 2.8 se muestra la simulación de un experimento PUF cuasi-ideal de parámetros $N^{\text{inst}} = 40$, $N^{\text{rep}} = 100$, $N_b = 100$, $p^{\text{intra}} = 0{,}02$ y $p^{\text{inter}} = 0{,}45$, elegidos arbitrariamente por su verosimilitud con respecto de los valores típicos encontrados en un experimento PUF real. Las respuestas simuladas para la i-ésima instancia y k-ésima repetición se han construido mediante la prescripción del modelo cuasi-ideal $\vec{y}_{i,k} = \vec{y}_i^{\star} \oplus \Delta y_{i,k}$, generando los vectores $\vec{y}_i^{\star}$ y $\Delta\vec{y}_{i,k}$ utilizando un generador de 100 bits aleatorios con sesgos respectivos $q^{\text{inst}} = 0{,}34$ y $q^{\text{rep}} = 0{,}010$; estos sesgos han sido calculados resolviendo las ecuaciones (2.107) y (2.112) para las cantidades

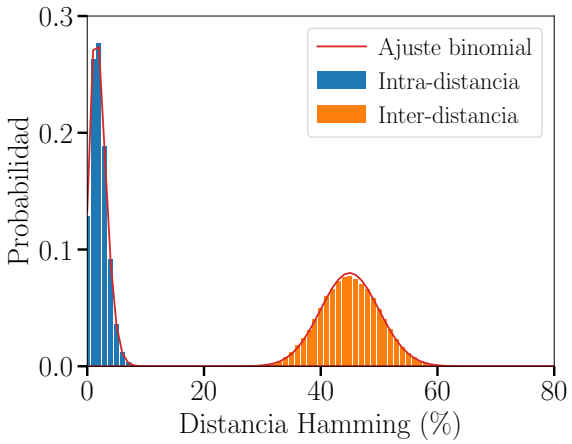


Fig. 2.8.: Histogramas de las intra/inter-distancias de Hamming simuladas con un modelo PUF cuasi-ideal de parámetros $N_b = 100$, $p^{\text{intra}} = 0{,}02$, $p^{\text{inter}} = 0{,}45$, junto con las curvas binomiales esperadas.

p^{intra} y p^{inter} utilizadas. Junto con los histogramas obtenidos por simulación para las inter/intra-distancias se han representado las curvas binomiales dadas por (2.114) y (2.108), con parámetros $n = N_b$ y $p = p^{\text{intra}}$, $p = p^{\text{inter}}$ respectivamente. Este modelo servirá como hipótesis nula para estimar la bondad del ajuste binomial de las distribuciones de inter/intra-distancia medidas experimentalmente en los capítulos 4 y 5, de forma que un resultado positivo justificará el uso de una aproximación binomial para el cálculo de la identificabilidad de una PUF, tal y como fue presentado en 2.3.3.

2.3.5. Resistencia ambiental y compensación de la medida

A través de la variable aleatoria $\mathcal{P}(\lambda, \vec{x})$ definida sobre el espacio de retos $\mathbf{X}$, las respuestas de una función no-clonable físicamente están influidas por ruido ambiental de alcance local debido a las fluctuaciones aleatorias de temperatura y tensión en cada punto del circuito PUF. Para mitigar este fenómeno, Gassend *et al.* propusieron en 2001 [24] construir la respuesta de una PUF basada en osciladores de anillo a partir de la comparación entre las frecuencias medidas por pares de osciladores diseñados de forma idéntica y físicamente próximos en el sustrato de silicio. De este modo, las desviaciones se compensarían, produciendo una respuesta más estable. Esta técnica, denominada "medida compensada", será de aplicación para cualquier función física construida mediante la repetición de una misma estructura idéntica por diseño ("celdas", *e.g.*, una matriz de osciladores de anillo). Además, proporciona una forma natural de digitalizar la respuesta analógica a partir de la

comparación entre las mediciones de cada celda, de forma que el bit resultante de comparar un par de celdas (i, j) cuyas respuestas físicas sean respectivamente ψ_i, ψ_j se construye como:

$$\mathrm{bit}(i,j) = \begin{cases} 1 & \text{si } \psi_i - \psi_j > 0 \\ 0 & \text{en otro caso.} \end{cases} \tag{2.115}$$

Este esquema de construcción de las respuestas binarias es de hecho muy frecuente, incluso en arquitecturas no diseñadas explícitamente de forma celular. Por ejemplo, una función no-clonable físicamente SRAM-PUF basada en una memoria estática de acceso aleatorio (*Static Random Access Memory*, SRAM), extrae su respuesta del valor almacenado por defecto en cada celda de memoria inmediatamente después del ciclo de encendido. Sin embargo, este valor se debe a la competencia entre las tensiones umbrales características de los transistores que componen cada celda SRAM, de modo que el bit de respuesta se debe a un esquema de medida compensada.

Dada una PUF de N celdas, la compensación de la medida permite extraer como máximo $N(N-1)/2$ bits (*i.e.*, tantos bits como parejas distintas son posibles). Sin embargo, resulta evidente que incluso para un conjunto de celdas cuyas respuestas físicas sean perfectamente independientes, todavía existirá una correlación entre bits: por ejemplo, debido a la propiedad transitiva de la comparación, dado un conjunto de tres celdas a, b y c, tales que $\psi_a > \psi_b$ y $\psi_b > \psi_c$ (donde ψ_i representa la medición de la celda "i"), se tiene $\psi_a > \psi_c$ y por lo tanto el bit obtenido de comparar las celdas a y c no será aleatorio. Esto limita la entropía máxima extraíble de este sistema a ser menor que $N(N-1)/2$ bits. Se puede establecer una cota superior a esta cantidad notando que, dado que únicamente son de interés las diferencias relativas entre las mediciones de celdas a la hora de generar la respuesta binaria (en lugar de la medida en términos absolutos), siempre podremos etiquetar cada celda del total de N elementos con un entero en el intervalo $[1, N]$, en función del orden que ocupa el valor de su respuesta física característica ψ en relación al resto de celdas de la matriz. De esta manera, cualquier realización de una matriz de N celdas se corresponderá con una permutación $\sigma\,(1, \ldots, N)$, *i.e.*, el proceso de fabricación de una N-tupla puede representarse como una variable aleatoria con distribución de probabilidad $p\,[\sigma\,(1, \ldots, N)] \equiv p_\sigma$. En el caso ideal de que todas las permutaciones sean equiprobables, $p_\sigma = \text{cte} = 1/N!$, y la entropía del proceso se define simplemente como el logaritmo cambiado de signo de esta cantidad,

$$H^{\max} = -\log_2\,(1/N!) = \log_2\,(N!) \tag{2.116}$$

Las diferentes maneras de seleccionar parejas de celdas para construir la respuesta binaria tendrán un impacto significativo en las propiedades de seguridad de la PUF resultante, lo cual será el objeto de estudio en el capítulo 3.

Probabilidad del vuelco de un bit

Podemos representar matemáticamente la respuesta de la i-ésima celda física implementada sobre un sustrato de silicio como parte de una PUF de medida compensada mediante la magnitud de su respuesta física ψ_i como [104]:

$$\psi_i = \psi_i^0 + \Delta\psi^{gD} + \Delta\psi_i^{lD} + \Delta\psi^{gA} + \Delta\psi_i^{lA} \tag{2.117}$$

donde ψ_i^0 es la respuesta física nominal obtenida al medir la i-ésima celda, y las cantidades $\Delta\psi$ representan variaciones de la respuesta debidas a:

gD: efectos globales y deterministas (*e.g.*, variaciones de temperatura o tensión de alimentación nominales que afectan a toda la superficie de silicio).

lD: efectos locales y deterministas (defectos sistemáticos en el proceso de fabricación de los osciladores). Esta cantidad puede interpretarse como un "ruido de fabricación".

gA: efectos globales y aleatorios, *e.g.*, fluctuaciones en la temperatura o tensión de alimentación del chip.

lA: efectos locales y aleatorios (ruido térmico intrínseco de los dispositivos, o ruido electromagnético extrínseco perturbando el circuito de forma local).

Las cantidades aleatorias $\Delta\psi^{gA} + \Delta\psi_i^{lA} \equiv \psi_i^A$ se pueden modelar como una variable aleatoria Ψ_i^A distribuida normalmente de media nula y desviación estándar dependiente de los parámetros ambientales que afectan al chip, en particular temperatura (T) y tensión de alimentación (V), $\sigma_i = \sigma_i(T, V, \dots)$,

$$\mathrm{Prob}\left(\Psi_i^A = \psi_i^A\right) = \underset{0,\sigma_i}{\mathrm{Norm}}(\psi_i^A) \tag{2.118}$$

donde $\mathrm{Norm}_{0,\sigma_i}$ representa la densidad de probabilidad normal de media nula y varianza σ_i^2. Esto nos permite tomar el promedio en (2.117):

$$\overline{\psi_i} \equiv \mu_i = \psi_i^0 + \psi_i^D + \cancelto{0}{\overline{\psi_i^A}} \tag{2.119}$$

con $\psi_i^D \equiv \Delta\psi^{gD} + \Delta\psi_i^{lD}$. Así, podemos modelar la respuesta de la i-ésima celda como una variable aleatoria Ψ_i distribuida normalmente de media $\mu_i = \psi_i^0 + \psi_i^D$ y varianza σ_i^2,

$$\mathrm{Prob}\left(\Psi_i = \psi_i\right) = \underset{\mu_i,\sigma_i}{\mathrm{Norm}}(\psi_i) \tag{2.120}$$

Este modelado permite representar el resultado de una medición en sucesivos instantes temporales indexando la variable aleatoria Ψ_i mediante un índice t para dar lugar a un proceso estocástico $\vec{\Psi}_i = (\Psi_i^t)_{t=0}^{\infty}$. Utilizando la función (2.115), es posible calcular la probabilidad de que se produzca el vuelco de un bit debido a fluctuaciones estadísticas en un esquema de medida compensada. El evento de que se produzca la inversión de un bit entre dos medidas sucesivas realizadas en los instantes t y $t+1$, que denotaremos "Flip", se describe como que en el instante t, $\Psi_i^t > \Psi_j^t$, y en el instante $t+1$, $\Psi_i^{t+1} < \Psi_j^{t+1}$, o bien su complementario (*i.e.*, que en el instante t, $\Psi_i^t < \Psi_j^t$, y en el instante $t+1$, $\Psi_i^{t+1} > \Psi_j^{t+1}$). Por lo tanto, podemos escribir la inversión del bit como la unión de la intersección de los eventos $(\Psi_i^t > \Psi_j^t) \cap (\Psi_i^{t+1} < \Psi_j^{t+1})$ y $(\Psi_i^t < \Psi_j^t) \cap (\Psi_i^{t+1} > \Psi_j^{t+1})$:

$$\text{Flip} \equiv \left[\left(\Psi_i^t > \Psi_j^t\right) \cap \left(\Psi_i^{t+1} < \Psi_j^{t+1}\right)\right] \cup \left[\left(\Psi_i^t < \Psi_j^t\right) \cap \left(\Psi_i^{t+1} > \Psi_j^{t+1}\right)\right] \tag{2.121}$$

Dado que los eventos a ambos lados del símbolo "unión" son disjuntos podemos escribir para la probabilidad de vuelco, $\mathrm{Prob}(\text{Flip})$:

$$\begin{aligned}\mathrm{Prob}\left(\text{Flip}\right) = \mathrm{Prob}&\left(\left[\Psi_i^t > \Psi_j^t\right] \cap \left[\Psi_i^{t+1} < \Psi_j^{t+1}\right]\right) \\ + \mathrm{Prob}&\left(\left[\Psi_i^t < \Psi_j^t\right] \cap \left[\Psi_i^{t+1} > \Psi_j^{t+1}\right]\right)\end{aligned} \tag{2.122}$$

Si las medidas sucesivas en instantes t y $t+1$ se realizan en un intervalo lo suficientemente espaciado en el tiempo como para que ambas cantidades estén descorrelacionadas, se tiene para la variable en el instante t Ψ^t, $\mathrm{Prob}(\Psi^{t+1}|\Psi^t) = \mathrm{Prob}(\Psi^{t+1})$, y se puede escribir la probabilidad de la intersección como el producto de probabilidades en (2.122):

$$\begin{aligned}\mathrm{Prob}\left(\text{Flip}\right) = \mathrm{Prob}&\left(\Psi_i^t > \Psi_j^t\right)\ \mathrm{Prob}\left(\Psi_i^{t+1} < \Psi_j^{t+1}\right) \\ + \mathrm{Prob}&\left(\Psi_i^t < \Psi_j^t\right)\ \mathrm{Prob}\left(\Psi_i^{t+1} > \Psi_j^{t+1}\right)\end{aligned} \tag{2.123}$$

Por otra parte, si las medidas en t, $t+1$ se llevan a cabo en un intervalo lo suficientemente breve como para que el dispositivo se mantenga localmente en régimen estacionario, las variables Ψ^t y Ψ^{t+1} serán idénticamente distribuidas y podemos suprimir el índice "t" en (2.123), lo cual permite escribir:

$$\begin{aligned}\text{Prob}\,(\text{Flip}) &= \text{Prob}\,(\Psi_i > \Psi_j)\ \text{Prob}\,(\Psi_i < \Psi_j) + \text{Prob}\,(\Psi_i < \Psi_j)\ \text{Prob}\,(\Psi_i > \Psi_j) \\ &= 2 \times \text{Prob}\,(\Psi_i > \Psi_j)\ \text{Prob}\,(\Psi_i < \Psi_j)\end{aligned} \tag{2.124}$$

Las correspondientes probabilidades acumuladas de cada uno de estos eventos:

$$\text{Prob}(\Psi_i > \Psi_j) = \int_{-\infty}^{\infty} d\psi_j \int_{\psi_j}^{\infty} d\psi_i\, \text{Prob}(\Psi_i = \psi_i, \Psi_j = \psi_j) \tag{2.125}$$

$$\text{Prob}(\Psi_i < \Psi_j) = 1 - \text{Prob}(\Psi_i > \Psi_j) \tag{2.126}$$

Dado el modelo (2.120) la distribución de probabilidad conjunta $\text{Prob}(\Psi_i, \Psi_j)$ es:

$$\text{Prob}(\Psi_i = \psi_i, \Psi_j = \psi_j) = \frac{1}{2\pi\sigma_i\sigma_j}\, e^{-\frac{1}{2}\frac{(\psi_i-\mu_i)^2}{\sigma_i^2}}\, e^{-\frac{1}{2}\frac{\left[\psi_j-\mu_j\right)^2}{\sigma_j^2}} \tag{2.127}$$

Utilizando esta expresión en 2.125 la integral resulta:

$$\begin{aligned}\text{Prob}(\Psi_i > \Psi_j) &= \frac{1}{2\pi\sigma_i\sigma_j} \int_{-\infty}^{\infty} d\psi_j\, e^{-\frac{1}{2}\frac{(\psi_j-\mu_j)^2}{\sigma_j^2}} \int_{\psi_j}^{\infty} d\psi_i\, e^{-\frac{1}{2}\frac{(\psi_i-\mu_i)^2}{\sigma_i^2}} \\ &= \frac{1}{2}\left[\text{erf}\left(\frac{\sqrt{2}}{2}\frac{\mu_i-\mu_j}{\sqrt{\sigma_i^2+\sigma_j^2}}\right) + 1\right]\end{aligned} \tag{2.128}$$

Esta integral es la misma que apareció en (2.57), renombrando ahora a las variables $k' \to \psi_i$, $k \to \psi_j$, $\mu_0 \to \mu_j$, $\sigma_0 \to \sigma_j$. Del mismo modo, utilizando 2.126 tenemos para la probabilidad del evento complementaria:

$$\text{Prob}(\Psi_i < \Psi_j) = \frac{1}{2}\left[1 - \text{erf}\left(\frac{\sqrt{2}}{2}\frac{\mu_i-\mu_j}{\sqrt{\sigma_i^2+\sigma_j^2}}\right)\right] \tag{2.129}$$

Insertando ahora (2.128) y (2.129) en (2.124) se tiene para la probabilidad total de vuelco:

$$\text{Prob}(\text{Flip}) = \frac{1}{2}\left[1 - \text{erf}^2\left(\frac{\sqrt{2}}{2}\frac{\mu_i-\mu_j}{\sqrt{\sigma_i^2+\sigma_j^2}}\right)\right] \tag{2.130}$$

En la figura 2.9a se han representado los resultados de simulación para la probabilidad de inversión de un bit, calculada utilizando el modelo estocástico

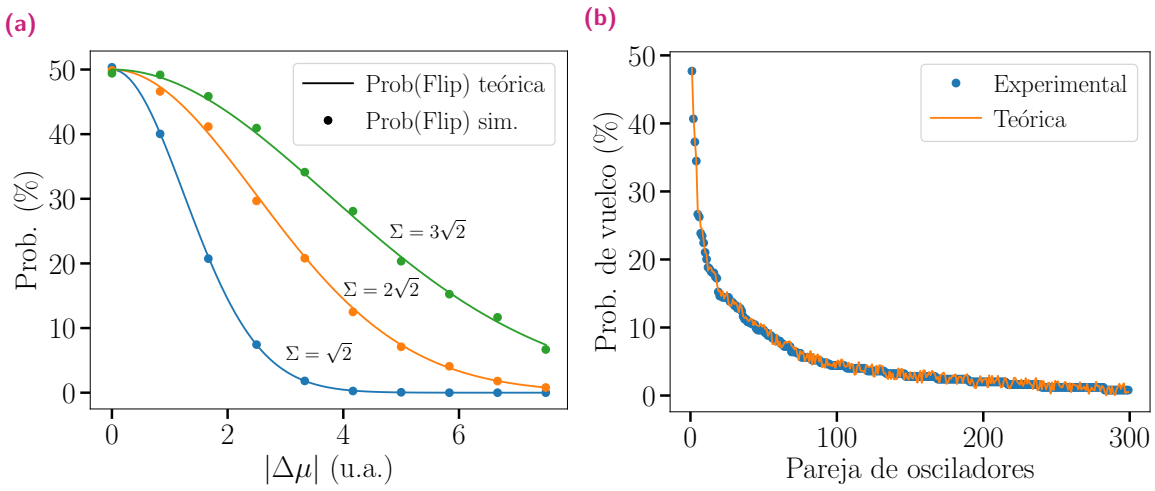


Fig. 2.9.: (a) Simulación y curva de interpolación para las probabilidades del vuelco de un bit, para tres valores Σ de desviación combinada. (b) Probabilidad teórica y experimental de la inversión de un bit para una matriz de osciladores de anillo. Las cantidades $|\Delta\mu|$ y Σ están expresadas en unidades arbitrarias.

(2.120) para una pareja de celdas (i, j) cuyas respuestas físicas ψ_i, ψ_j se distribuyen normalmente con promedios μ_i, μ_j y desviaciones σ_i, σ_j respectivamente. Estas probabilidades se muestran en función de la diferencia de valores promedio $|\Delta\mu| \equiv |\mu_i - \mu_j|$ (unidades arbitrarias), para distintas desviaciones $\Sigma \equiv \sqrt{\sigma_i^2 + \sigma_j^2}$, interpoladas por la curva dada por la expresión (2.130). Cada punto de esta simulación se ha obtenido extrayendo pares de valores de sendos generadores pseudoaleatorios distribuidos normalmente, calculando a continuación el número de inversiones sucesivas. Por otra parte, en la figura 2.9b se muestran las probabilidades de inversión de un bit medidas en una FPGA Artix 7 al comparar una serie de parejas de osciladores en una matriz de 200 anillos, superpuestas con la probabilidad teórica dada por (2.130), donde las cantidades $|\Delta\mu|$ y Σ han sido calculadas a partir de la distribución experimental de frecuencias para cada anillo de la pareja. El eje de abscisas muestra un índice arbitrario, asignado a cada pareja en orden decreciente atendiendo a la diferencia de sus frecuencias promedio: sea $|\mu|_k$ la diferencia de frecuencia promedio para la k-ésima pareja, se tiene $|\mu|_k < |\mu|_{k'} \Rightarrow k < k'$.

Corrección de errores

La variabilidad en la respuesta de una función no-clonable físicamente se puede tratar formalmente como un canal de transmisión ruidoso, *i.e*, la lectura de un

vector digital diferente de la respuesta "dorada" de referencia equivale a un canal de comunicación que introduce variaciones en algunos bits de un mensaje circulante, dando lugar a una cierta discrepancia entre el mensaje enviado y el recibido o, en este caso, entre una respuesta ideal y la respuesta medida físicamente. De este modo, se pueden aplicar a los sistemas PUF las técnicas bien establecidas de corrección de errores propias de la teoría de la comunicación, en particular los códigos de corrección de errores (*Error Correcting Code*, ECC). Esta consiste en expandir la respuesta generada con una determinada cantidad de información redundante, lo que permite recuperar la respuesta incluso si se produce un cierto número de bits erróneos. Esta información auxiliar (*Helper Data*, hd) es generada mediante una aplicación ECC, a partir de un conjunto de N^{rep} respuestas PUF medidas para una instancia λ_i y un reto $\vec{x}_j$, $\{\vec{y}^{\mathcal{P}}_{ijk}\}_{k=1}^{N^{\mathrm{rep}}}$:

$$\mathrm{hd}^{\mathcal{P}}_{ij} = \mathrm{ECC}\left(\left\{\vec{y}^{\mathcal{P}}_{ijk}\right\}_{k=1}^{N^{\mathrm{rep}}}\right) \tag{2.131}$$

Con la propiedad de que existe un codificador dependiente de la información auxiliar, $\mathrm{Encod}_{\mathrm{hd}}$, el cual transforman las respuestas PUF,

$$\vec{y'}^{\mathcal{P}}_{ijk} = \mathrm{Encod}_{\mathrm{hd}}\left(\vec{y}^{\mathcal{P}}_{ijk}\right) \tag{2.132}$$

y análogamente, un decodificador $\mathrm{Decod}_{\mathrm{hd}}$, que permite recuperar la respuesta de referencia:

$$\vec{y}^{\mathcal{P}}_{ij,\mathrm{ref}} = \mathrm{Decod}_{\mathrm{hd}}\left(\vec{y'}^{\mathcal{P}}_{ijk}\right),\ \forall k \tag{2.133}$$

Notar que, en un esquema de corrección de errores, una vez determinada la instancia y el reto (*i.e.*, "fijada" la información auxiliar), esta función de decodificación es equivalente a la aplicación de identificación definida en la sección 2.3.3, $\mathrm{Enroll} = \mathrm{Decod}_{\mathrm{hd}}$. Debido a la dependencia con el reto expuesto a la PUF, la utilización de estas técnicas de corrección se limitan habitualmente a funciones no-clonables físicamente con un espacio de retos de tamaño limitado, utilizadas en protocolos de generación de claves (sección 2.3.7), o bien en test sintéticos como el análisis "V" que se describe a continuación (sección 2.3.5).

El ECC más sencillo es el "código de repetición", en el que la información redundante se genera simplemente enviando cada bit repetido n veces, y decidiendo el bit correcto en el extremo receptor de la transmisión mediante un sistema de votación, esto es, decodificando cada símbolo como la moda de cada conjunto de n símbolos recibidos. A pesar de que esto permite corregir un número arbitrariamente grande de errores de hasta $n/2$ bits, no es una técnica realmente aplicable en un sistema de comunicación porque la capacidad del canal para transmitir información

Tab. 2.2.: Acción combinada del código de repetición y un código de sustracción en la intra-distancia promedio (%) de una PUF de osciladores de anillo. Los valores de la tabla se han representado como un mapa de color para facilitar la comparación de los códigos ECC utilizados.

		Nº de repeticiones						
		1	**3**	**5**	**7**	**9**	**11**	**15**
Nº de bits suprimidos	**0**	1,47	1,24	1,12	1,08	1,08	1,02	1,03
	1	0,96	0,74	0,62	0,57	0,58	0,51	0,51
	2	0,46	0,24	0,12	0,08	0,07	0,00	0,00
	3	0,24	0,07	0,02	0,03	0,04	0,00	0,00
	4	0,17	0,05	0,00	0,00	0,00	0,00	0,00

escala como $1/n$. No obstante, en el caso de la respuesta de una PUF, donde el número de bits recuperados es finito y conocido *a priori*, esta técnica sí resulta de aplicación. Utilizando un argumento análogo al de la sección 2.2.1, la probabilidad de que el número de apariciones del símbolo menos probable en un conjunto de n repeticiones sea mayor a $n/2$ (y por lo tanto comprometa la votación) se puede hacer arbitrariamente pequeña aumentando n. Sin embargo, tal y como se ha mostrado en (2.130), puede darse el caso de que la probabilidad de vuelco sea muy próxima al $50\,\%$ si existen parejas de celdas con distribuciones de probabilidad parecidas. En estos casos, el número de repeticiones necesarias para corregir una respuesta puede resultar prohibitivo. Para solventar este inconveniente puede aplicarse un "código de supresión", que consista sencillamente en eliminar los bits que muestran una alta tendencia a volcar o, preferiblemente, sustituir estos bits por un símbolo convenido (por ejemplo "0") para no comprometer la comunicación con una interfaz diseñada para recibir un número predeterminado de bits. De nuevo, esta técnica está particularmente bien adaptada a la tecnología PUF, donde no es relevante la forma concreta de una respuesta con tal de que esta sea reproducible y única. En la tabla 2.2 se muestra la evolución de la intra-distancia promedio para una PUF de osciladores de anillo, la cual genera una respuesta binaria de cien bits (sección 4.1), utilizando un ECC combinado de repetición/supresión para varios números de bits repetidos/suprimidos; estos resultados ilustran la eficacia de los métodos de corrección descritos.

Análisis "V"

En las aplicaciones prácticas de un dispositivo PUF descritas en la sección 2.3.7, tanto los protocolos de autenticación como de generación de claves basados en funciones no-clonables físicamente comienzan con una primera fase de evaluación de

la PUF en un entorno seguro, de modo que la respuesta es almacenada por la entidad verificadora para ser reutilizada en un futuro proceso de autenticación. Esta medida primaria se realiza en unas condiciones ambientales que serán probablemente distintas de aquellas presentes en el momento de regenerar la respuesta PUF y, por lo tanto, es fundamental estimar el grado en que un dispositivo PUF es robusto frente a variaciones ambientales. La dependencia de una función no-clonable físicamente con las condiciones del entorno, que simbolizamos como "θ" (sin perjuicio de que bajo esta denominación se englobe de hecho todo un conjunto de varias magnitudes físicas, *e.g.*, temperatura, tensión, interferencias electromagnéticas, etc.) puede hacerse explícita añadiendo estas a los argumentos de la función, $\mathcal{P}\left(\lambda, \vec{x}; \theta\right)$. Esto proporciona una manera canónica de extender la noción de "respuesta PUF" descrita en (2.83) para incluir explícitamente las condiciones ambientales:

$$\vec{y}^{\mathcal{P}}_{ijkl} \equiv \mathcal{P}\left(\lambda_i, \vec{x}_j; \theta_l\right)_k \tag{2.134}$$

y de forma análoga a (2.84), definimos el experimento PUF $\mathbf{Y}^{\mathcal{P}*}$, ampliado a una serie de N^{env} medidas en diferentes condiciones ambientales $\{\theta_l\}_{l=1}^{N^{\text{env}}}$, como:

$$\mathbf{Y}^{\mathcal{P}*}\left(N^{\text{inst}}, N^{\text{retos}}, N^{\text{rep}}, N^{\text{env}}\right) \equiv \left\{\vec{y}^{\mathcal{P}}_{ijkl}\right\}_{i=1,\ j=1,\ k=1,\, l=1}^{N^{\text{inst}}, N^{\text{retos}}, N^{\text{rep}}, N^{\text{env}}} \tag{2.135}$$

y dado que idealmente la variación de las condiciones ambientales no afectará a la respuesta PUF, la validez de los argumentos dados en la sección 2.3.2 a propósito de la distribución de distancias para una PUF ideal se conserva intacta. La manera típica de evaluar esta dependencia es a través de la intra-distancia de Hamming en diferentes condiciones ambientales ($\text{HD}^{\text{intra-env}}$), definida como:

$$\text{HD}^{\text{intra-env}}_{ijkk'll'} \equiv \text{HD}\left(\vec{y}^{\mathcal{P}}_{ijkl}, \vec{y}^{\mathcal{P}}_{ijk'l'}\right) \tag{2.136}$$

A diferencia de los casos anteriores, aquí sí hay una referencia clara para considerar unas condiciones de entorno estándar; por lo tanto, a diferencia del conjunto de intra-distancias definido en (2.97), la resistencia ambiental de una propuesta PUF se evalúa habitualmente mediante una curva "V", $\mathbf{D}^{\text{intra-env}}$, definida sobre el experimento ampliado $\mathbf{Y}^{\mathcal{P}*}$ como:

$$\mathbf{D}^{\text{intra-env}}\left(\mathbf{Y}^{\mathcal{P}*}\right) \equiv \left\{\text{HD}^{\text{intra-env}}_{ijkll_0}\right\}_{i=1,\ j=1,\ k=1,\ k'=k+1,\, l=1}^{N^{\text{inst}},\ N^{\text{retos}},\ N^{\text{rep}},\ N^{\text{rep}},\ \ \ N^{\text{env}}} \tag{2.137}$$

donde la l_0-ésima condición ambiental $\theta_{l_0} \equiv \theta_0$ respecto de la cual se comparan todas las demás es una condición de referencia, típicamente tomada como la condición estándar de operación del dispositivo. Este análisis se denomina "V" porque los valores de distancias dados en (2.137) tienden a crecer al distanciarse de la condición

de referencia θ_0, formando una figura con un mínimo en (o cerca de) la condición de referencia [105].

2.3.6. Clasificación de PUF

Típicamente, las funciones no-clonables físicamente se han clasificado atendiendo a criterios diversos:

- La tecnología de fabricación. En particular, si esta es de naturaleza electrónica o no-electrónica, lo cual determina también el método de evaluación de la respuesta PUF, así como su contexto de aplicación. Este criterio distingue entre dispositivos PUF cuya respuesta depende de la variabilidad de elementos electrónicos o no electrónicos. Las primeras propuestas de PUF corresponden a tecnologías no-electrónicas, *e.g.*, la PUF óptica de Pappu [23]. Sin embargo, dado que la interfaz de aplicación de la PUF será un dispositivo electrónico que procese la respuesta en formato digital, desde el trabajo de Gassend *et al.* [24] la práctica totalidad de la investigación sobre PUF se centra en dispositivos electrónicos integrados.

- La integración del sistema PUF en el dispositivo a proteger (implícita/explícita). Se dice que un dispositivo es una PUF implícita si no ha sido diseñado para comportarse como PUF y, sin embargo, puede extraerse una funcionalidad desafío-respuesta a partir del comportamiento normal del dispositivo. Por ejemplo, un sensor en cuya respuesta se detecta un sesgo sistemático que es reproducible y único para cada dispositivo, como es el caso de las PUF basadas en dispositivos micro-electro-mecánicos (*Micro-Electro-Mechanical System*, MEMS) [106], [107]. Por otra parte, se dice que un dispositivo es una PUF explícita si ha sido diseñado para proporcionar una determinada relación desafío-respuesta. La utilización de una PUF explícita implica su integración con el sistema al cual va a proporcionar seguridad. La forma en que se lleva a cabo esta integración depende de la aplicación y de la hostilidad del entorno en el que se va a desplegar el sistema protegido. Sin embargo, en general será preferible que esta integración tenga lugar en un nivel tan bajo como sea posible, esto es, que el diseño de PUF explícita se implemente sobre el mismo silicio que el resto del sistema [108].

- La naturaleza del parámetro físico que proporciona la respuesta, *e.g.*, tiempo, tensión eléctrica, intensidad luminosa, intensidad del campo eléctrico o magnético, etc.

- Propiedades criptográficas: PUF fuerte/débil. Tradicionalmente las propuestas PUF se han clasificado en función del número de pares desafío-respuesta que son capaces de proporcionar. Originalmente, esta distinción hacía referencia al incremento asintótico del número de pares desafío-respuesta (CRP) en función del tamaño del dispositivo PUF. Así, se denomina PUF débil si el número de CRP crece de forma polinomial con el tamaño del dispositivo, $N^{\text{CRP}} \in \mathcal{O}(N^m)$, donde N representa el tamaño del dispositivo, *e.g.*, número de transistores o celdas LUT, y $m \geq 1$ es una constante (notar que $m = 1$ corresponde al caso mínimo de una PUF que consta de un único par CRP) [109], [110]. Por otro lado, una PUF se llama fuerte si el número de pares CRP que ofrece crece de forma más rápida que polinomial [109], [111]. Sin embargo, esta definición basada en el comportamiento asintótico de un sistema físico real resulta problemática, y existen críticas relacionadas con que la cantidad de información está fundamentalmente restringida en un sistema aislado a su superficie y, por lo tanto, su incremento no puede ser más rápido que polinomial. Sin que esto descarte automáticamente que el número de CRP pudiera crecer de forma más rápida que un polinomio, sí acota la entropía de la PUF y, por lo tanto, implica el crecimiento de la correlación entre pares CRP, que a la postre limitará el número efectivo de pares desafío-respuesta a una cota polinomial. Por ello, la práctica moderna y la que seguiremos en esta tesis es limitar el concepto de "PUF débil", también llamadas "claves físicamente ofuscadas" (*Physically Obfuscated Keys*, POK) a aquellas que proporcionan un número muy limitado de respuestas (típicamente una) [112], [113]. Dado que realizar un clon matemático de un sistema así consiste en el proceso trivial de registrar la respuesta, típicamente los POK se diseñan para operar en protocolos donde la respuesta no abandona el dispositivo, por ejemplo, en la generación segura de claves donde la respuesta PUF es utilizada como semilla (sección 2.3.7). Por otro lado, se llama "PUF fuerte" a un sistema para el cual es posible encontrar un desafío cuya respuesta es impredecible para un adversario después de que este haya tenido acceso durante un tiempo prolongado al dispositivo PUF (*i.e.*, el número de pares CRP es elevado y su correlación pequeña) [29], [114].

La atención a estos criterios ha dado lugar a una taxonomía PUF detallada y completa [115], la cual mostramos parcialmente en la figura 2.10.

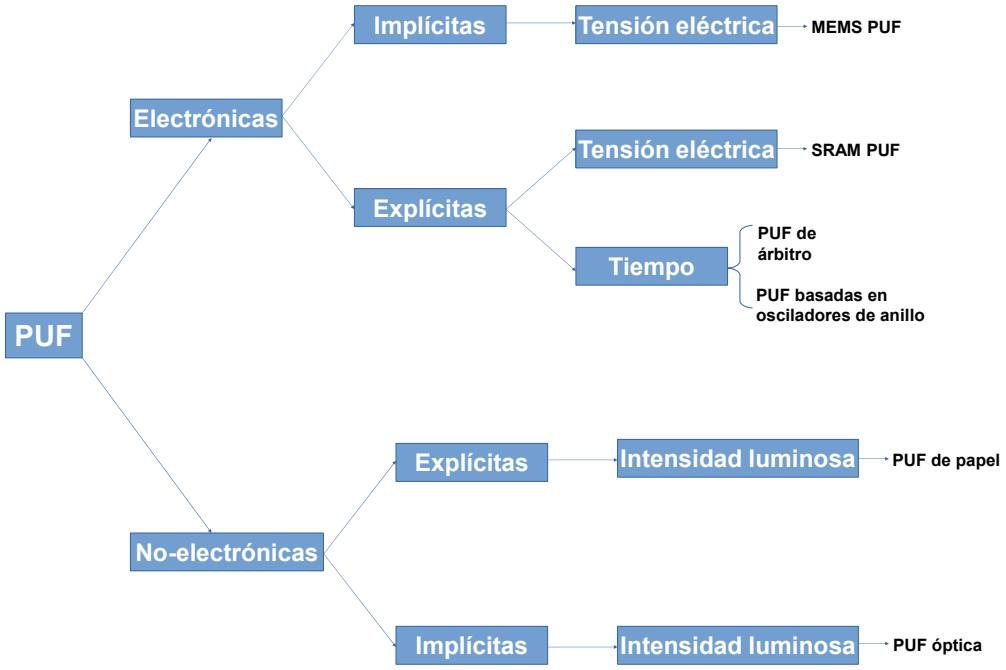


Fig. 2.10.: Taxonomía parcial de las funciones no-clonables físicamente.

2.3.7. Aplicaciones de PUF

En cuanto que primitivas criptográficas, las funciones no-clonables físicamente son aplicables en protocolos de (*i*) identificación y autenticación de dispositivos (PUF fuertes), y (*ii*) generación y almacenamiento seguro de claves (PUF fuertes o débiles).

Identificación y autenticación de entidades en un protocolo de comunicación

La autenticación es un proceso criptográfico que sigue inmediatamente a la reclamación de una identidad $\mathrm{ID}(c')$ por parte de una instancia solicitante c frente a un agente verificador v; tal y como se define en [86], este consiste en que c proporcione evidencias fehacientes de que $c = c'$ y, además, de que c estaba activa en el momento de proporcionar dicha evidencia. De este modo, la propiedad de que una instancia sea autenticable implica en general su identificabilidad, además

de requerir la capacidad de aportar pruebas adicionales respecto de su historia. Se puede decir que un determinado objeto es identificable si este es distinguible dentro de un lote "en el espacio", mientras que la propiedad de que dicho objeto sea autenticable se puede interpretar como la propiedad de que este sea distinguible "en el espacio y el tiempo". En general, un proceso de autenticación consistirá en decidir si una instancia concreta dispone de acceso a un cierto recurso. Esta condición es equivalente a ser capaz de reconocer si la instancia reclamante es la misma que en el pasado recibió el privilegio de acceder a un cierto recurso. Este marco enfatiza la autenticación como una forma fuerte de identificación; y es particularmente adecuado para el formalismo PUF introducido en la sección 2.3, donde se define una PUF como un objeto capaz de proporcionar una respuesta única y reproducible en el tiempo. Utilizando la funcionalidad desafío/respuesta como identidad inherente, una función no-clonable físicamente será autenticable de acuerdo con el esquema anterior. La aproximación a la autenticación de dispositivos mediante una instancia PUF tiene ventajas prácticas respecto de sistemas protegidos explícitamente a través de la adhesión de un secreto (*i.e.*, una clave), el cual debe ser mantenido en el tiempo mediante memorias no-volátiles (NVM) que requieren de *hardware* específico para garantizar que no se filtre información espuria de la clave fuera del dispositivo, así como para evitar la manipulación física de la memoria que almacena el secreto. Estos tres requisitos de seguridad son satisfechos de forma automática por una solución PUF, que "almacena" un secreto sin necesidad de consumir energía ni riesgo de filtración debido a que este sólo existe en el momento de ser regenerado (*i.e.*, la PUF evaluada) para su utilización. Además, es enormemente sensible a la manipulación física ya que su funcionalidad reto-respuesta depende precisamente de su microestructura, lo que dificulta la manipulación subrepticia de la PUF por parte de un adversario.

Desde el punto de vista del protocolo criptográfico, existen dos aproximaciones a la autenticación en función de la naturaleza del secreto compartido entre la instancia solicitante y el verificador, a saber, si este es una contraseña, o una relación desafío-respuesta (sección 2.2.2). Las PUF débiles pueden ser utilizadas en protocolos de la primera clase, utilizando la POK como un método de almacenamiento seguro de claves. Sin embargo, esta solución todavía requiere de un módulo criptográfico que evite la exposición directa de la respuesta POK fuera del dispositivo, difuminando la ventaja competitiva de la tecnología PUF respecto de la memoria NVM. Por otra parte, las PUF fuertes son adecuadas en protocolos CRP, resolviendo muchos de los inconvenientes previos debido a la premisa de un espacio de retos ilimitado a todos los efectos, lo que permite enviar las respuestas PUF remotamente sin necesidad de posprocesado, con la única precaución de evitar la repetición de un mismo par CRP

en distintas iteraciones del protocolo de autenticación a fin de desactivar ataques de tipo "hombre-en-el-medio" (*Man-In-The-Middle*, MITM).

El protocolo básico de autenticación basada en CRP utilizando una PUF fuerte consta de dos fases, y está diseñado para verificar la identidad reclamada por un cliente c, el cual dispone de una función no-clonable físicamente $\mathcal{P}_c \equiv \mathcal{P}(\lambda_c)$, frente a un servidor central v:

1. **Inscripción**: esta es la recopilación por parte de la entidad verificadora v de un número significativo de pares CRP para cada instancia PUF, mediante un procedimiento Enroll tal y como se definió en (2.103); estos son tabulados y almacenados en una base de datos junto con un identificador $\mathrm{ID}(c)$ asignado a cada instancia. Esta fase tiene lugar con anterioridad al despliegue de las instancias PUF, y se lleva a cabo en un entorno seguro.

2. **Verificación**: esta fase tiene lugar en un entorno hostil, y comienza con una instancia c enviando una reclamación al verificador v donde incluye la identidad reclamada, $\mathrm{ID}(c')$. A continuación, el verificador selecciona aleatoriamente un par CRP $(\vec{x}_{c'}, \vec{y}_{c'})$ correspondientes a la identidad solicitada, c', de su base de datos, y envía el desafío $\vec{x}_{c'}$ a la instancia cliente c. Esta evalúa su PUF asociada y devuelve la respuesta $\vec{y}'_c = \mathcal{P}_c(\vec{x}_{c'})$. Finalmente, el verificador juzga la autenticación positiva si la distancia (típicamente, la distancia de Hamming) entre la respuesta devuelta y aquella almacenada en su base de datos es inferior a un cierto umbral de identificación, $\mathrm{dist}(\vec{y}_{c'}, \vec{y}'_c) < u^{\mathrm{id}}_{c'}$, en cuyo caso se considera probado $c = c'$; en caso contrario la autenticación será fallida. El umbral de identificación permite ajustar el compromiso entre seguridad/operatividad, y se selecciona en función de la aplicación. Una vez este proceso ha terminado, el verificador borra el par CRP utilizado para evitar su reutilización en una iteración futura del protocolo.

Este protocolo tiene una serie de inconvenientes, *e.g.*, el número de iteraciones está limitado por el número de pares CRP que son almacenados durante la fase de inscripción; si estos se agotan, la instancia c deberá ser llevada físicamente a un entorno seguro donde el verificador pueda refrescar la base de datos, o bien el diseño PUF deberá incluir un mecanismo para permitir esto de forma remota, lo cual incluirá *hardware* criptográfico que limitará las ventajas de emplear una tecnología PUF para la autenticación. Además, dado que los pares CRP se hacen públicos una vez utilizados, este protocolo sólo es seguro para PUF que sean no-clonables matemáticamente (*i.e.*, resistentes a ataques de modelado, ver 2.3.3). Esta

condición, si bien deseable en una PUF fuerte, no es un requisito dada la definición de PUF. Finalmente, este protocolo requiere de una red estrellada que disponga de un nodo central (la instancia verificadora), y no permite la autenticación en redes P2P (*peer-to-peer*) donde no hay una jerarquía de nodos. Para evitar estos inconvenientes existen propuestas de protocolos alternativos, necesariamente más complejos que el protocolo básico pero idealmente sólo de forma marginal. Por ejemplo, la información volcada sobre el canal de comunicación remoto puede ser enmascarada mediante el uso de funciones *hash* criptográficas, permitiendo la reutilización de pares CRP y complicando el entrenamiento de un modelo matemático de la PUF [116], [117].

Generación y almacenamiento seguro de claves

La respuesta de una PUF débil se puede utilizar para sustituir la clave secreta almacenada en una memoria no-volátil como parte de un módulo criptográfico embebido en un sistema electrónico; sin embargo, a diferencia del contenido de una memoria, la respuesta PUF no es perfectamente reproducible en el tiempo y en general no se distribuye uniformemente, de modo que su utilización como clave requiere de una estrategia para corregir los bits erróneos de la respuesta. Típicamente esto se lleva a cabo codificando la respuesta PUF mediante un código ECC, que permite detectar y corregir un cierto número de inversiones de bits en sucesivas respuestas de una misma instancia PUF (sección 2.3.5). Para ello, una vez fabricada y con anterioridad al despliegue de una solución PUF, deben evaluarse las respuestas del sistema a fin de generar una secuencia de bits auxiliar que permita recuperar una respuesta de referencia después de una medida ruidosa. Este vector binario se almacenará junto con la instancia PUF utilizando alguna solución de almacenamiento no-volátil, ya que el sistema en el que se integre la función no-clonable físicamente requerirá acceder a esta información en tiempo real para regenerar la clave. Idealmente, las secuencias auxiliares sólo producen una leve reducción en la entropía de las respuestas PUF, de forma que esta información es pública y no necesita de ninguna medida específica de seguridad.

El protocolo básico de autenticación de entidades utilizando funciones no-clonable físicamente consta de dos etapas [118]:

1. **Inicialización de la clave**: esta tiene lugar una única vez al comienzo de la vida del dispositivo PUF. En esta fase se evalúa la función $\mathcal{P}$ y se obtiene una

respuesta $\vec{y}$ de N_b bits, la cual es procesada mediante un codificador ECC para producir una palabra auxiliar ("síndrome", $\vec{\text{hd}}$) de $N_b - m$ bits, $\vec{\text{hd}} = \text{ECC}(\mathcal{P})$. Este vector es público y debe resultar accesible para la PUF durante la fase de regeneración.

2. **Regeneración de la clave**: esta tiene lugar cuando el circuito PUF se utiliza para regenerar la clave criptográfica que conserva "almacenada" en la complejidad de su microestructura física. Se lleva a cabo mediante la evaluación del circuito PUF, que produce una respuesta $\vec{y}'$ de N_b bits, probablemente diferente de la respuesta obtenida en la fase de inicialización, $\vec{y}' \neq \vec{y}$, debido a la presencia de bits erróneos. Estos errores se corrigen mediante un decodificador en combinación con el síndrome generado previamente, $\vec{y} = \text{Decod}_{\vec{\text{hd}}}(\vec{y}')$. Finalmente, la respuesta regenerada se utiliza como semilla para producir una clave criptográfica segura, $\vec{z}$, a través de una función *hash*, $\vec{z} = \text{Hash}(\vec{y})$ de m bits. Notar que, en el escenario más pesimista, el síndrome no revela más de $N_b - m$ bits sobre la respuesta $\vec{y}$, de forma que la clave $\vec{z}$ conservará al menos m bits de entropía.

2.3.8. Ataques a un sistema PUF

A nivel de protocolo, las amenazas de un sistema que implemente una solución PUF son las mismas a las que se expone un sistema protegido con cualquier otra tecnología, *e.g.*, una lista fija de pares CRP para ejecutar un protocolo de autenticación basado en contraseñas de un solo uso (*One-time Password*, OTP), o una memoria no-volátil para el almacenamiento de claves. Algunas de estas amenazas incluyen ataques de denegación del servicio (un adversario impide la comunicación entre una instancia PUF y un verificador), ataques de repetición (un adversario trata de suplantar la identidad de una instancia devolviendo al verificador una respuesta previa a la que haya podido acceder), o ataques de fuerza bruta o diccionario para ganar acceso a un recurso protegido por la respuesta de una PUF débil. La minimización de estas amenazas es un objetivo clásico de la criptografía aplicada a protocolos; por ello, existe un nutrido cuerpo de técnicas dirigidas a mitigar estos problemas [119].

Sin embargo, las particularidades de las funciones no-clonables físicamente como primitivas criptográficas han dado lugar a un cuerpo de ataques dirigidos a comprometer el sistema a través de la vulneración de la instancia PUF. En general, la raíz de la confianza de un sistema protegido mediante una PUF radica en su

funcionalidad CRP. En aplicación del principio de Kerchoff no se puede confiar la seguridad del sistema a ningún parámetro operativo, esto es, debe considerarse que el posprocesado de las respuestas PUF (por ejemplo, el proceso que ofusca la respuesta plana de un POK) es público, así como el conjunto de posibles retos para una cierta arquitectura, etc. Por lo tanto, diremos que un adversario ha comprometido una instancia PUF si es capaz de imitar total o parcialmente la funcionalidad CRP de la misma; dado que la PUF es no-clonable físicamente por definición, esto equivale a crear un modelo matemático de acuerdo con el formalismo dado en la sección 2.3.3. Estos ataques generales se denominan "ataques de modelado" contra la PUF, y en la literatura es habitual encontrarlos categorizados en función de criterios diversos, por ejemplo, la estrategia seguida para construir el modelo (*e.g.*, "ataques de *machine learning*"), o la técnica utilizada para acceder al subconjunto de pares CRP u otra información utilizada para entrenar el modelo (*e.g.*, ataques de canal paralelo basados en la monitorización de la radiación electromagnética emitida por una matriz de osciladores de anillo para estimar sus frecuencias características).

2.4. Implementación física

Tal y como se ha expuesto en la sección 1.2.1, la tecnología PUF está concebida para ser integrada en un ecosistema digital; por la tanto, la elección de una plataforma *hardware* adecuada sobre la cual implementar este sistema debe comenzar con la elección de una familia lógica, esto es, un esquema de diseño e implementación de puertas lógicas a nivel de sus componentes electrónicos, lo cual determina la tensión de alimentación, V_{dd}, y los niveles de tensión interpretados como "1" y "0" lógicos. La principal familia lógica en uso actualmente es la tecnología de metal-óxido-semiconductor complementario (CMOS) y sus variantes de baja tensión, las cuales ocupan la práctica totalidad de la electrónica de consumo [120]. En esta familia, las puertas lógicas se fabrican utilizando pares duales de transistores metal-óxido-semiconductor de efecto de campo (MOSFET) tipo "P" (PMOS) y tipo "N" (NMOS) de forma complementaria, esto es, de manera que el circuito eléctrico que conforma la puerta siempre esté abierto, ya sea por la red de transistores NMOS, o de forma complementaria, por sus duales PMOS. Esta topología reduce drásticamente el consumo de energía estático, lo cual permite densidades de integración extremadamente elevadas en la lógica CMOS. A pesar de mostrar inicialmente velocidades de conmutación inferiores a sus contrapartidas basadas en transistores bipolares, esta ventaja ha impulsado la tecnología CMOS como la

alternativa dominante en la microelectrónica actual, alcanzando velocidades de operación de nanosegundos. Por ello, existen una gran cantidad de soluciones de integración (*i.e.*, circuitos integrados) que utilizan esta tecnología, entre las cuales destacan:

- Circuitos integrados "*full-custom*", el diseñador es responsable de trasladar al fabricante el circuito descrito a nivel de transistor; esta opción ofrece la máxima capacidad de optimización de un proceso tecnológico concreto, sin embargo implica la fabricación de máscaras litográficos específicas, lo que conlleva unos costes desorbitados referidos habitualmente como "costes de ingeniería no-recurrentes" (*Nonrecurring Engineering Costs*, NRE), del orden de los cientos de millones de euros. Así mismo, los plazos típicos de fabricación pueden exceder el año.

- Circuitos integrados digitales de propósito específico (*Application Specific Integrated Circuit*, ASIC). En esta opción de integración, el diseñador se limita a utilizar una serie de celdas estándar, las cuales son proporcionadas por un tercero y contienen los diseños a nivel de transistor de las puertas lógicas y elementos secuenciales característicos de una cierta tecnología digital; de forma que el diseñador proporciona al fabricante la disposición de celdas estándar en su diseño, así como sus interconexiones. Dado que las máscaras litográficas de las celdas están preconstruidas, esta opción tiene unos costes NRE menores que el chip *full-custom*, así como tiempos de fabricación menores.

- Matriz de puertas programable en campo (FPGA). Este es un circuito integrado prefabricado y vendido como "producto de caja" (*Commercial Off-the-shelf*, COTS) programable. El funcionamiento de una FPGA se basa en la idea de que una memoria puede ser utilizada para implementar lógica combinacional: dada una memoria con capacidad para 2^N bits, el proceso de recuperar alguno de los bit almacenados consistirá en seleccionar una dirección de memoria utilizando N líneas de entrada. Desde un punto de vista de "caja negra", la memoria será un bloque combinacional (*i.e.*, sin entrada de reloj) con N entradas y una salida, denominadas "tablas de búsqueda" (*Look-up Tables*, LUT) en el contexto de las FPGA; permitiendo una alta densidad de "puertas" lógicas. Además, una FPGA incluye una serie de circuitos secuenciales *flip-flop* que permiten implementar lógica secuencial. Finalmente, los distintos elementos de una FPGA se conectan entre sí a través de nodos de multiplexores denominados "matrices de interconexión". En la sección 2.4.1 se proporcionará una descripción más detallada de la estructura interna de las FPGA. Estos dispositivos permiten im-

plementar circuitos digitales complejos mediante la programación vía *software*, por lo cual se ha convertido en un actor estándar del flujo de diseño digital como plataforma de pruebas y prototipado, perfilándose como la solución de integración COTS más utilizada en la industria del diseño microelectrónico digital. Además, la migración progresiva a nodos tecnológicos más pequeños, más eficientes energéticamente y capaces de ofrecer una mayor densidad de integración, tiene un impacto significativo en reducir la brecha en cuanto a rendimiento entre los dispositivos FPGA y las soluciones ASIC digital. Este fenómeno está llevando a la progresiva adopción de FPGA como sustitución de los circuitos integrados de propósito específico incluso en productos finales.

- Procesador programable, esto es un circuito digital caracterizado por que la computación que ejecuta en un instante dado se encuentra almacenada en una memoria, en lugar de implementada físicamente en la topología del circuito. Esto permite que circuitos digitales con un mismo diseño realicen tareas muy diferentes, a costa de importantes sacrificios en cuanto a rendimiento respecto de un circuito diseñado *ad hoc* para una determinada tarea. Las unidades centrales de procesamiento que ejecutan un sistema operativo y constituyen el componente central de los ordenadores personales, así como los procesadores embebidos en sistemas de control específicos (por ejemplo, las luces de un semáforo) son ejemplos de procesadores programables.

- Unidad de procesamiento gráfico (*Graphic Processing Unit*, GPU), es un caso particular de procesador programable diseñado específicamente para realizar operaciones en coma flotante con una alta paralelización. Como en el caso de procesador general, dispone de una memoria que almacena una sucesión de instrucciones, indicando la computación, *i.e.*, la transformación que debe aplicar a cada dato de entrada. Las GPU nacieron en la década de 1980 como co-procesadores para descargar a los procesadores centrales (*Central Processing Unit*, CPU) del trabajo de procesar las transformaciones de los píxeles individuales mostrados en un monitor, de tal modo que estas fueran realizadas de forma paralela y eficiente. Actualmente, esta capacidad es utilizada en ámbitos computacionalmente intensivos donde las posibilidades de paralelización son muy grandes, *e.g.*, simulaciones de química computacional, o encriptación de datos [121].

A la hora de elegir una solución de implementación para un proyecto digital, los principales factores a valorar son el grado de optimización, coste de desarrollo, duración del ciclo de diseño y capacidad de actualización del *hardware* una vez

desplegado. Las opciones *full-custom* y ASIC proporcionan excelentes rendimientos, pero requieren tiempos de fabricación largos y no permiten la actualización del diseño una vez implementado. Además, sus costes NRE elevados los convierten en opciones viables únicamente para grandes volúmenes de unidades, por ejemplo módulos montados sobre productos de electrónica de consumo masivos (*e.g.*, teléfonos móviles), o bien elementos sin restricción presupuestaria y que requieren de rendimientos máximos (*e.g.*, industrias nacionales de defensa o aeroespacial). En cuanto a las opciones COTS, estas ofrecen menor densidad de integración y mayor consumo energético, ya que la flexibilidad de que dispone el diseñador para optimizar la implementación física del circuito es menor. Sin embargo, la reprogramabilidad de estos dispositivos permite actualizar el *hardware* después de haber sido desplegado, adaptando el sistema a un entorno cambiante. En relación al coste de despliegue, este está determinado únicamente por el coste de adquisición de cada dispositivo COTS ya que carecen de NRE. Análogamente, dado que el tiempo requerido para la implementación física es virtualmente cero, el plazo de llegada al mercado está limitado únicamente por el tiempo de diseño del proyecto.

Tomando en consideración estas cuestiones, decidimos optar por la solución de integración FPGA. Por construcción, las funciones no-clonables físicamente que son objeto de este trabajo operan en el límite de incertidumbre de las herramientas de simulación microelectrónica, lo cual redunda en el peso específico de la evaluación experimental y, por lo tanto, el factor del tiempo de acceso a la implementación física de los prototipos diseñados es determinante. Las FPGA ofrecen un buen rendimiento y densidad, además de existir precedentes (y de hecho, una tendencia creciente) en la utilización de dispositivos FPGA para sistemas embebidos, en particular para IoT (sección 1.2.2). Respecto de la amplia oferta de modelos comerciales, utilizaremos la solución "sistema en chip" (*System-on-Chip*, SoC) Zynq-7000 de Xilinx, que consta de una FPGA Artix 7 construida en tecnología CMOS de 28 nanómetros y un microprocesador ARM Cortex-A9 , integrados conjuntamente en un único chip. De este modo, utilizaremos la FPGA como soporte para la implementación microelectrónica, y las capacidades del microprocesador para establecer una interfaz de comunicación ordenador – FPGA (apéndice D). A continuación describimos en detalle algunos aspectos estructurales del modelo FPGA seleccionado, que serán de relevancia a lo largo de esta tesis.

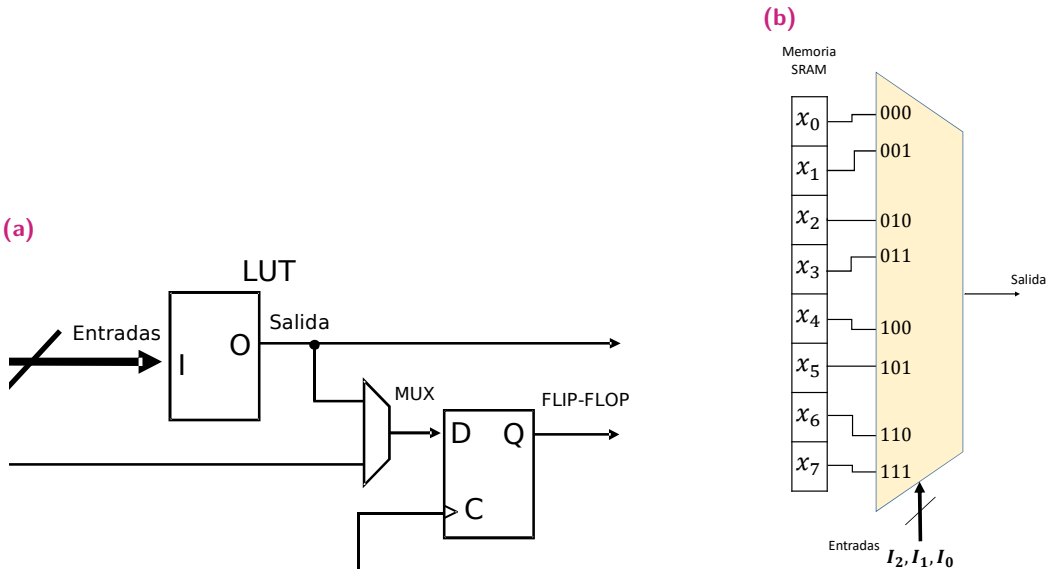


Fig. 2.11.: (a) Esquema de celda lógica (LC) en FPGA. (b) Esquema de tabla de búsqueda (LUT) de tres entradas.

2.4.1. FPGA

Una FPGA es un dispositivo electrónico digital que consta de una serie de elementos combinacionales y secuenciales preconstruidos de tipo muy general (circuitos biestables sencillos, multiplexores, etcétera), lo cual permite a un diseñador implementar un circuito digital arbitrario mediante la adecuada interconexión de los recursos disponibles en el dispositivo.

Estructura interna y elementos físicos constitutivos de una FPGA

Una FPGA consta de varios elementos físicos integrados los cuales se pueden reconocer en diversos niveles organizativos. El elemento básico constitutivo de una FPGA que resulta accesible para el diseñador es la "celda lógica" (*Logic Cell*, LC)[14], tal y como se muestra en la figura 2.11a. Desde un punto de vista electrónico, esta unidad mínima consta de una LUT, un registro *flip-flop* y un multiplexor que permite utilizar la salida de la LUT de forma síncrona, *i.e.*, registrada en el *flip-flop*, o asíncrona. Una LUT es el elemento programable elemental de una FPGA, capaz

[14]Nomenclatura de Xilinx; otros fabricantes emplean nombres diferentes para cada elemento y nivel organizativo, sin embargo el contenido de estos es análogo.

de implementar cualquier función booleana no recursiva de N variables, donde N es el número de entradas de la LUT; históricamente, las propuestas FPGA más populares han implementado LUT de tres a seis entradas[15] [122]. Físicamente, una LUT es un circuito combinacional formado por una memoria SRAM con capacidad para 2^N bits, de tal modo que la "entrada de la LUT" son los N bits utilizados para el direccionamiento de la memoria: así, para cada una de las 2^N posibles combinaciones de entradas, el circuito LUT extraerá un bit previamente guardado en la correspondiente dirección de la memoria SRAM. Seleccionando cuidadosamente el contenido de esta memoria dispondremos de la implementación física de una función lógica arbitraria. En la figura 2.11b hemos representado esquemáticamente una LUT de tres entradas; notar que, por ejemplo, si introducimos en esta memoria SRAM el vector $(x_0, \cdots x_7) = (0, 0, 0, 0, 0, 0, 1)$, la LUT resultante realizará una puerta AND de tres entradas, cuya salida será $I_2 * I_1 * I_0$; en cambio, si almacenamos los bits $(0, 1, 1, 1, 1, 1, 1)$, el elemento resultante equivaldrá a una puerta lógica OR de tres entradas, de salida igual a $I_2 + I_1 + I_0$. El *software* de automatización del diseño electrónico (*Electronics Design Automation*, EDA) Vivado de Xilinx permite resolver tanto elementos LUT como *flip-flop* individualmente a través de sendos modelos descritos en lenguaje de descripción de hardware (*Hardware Description Language*, HDL). Para el caso de las LUT se proporcionan modelos diferentes en función del número de señales de entrada que se vayan a utilizar, siendo estos: LUT1, LUT2, LUT3, LUT4, LUT5, o LUT6 para tablas de una, dos, tres, cuatro, cinco o seis entradas, respectivamente. Por otro lado, los *flip-flops* son circuitos biestables que actúan como registros, transfiriendo el valor lógico presente en la entrada "D" a la salida "Q" cada vez que la señal en la entrada "C" realiza un recorrido lógico ascendente $(0 \rightarrow 1)$, típicamente el flanco ascendente de una señal de reloj. Como en el caso anterior, Vivado proporciona modelos HDL de estos elementos.

El siguiente nivel organizativo en la FPGA es la "sección lógica" (*logic slice*, al que nos referiremos como *slice*); estos están formados por cuatro elementos básicos (*Basic Element*, BEL), etiquetados A, B, C y D (figura 2.12), cada uno de los cuales consta de una celda lógica. Cada *slice* se identifica inequívocamente en la matriz de la FPGA mediante dos coordenadas X e Y, y varios de estos (típicamente dos o cuatro, en función del modelo de FPGA) se agrupan en una estructura superior denominada "bloque lógico configurable" (*Configurable Logic Block*, CLB), que se muestra esquemáticamente en la figura 2.13. En el modelo de FPGA Artix 7 empleado en este trabajo, cada CLB está compuesto por dos *slice* identificados con los índices

[15] Los modelos utilizados en este trabajo han sido integrados en 28 nm de la serie 7 de Xilinx, que montan LUT las cuales se pueden configurar con seis entradas/una salida, o cinco entradas/dos salidas. En este trabajo se ha utilizado únicamente la primera configuración.

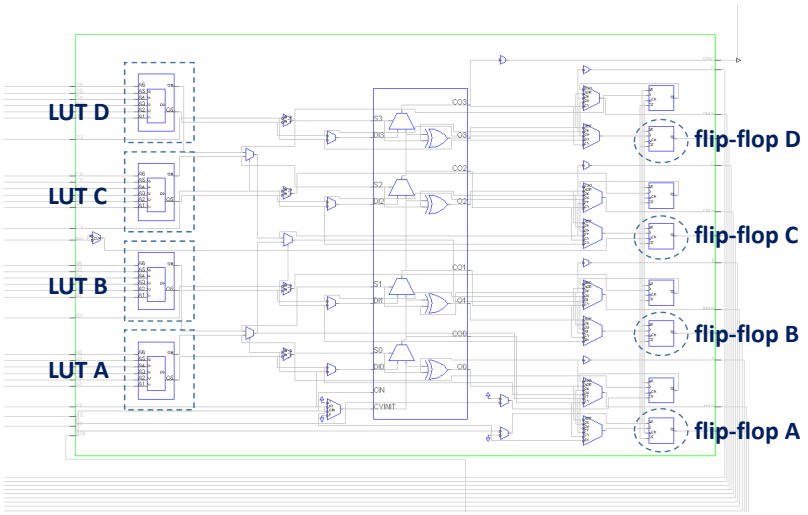


Fig. 2.12.: Detalle de una *slice*. Se han destacado los recursos que se implementan en detalle (LUT y *flip-flop*).

"0" y "1", que denominaremos *slice* (0) y *slice* (1) respectivamente. En la matriz FPGA, los *slice* (0) se corresponden con *slice* de coordenada X par, mientras que el *slice* (1) se corresponde con una X impar. Además, existen dos tipos diferentes de LUT, denominadas "L" y "M"; las primeras son utilizadas exclusivamente para implementar funciones booleanas, mientras que las segundas pueden ser utilizadas como memoria RAM, denominada "RAM distribuida" para distinguirla de los grandes bloques de memoria RAM que las FPGA modernas también suelen incluir. Finalmente, tal y como se puede apreciar en la figura 2.12, cada CLB está acompañado de una caja de conexiones (*switch box*) que se encarga de enrutar las entradas/salidas de cada *slice* hacia elementos del mismo *slice*, *slice* diferentes dentro del mismo CLB, o CLB distintos dentro de la FPGA.

El nivel más alto de organización que introducimos en este trabajo implica a un conjunto de CLB, tal y como puede verse en la figura 2.14, donde se ha representado la disposición de varios CLB junto con sus respectivas cajas de conexiones. Además de las diferencias entre *slice* (0/1) y LUT (L/M) que se han advertido, también pueden darse dos disposiciones diferentes de la caja de conexiones respecto del CLB, a saber, este puede situarse a la derecha (*slice* X0 y X1 en la figura 2.14) o a la izquierda (*slice* X2 y X3). Esta distinción es irrelevante para el comportamiento lógico de un diseño, sin embargo sí introduce diferencias a nivel físico que pueden llegar a tener un impacto significativo en el comportamiento PUF de una determinada estructura. Por lo tanto, en general, distinguiremos ambos casos calificando un CLB

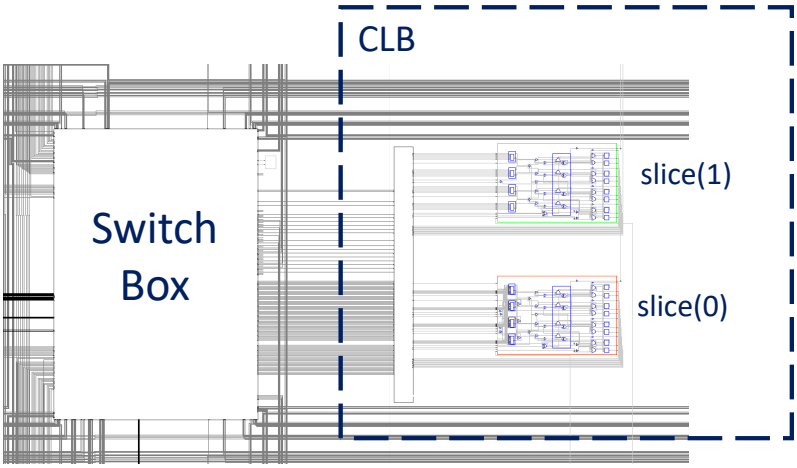


Fig. 2.13.: Representación esquemática de los recursos de un bloque lógico configurable (CLB).

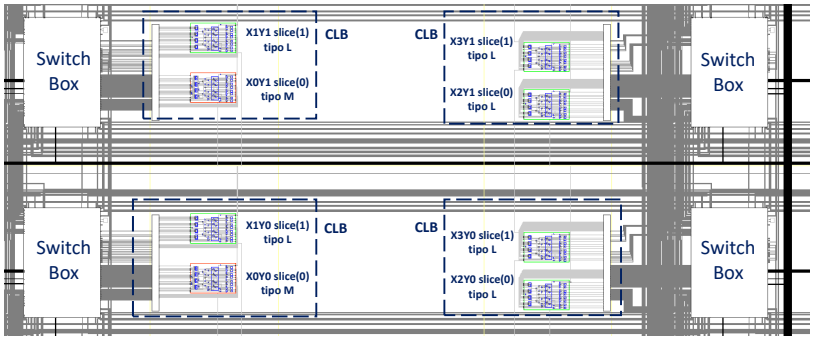


Fig. 2.14.: Representación esquemática de los recursos de una FPGA detallando la disposición de cuatro CLB con sus respectivas cajas de conexiones a izquierda y a derecha.

como “I” cuando este se encuentre a la izquierda de su caja de conexiones, y “D” en caso contrario.

Ruteado en FPGA

Las LUT del modelo de FPGA utilizado constan de seis entradas; cada una de estas es accedida a través de unos recursos de conexionado predefinidos, los cuales incluyen la caja de conexiones asociada al propio CLB, así como otras distribuidas sobre la superficie del chip. De esta manera, cada ruta conectando dos puntos de la FPGA se puede especificar como una lista de nodos desde la salida del elemento que

Tab. 2.3.: Entradas LUT para las cuales el número de nodos de conexionado es mínimo dentro de un mismo CLB.

LUT	**slice**		***Load***							
			(0)				(1)			
		BEL	A	B	C	D	A	B	C	D
Driver	(0)	A	6	5	1	1	3	3	2	2
		B	4	4	6	6	2	2	3	3
		C	6	6	4	4	3	3	2	2
		D	1	1	5	6	2	2	3	3
	(1)	A	3	3	2	2	6	5	1	1
		B	2	2	3	3	4	4	6	6
		C	3	3	2	2	6	6	4	4
		D	2	2	3	3	1	1	5	6

genera la señal (*driver*) hasta el elemento que recibe la señal (*load*). En la tabla 2.3 se muestran las entradas de la LUT *load* tales que la LUT *driver* perteneciente a su mismo CLB se conecta a través de una ruta para la que el número de nodos es mínimo, y en particular todos pertenecen al mismo CLB. Este nivel de control en el diseño será necesario a la hora de implementar estructuras físicas con buenas propiedades PUF, especialmente en el caso de funciones explícitas que deben implementarse junto con diseños diversos y, por lo tanto, es deseable que el módulo PUF sea lo más compacto posible. Por ejemplo, para implementar un oscilador de anillo de tres etapas inversoras situadas en las posiciones BEL B, C y D, junto con una puerta AND inicial situada en BEL A, de forma que el número de conexiones en el bucle de realimentación sea mínimo y todos los elementos estén en *slice* (0); acudimos a la tabla 2.3 para seleccionar la entrada de la LUT B (*load*) tal que la salida de A (*driver*) se conecta con ella de forma mínima, obteniendo para este caso la entrada “5”. A continuación, buscamos la entrada de la segunda etapa inversora LUT C (*load*) tal que la salida de la LUT B precedente (*driver*) se conecta con ella por una ruta mínima, obteniendo la entrada “6” para la LUT C. Repetimos este proceso para las dos conexiones restantes, *i.e.*, LUT C a LUT D, y LUT D a LUT A, obteniendo respectivamente las entradas “4” y “1”. De etse modo, para implementar un oscilador de anillo cuyo bucle de realimentación tiene un número mínimo de nodos de conexión, debemos asignar las entradas lógicas de las LUT A, B, C y D a los pines físicos 1, 5, 6 y 4 respectivamente.

2.5. Conclusión

En este capítulo se ha proporcionado una introducción a la teoría de la seguridad de la información y se ha definido formalmente el concepto de "criptosistema". A continuación, hemos profundizado en la noción de función no-clonable físicamente a través de una propuesta de formalización que permite capturar tanto las características de una PUF en tanto que dispositivo físico, como facilitar su integración como primitiva criptográfica en un protocolo de comunicaciones seguras. En relación a esto último, se han enunciado explícitamente los límites ideales de una función no-clonable físicamente, y se ha propuesto un modelo cuasi-ideal, el cual permite evaluar el grado de idealidad de una PUF atendiendo a las métricas medidas típicamente en un experimento PUF, a saber, las distribuciones de intra/inter-distancias de Hamming de las respuestas binarias.

También se ha explorado en detalle el concepto de compensación de la medida en PUF, ilustrado por medio de una estimación para la probabilidad de inversión de un bit de respuesta en el caso de una matriz de osciladores de anillo implementada sobre FPGA, y se ha introducido la técnica de corrección de errores, dando así mismo ejemplos de su aplicabilidad a sistemas implementados en FPGA.

Finalmente, se ha examinado en detalle la arquitectura interna de los dispositivos FPGA, profundizando en el modelo utilizado para este trabajo y destacando aquellos aspectos estructurales que tienen una influencia máxima en el diseño e implementación de una función no-clonable físicamente.

3 Extracción de entropía en PUF de medida compensada

La medida compensada es una técnica de digitalización con importantes ventajas prácticas a la hora de diseñar un sistema PUF, destacando el robustecimiento de la respuesta frente a variaciones ambientales de voltaje y temperatura en el entorno de operación. En este capítulo estudiamos de forma detallada las distribuciones de probabilidad que emergen como resultado del esquema de pares de celdas utilizado para generar la respuesta binaria, con especial énfasis la influencia de esta decisión de diseño en las propiedades de seguridad del sistema, particularmente en relación a deficiencias de uniformidad y cómo esto se traduce en vulnerabilidades susceptibles de ser explotadas criptoanalíticamente. Las métricas fundamentales utilizadas para esta caracterización han sido la entropía y la minentropía, definidas en (2.31) y (2.50) respectivamente. La importancia de la entropía de Shannon en la evaluación de sistemas criptográficos fue discutida detalladamente en el capítulo 2, a propósito de la cual destaca la relación entre la entropía de un espacio de claves y la condición de cifrado perfecto dada en 2.66. No obstante, y a pesar de la popularidad de que goza esta magnitud como una norma estándar para comparar diferentes soluciones criptográficas abstraídas de los detalles de su implementación [123], existen críticas al uso de la entropía de Shannon como medida de calidad en el ámbito de la criptografía[1] en favor de la minentropía [124]-[126]. Finalmente, en este capítulo también proponemos una nueva familia de topologías, bautizada como “K-modular”, que exhiben una excelente uniformidad en la distribución de respuestas así como una alta densidad de entropía tanto en el espacio físico como en el espacio de respuestas binarias, y un buen compromiso en cuanto al nivel de seguridad proporcionado frente al consumo de potencia y superficie de silicio.

[1] Estas están relacionadas con el hecho de que, incluso si la cantidad de información promedio de un cierto espacio de claves es alta, existe la probabilidad —quizá no despreciable— de seleccionar una clave concreta con una cantidad de información pequeña, comprometiendo la seguridad del sistema.

3.1. Topologías PUF de medida compensada

Tal y como se discutió en la sección 2.3.5, en general el conjunto completo de bits extraíbles al realizar todas las posibles comparaciones en una matriz de N celdas presentará fuertes correlaciones. La predicción de qué comparaciones serán independientes y proporcionarán respuestas de entropía máxima es difícil de realizar y depende de cada instancia específica, de modo que para una aplicación concreta la opción de diseño preferible es seleccionar *a priori* un subconjunto de parejas de celdas, y utilizar estas para producir la respuesta PUF. El conjunto de tales pares de celdas que son comparadas entre sí para producir un bit se puede representar mediante un grafo dirigido (digrafo) $D = (V, A)$, donde $V = \{i\}_{i=1}^{N}$ es el conjunto de vértices y $A \subset V \times V$ es el conjunto de aristas del digrafo, expresada cada una como una dupla ordenada (i, j), $i \neq j$, indicando que dicha arista conecta los vértices correspondientes con un sentido definido, $i \to j$. Dado que tanto la numeración de celdas en una N-tupla construida físicamente como la elección de paridad en la definición de una función "bit" (2.115) son arbitrarias, definimos una topología $\mathcal{T}_N$ como una clase de equivalencia formada por los digrafos de N vértices cuyos grafos subyacentes[2] son isomorfos entre sí, donde el grafo $G' = (V', E')$ se dice isomorfo a G si existe una función biyectiva $f : V \longrightarrow V'$ tal que $\{i, j\} \in E \longleftrightarrow \{f(i), f(j)\} \in E'$; con esta definición de equivalencia, dos digrafos pertenecerán a una misma topología con tal de que el número de vértices mutuamente conectados y el número de aristas total sea el mismo. Esto se ilustra en la figura 3.1a, donde se han representado varios grafos pertenecientes a una misma topología a pesar de presentar aspectos diferentes, frente al caso opuesto ejemplificado en 3.1b, donde se muestran varios grafos pertenecientes a topologías diferentes pese a disponer del mismo número de celdas $N = 4$. En la realización física de un diseño PUF, una topología $\mathcal{T}_N$ estará representada típicamente por algún digrafo concreto $D_0 \in \mathcal{T}_N$, el cual se puede interpretar como una función actuando sobre un vector de respuestas físicas $\vec{\psi}$ para producir un vector binario $\vec{y} = D_0(\vec{\psi})$, el cual consta de tantos elementos como aristas presente el digrafo D_0, y cada elemento estará construido mediante la comparación del par de celdas conectadas por un arco, *i.e.*, el digrafo D_0 constituye una función de codificación binaria ADC (*analog-to-digital converter*) tal y como se definió en (2.78). Cuando en una aplicación concreta no haya ambigüedad en el digrafo representante de una topología (por ejemplo, porque se haya convenido una manera de etiquetar cada

[2] El grafo $G = (V, E)$ subyacente a un digrafo $D = (V, A)$ es el grafo construido mediante la sustitución de cada arco dirigido por una arista no dirigida, $\{i, j\} \in E \longleftrightarrow (i, j) \in A \vee (j, i) \in A$.

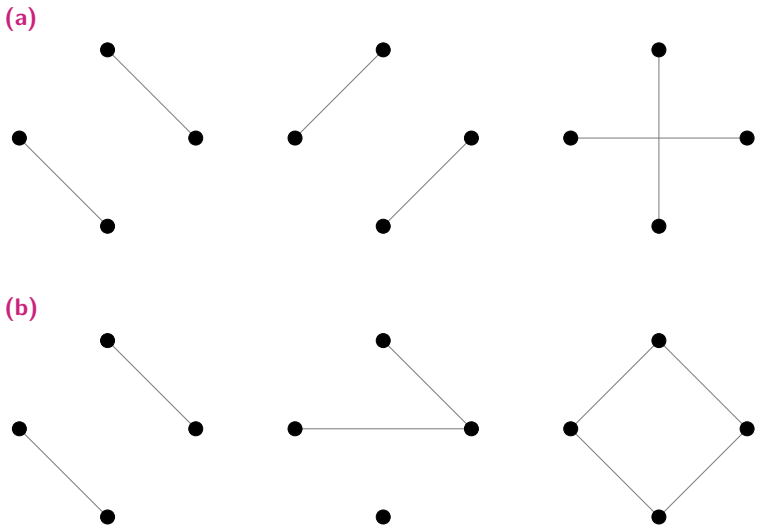


Fig. 3.1.: Ejemplo de grafos pertenecientes a una misma topología de $N = 4$ vértices: (a) equivalentes y (b) no-equivalentes. Cada punto (vértice) representa una celda y cada enlace (arista) una comparación, *i.e.*, un bit de salida.

celda física), abusaremos de la notación para escribir un vector binario como la imagen de la propia topología actuando sobre un vector de respuestas físicas $\vec{\psi}$:

$$\mathcal{T}_N(\vec{\psi}) = \vec{y} \equiv \vec{y}_{\mathcal{T}_N} \longleftrightarrow \mathcal{T}_N = \mathrm{ADC} \tag{3.1}$$

y análogamente denotamos la variable aleatoria inducida por esta relación:

$$\mathcal{T}_N\,(\Psi) = \mathsf{Y} \equiv \mathsf{Y}_{\mathcal{T}_N} \tag{3.2}$$

la cual se distribuye con probabilidad $p_{\vec{y}}(\mathcal{T}_N) \equiv \mathrm{Prob}(\mathsf{Y}_{\mathcal{T}_N} = \vec{y})$.

Finalmente, diremos que un subsistema dado por una topología $\mathcal{T}'_N$ es un módulo del sistema $\mathcal{T}_{N+M}$ si, para cualquier grafo $G \in \mathcal{T}_{N+M}$, existe un subgrafo[3] $G' \subseteq G$ que pertenece a $\mathcal{T}'_N$. Así mismo, diremos que dos módulos del sistema $\mathcal{T}_{N+M}$ dados por las topologías $\mathcal{T}'_N$, $\mathcal{T}''_M$ son inconexos si, para cualquier grafo $G \in \mathcal{T}_{N+M}$, los subgrafos $G' \in \mathcal{T'}_N$, $G'' \in \mathcal{T}''_M$ son inconexos, *i.e.*, la intersección de sus respectivos espacios de vértices es vacía, $V' \cap V'' = \emptyset$. En la figura 3.2a se ha representado esquemáticamente un sistema de $N = 7$ celdas separado en dos módulos inconexos, que se muestran así mismo de forma independiente en 3.2b. Utilizando esta nomenclatura, diremos que una topología $\mathcal{T}_{N+M}$ definida sobre un

[3] Un grafo $G' = (V', E')$ es un subgrafo de $G = (V, E)$ si y sólo si $V' \subseteq V$ y $E' \subseteq E$, y lo denotamos $G' \subseteq G$.

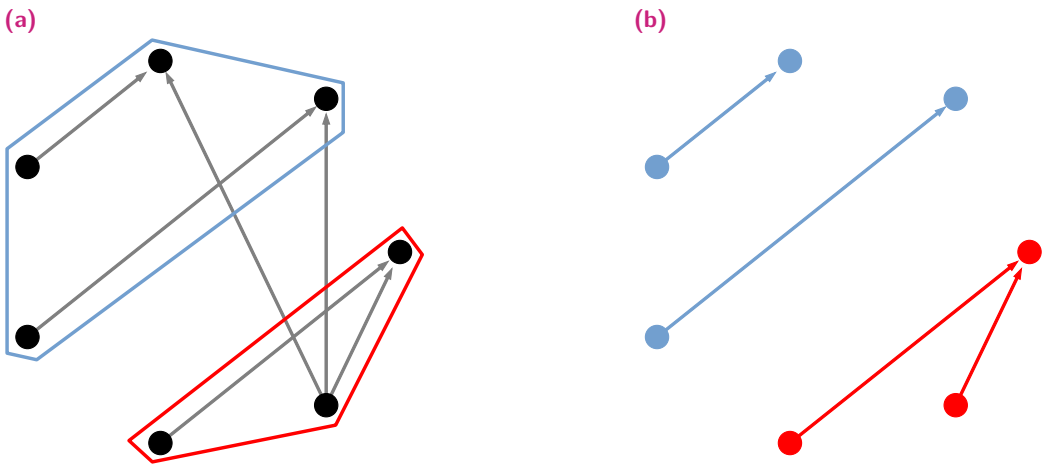


Fig. 3.2.: (a) Topología modular $\mathcal{T}_7$ de $N = 7$ vértices, donde se han destacado dos módulos inconexos $\mathcal{T}_4'$ de $N = 4$ vértices (cuadro azul), y $\mathcal{T}_3''$ de $N = 3$ vértices (cuadro rojo). (b) Topologías $\mathcal{T}_4'$ y $\mathcal{T}_3''$ representadas de forma independiente.

sistema de $N + M$ celdas es modular cuando se puede separar en módulos disjuntos, $\mathcal{T}_N$, $\mathcal{T}_M$, de forma que la variable aleatoria $\mathcal{T}_{N+M}(\Psi_{N+M}) = \mathsf{Y}_{\mathcal{T}_{N+M}}$ se puede escribir como la concatenación $\mathsf{Y}_{\mathcal{T}_{N+M}} = (\mathsf{Y}_{\mathcal{T}_N}, \mathsf{Y}_{\mathcal{T}_M}, \Delta)$, donde $\mathsf{Y}_{\mathcal{T}_N} = \mathcal{T}_N(\Psi_N)$, $\mathsf{Y}_{\mathcal{T}_M} = \mathcal{T}_M(\Psi_M)$, y la variable aleatoria binaria Δ contiene los bits de la interacción cruzada entre módulos. Notar que dado que los módulos son físicamente inconexos, las variables aleatorias $\mathsf{Y}_{\mathcal{T}_N}$, $\mathsf{Y}_{\mathcal{T}_M}$ son independientes entre sí.

3.2. Modelo de fabricación

Tal y como se introdujo en la sección 2.3.5, una función física diseñada bajo un esquema de medida compensada se puede expresar como la concatenación de N funciones físicas elementales (celdas) idénticas por diseño; siguiendo la notación presentada en (2.67), escribimos una tal función física modular de N celdas como $\mathcal{F}^N(\vec{\lambda}, \vec{\xi}) \equiv [\mathcal{F}(\lambda_1, \xi_1), \cdots, \mathcal{F}(\lambda_N, \xi_N)] = \vec{\Psi}|\vec{\lambda}$. En este capítulo supondremos que el único estímulo físico aceptado por la función es el propio acto de medir, lo cual permite eliminar la dependencia explícita de la cantidad "ξ" en las funciones físicas, así como del reto "$\vec{x}$ " en las funciones no-clonables físicamente, y escribir $\mathcal{F}^N = \mathcal{F}^N(\vec{\lambda})$. Además, supondremos que la distribución de respuestas físicas de

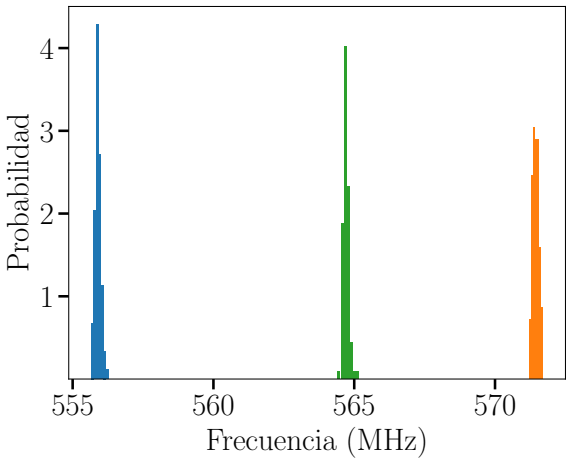


Fig. 3.3.: Distribuciones de probabilidad para las frecuencias de oscilación de tres anillos implementados en FPGA.

cada celda $\mathcal{F}(\lambda_i) = \Psi_i|\lambda_i$ es ideal, lo cual se puede expresar utilizando la distribución delta de Dirac "δ" en virtud de (2.68) como:

$$p_{\psi_i|\lambda_i} = \delta\left(\psi_i - \psi_i^0\right) \tag{3.3}$$

de donde se sigue la identidad $\psi_i^0 = \mathcal{F}(\lambda_i)$; esta forma de modelar la respuesta de una celda física se justifica en el resultado (2.130), donde se demuestra que la probabilidad de vuelco de un bit en una PUF de medida compensada es de hecho una medida de la dispersión entre mediciones de diferentes celdas en relación con la anchura con la que se distribuyen los valores típicos para una misma celda. Bajo la hipótesis de que se está modelando una buena PUF, las probabilidades de vuelco serán pequeñas y por tanto la anchura típica de las distribuciones de cada celda σ en relación con la desviación típica de sus valores promedio $|\Delta\mu|$ cumplirá $\sigma \ll |\Delta\mu|$, lo cual permite tomar la delta de Dirac como una primera aproximación razonable; esto se ha ilustrado en la figura 3.3, donde se representan las distribuciones de frecuencia obtenidas para tres osciladores de anillo idénticos implementados en distintas localizaciones de un mismo chip FPGA (las PUF de medida compensada basadas en osciladores de anillo se estudiarán con detalle en el capítulo 4). El modelo dado por la expresión (3.3) se extiende de forma natural a una N-función física:

$$p_{\vec{\psi}|\vec{\lambda}} = \delta\left(\vec{\psi} - \vec{\psi}^{\,0}\right) = \delta\left[\vec{\psi} - \mathcal{F}^N(\vec{\lambda})\right] \tag{3.4}$$

con $\vec{\psi}^{\,0} = \mathcal{F}^N(\vec{\lambda})$. Utilizando esta expresión podemos calcular la distribución de probabilidad para la respuesta binaria $p_{\vec{y}|\vec{\lambda}}$ mediante (2.80):

$$p_{\vec{y}|\vec{\lambda}} = \int_{\mathrm{ADC}^{-1}(\vec{y})} \delta\left[\vec{\psi} - \mathcal{F}^N(\vec{\lambda})\right] d\vec{\psi} = \begin{cases} 1 & \text{si } \mathcal{F}^N(\vec{\lambda}) \in \mathrm{ADC}^{-1}(\vec{y}) \\ 0 & \text{en otro caso} \end{cases}$$

$$= \begin{cases} 1 & \text{si } \mathrm{ADC}\left[\mathcal{F}^N(\vec{\lambda})\right] = \vec{y} \\ 0 & \text{en otro caso} \end{cases}$$

$$= \delta_{\vec{y},\mathrm{ADC}\left[\mathcal{F}^N(\vec{\lambda})\right]} \tag{3.5}$$

donde ahora la cantidad "δ" representa la delta de Kronecker. Si utilizamos una topología $\mathcal{T}_N$ como codificador $\mathrm{ADC} = \mathcal{T}_N$, podemos escribir $\mathcal{T}_N(\vec{\lambda}) = \mathrm{ADC}[\mathcal{F}^N(\vec{\lambda})]$ y obtener finalmente:

$$p_{\vec{y}|\vec{\lambda}} = \delta_{\vec{y},\mathcal{T}_N(\vec{\lambda})} \tag{3.6}$$

Por otra parte, dado que el número de celdas físicamente realizables es finito, podemos considerar el subespacio $\mathbf{\Lambda}'$, $\lambda \in \mathbf{\Lambda}' \subset \mathbf{\Lambda}$ que contiene todos los posibles índices λ correspondientes a celdas que han sido o serán fabricados. Al contrario que el espacio $\mathbf{\Lambda}$, el subespacio $\mathbf{\Lambda}'$ será numerable, $|\mathbf{\Lambda}'| = M$, donde M es el número de celdas fabricables físicamente, que sin embargo podrá ser una cantidad arbitrariamente grande. Así, es posible asignar cada índice λ a un natural m, $\lambda(m)$, y sin pérdida de generalidad numerar cada celda $\mathcal{F}\left[\lambda(m)\right] \to \mathcal{F}(m)$ de manera ordenada atendiendo a la magnitud de su respuesta:

$$\mathcal{F}(m) < \mathcal{F}(m') \Longleftrightarrow 1 \leq m < m' \leq M \tag{3.7}$$

Bajo este precepto podemos sustituir todas las referencias previas al vector de celdas por su homólogo entero, $\vec{\lambda} \to \vec{m}$, y reescribir (2.81) para la probabilidad marginal de obtener una respuesta binaria $\vec{y}$:

$$p_{\vec{y}} = \sum_{\vec{m}} p_{\vec{y}|\vec{m}}\, p_{\vec{m}} = \sum_{\vec{m}} \delta_{\vec{y},\mathcal{T}_N(\vec{m})}\, p_{\vec{m}} \tag{3.8}$$

Suponiendo que el proceso de fabricación en tanto que proceso estocástico es no-sesgado en la producción de índices λ (o, equivalentemente, índices m), podemos asumir que la distribución de índices m es plana, $p_m = \mathrm{cte} = 1/\Omega_N$, donde Ω_N es el

número total de posibles N-tuplas (combinaciones ordenadas) diferentes a extraer del reservorio de M celdas:

$$\Omega_N = \binom{M}{N} N! = \frac{M!}{(M-N)!} = \prod_{i=1}^{N} (M-i-1) \\ = M^N + \mathcal{O}\left(M^{N-1}\right) \tag{3.9}$$

donde se ha utilizado la "O" grande de Landau para representar el comportamiento asintótico de Ω_N con M. Así, reescribimos (3.8) como

$$p_{\vec{y}} = p_{\vec{y}}\left(\mathcal{T}_N\right) = \frac{\sum_{\vec{m}} \delta_{\vec{y},\mathcal{T}_N(\vec{m})}}{\Omega_N} \tag{3.10}$$

Si definimos la clase de equivalencia de todas las instancias (*i.e.*, N-tuplas $\vec{m}$) tales que codifican para un mismo vector binario $\vec{y}$ dada una topología $\mathcal{T}_N$,

$$\mathbf{\Lambda}'/\vec{y} \equiv \{\vec{m} | \mathcal{T}_N\left(\vec{m}\right) = \vec{y}\} \subset \mathbf{\Lambda} \tag{3.11}$$

entonces tenemos para el numerador de (3.10)

$$\sum_{\vec{m}} \delta_{\vec{y},\mathcal{T}_N(\vec{m})} = \left|\mathbf{\Lambda}'/\vec{y}\right| \equiv \omega_{\vec{y}}\left(\mathcal{T}_N\right) \tag{3.12}$$

y finalmente escribimos la distribución de probabilidad (3.10) para una topología $\mathcal{T}_N$:

$$p_{\vec{y}}\left(\mathcal{T}_N\right) = \frac{\omega_{\vec{y}}\left(\mathcal{T}_N\right)}{\Omega_N} \tag{3.13}$$

que es la distribución de probabilidad asociada a la variable aleatoria $\mathsf{Y}_{\mathcal{T}_N}$ inducida por la topología $\mathcal{T}_N$ (3.2). Además, esta variable aleatoria tendrá asociada una entropía:

$$H\left(\mathsf{Y}_{\mathcal{T}_N}\right) = -\sum_{\vec{y}} p_{\vec{y}}\left(\mathcal{T}_N\right) \log_2\left[p_{\vec{y}}\left(\mathcal{T}_N\right)\right] \equiv H\left(\mathcal{T}_N\right) \tag{3.14}$$

que se define como la entropía de la topología $\mathcal{T}_N$, y que caracteriza las propiedades criptográficas de este sistema.

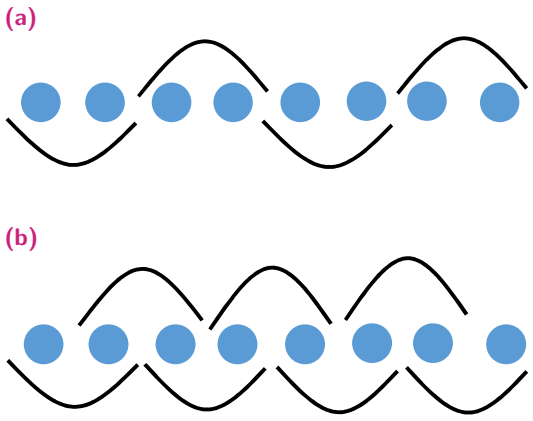


Fig. 3.4.: Esquemas de comparación: (a) $\mathcal{N}_{/2}$ y (b) $\mathcal{N}_{-1}$.

3.3. Topologías en el diseño de PUF de medida compensada

En la práctica del diseño de funciones no-clonables físicamente de medida compensada, existen dos topologías utilizadas de forma predominante [29]:

1. El enmascaramiento *1-out-of-k*, propuesto por Suh y Devadas en [118] como una manera de obtener bits descorrelacionados e incrementar la resistencia del sistema frente a variaciones ambientales. En esta aproximación, el conjunto de N celdas se divide en N/k grupos de k celdas cada uno. La función de codificación extrae un bit de cada uno de estos grupos, obtenido mediante la comparación del par de celdas tales que la diferencia de sus magnitudes características es máxima, lo cual reduce la probabilidad de que las fluctuaciones ambientales afecten al bit. Para el análisis llevado a cabo en este capítulo nos atendremos a la forma más simple de enmascaramiento utilizando $k = 2$; denotaremos esta topología como "$\mathcal{N}_{/2}$ ", la cual se caracteriza porque para cada colección de N celdas proporcionará una respuesta de $N/2$ bits utilizando cada celda sin repetición, tal y como se muestra en el esquema representado en la figura 3.4a.

2. La comparación de celdas sucesivas con repetición, propuesto por Maiti *et al.* [127], es un esquema de digitalización en el que cada bit se extrae de la comparación entre celdas vecinas (figura 3.4b), de tal modo que para una

matriz de N celdas se obtiene una palabra binaria de $N_b = N - 1$ bits. Nos referiremos simbólicamente a esta topología en la que comparamos todas las celdas inmediatamente contiguas repitiendo una cada vez como "$\mathcal{N}_{-1}$ ".

Además de estas arquitecturas, en subsiguientes secciones de este capítulo estudiaremos también la topología trivial, cuyas respuestas se construyen comparando todas las parejas posibles dada una matriz de N celdas y que denotaremos con el símbolo "$\mathcal{N}^2$ ". Una vez expuestos algunos defectos hallados en estas estructuras, propondremos una novedosa familia de esquemas de comparación "K-modular" denotada como "$\mathcal{N}^2_{/K}$ "y que solventan algunas de las deficiencias encontradas en los esquemas de comparación mencionados.

La metodología de estudio seguida en esta sección es el análisis teórico de las distribuciones de probabilidad que emergen de cada topología dada, así como resultados de simulación y medidas experimentales. La variable aleatoria Λ que modela el proceso de fabricación ha sido simulada mediante una distribución plana en el espacio $\mathbf{\Lambda} = (0, 1) \subset \mathbb{R}$. Por otra parte, el comportamiento físico de cada celda se simula siguiendo el modelo definido por la expresión (3.3), lo cual permite asignar unívocamente una cantidad a cada celda $\psi_i = \psi_i(\lambda_i)$. Dado que el proceso de digitalización depende únicamente de la comparación entre magnitudes, podemos escalar linealmente las respuestas de cada celda sin alterar el comportamiento de la PUF, *i.e.*, dado un par de respuestas físicas ψ_0, ψ_1, se tiene:

$$\psi_0 \gtrless \psi_1 \iff a\psi_0 + b \gtrless a\psi_1 + b \tag{3.15}$$

con a, b constantes mayores que cero. Llamando $\psi_{\text{min}} \equiv \text{mín}\,(\psi)$, $\psi_{\text{max}} \equiv \text{máx}\,(\psi)$ podemos normalizar las respuestas de cada celda $\psi' \equiv a\psi + b$ utilizando las constantes $a = 1/(\psi_{\text{max}} - \psi_{\text{min}})$, $b = a\psi_{\text{min}}$; estas cantidades normalizadas se encuentran por construcción en el intervalo $(0, 1)$, y bajo la hipótesis de que en efecto estas celdas son identificables y físicamente no-clonables podemos identificar la i-ésima celda de índice λ_i con su respuesta normalizada, $\lambda_i \equiv \psi'_i$, o equivalentemente $\psi'_i = \lambda_i$. Así, la medición de una matriz de N celdas es simulada como una serie de N números aleatorios tomados en el intervalo $(0, 1)$. En cuanto a los datos experimentales, estos están formados por una batería de frecuencias de oscilación medidas sobre varias series de 200 osciladores de anillo, implementados en 40 FPGA Artix 7 diferentes, de tal forma que se utilizan los grupos de N anillos más próximos en la construcción de cada respuesta binaria. En el capítulo 4 se aborda en detalle la implementación de estas estructuras en FPGA, así como sus propiedades en el diseño PUF.

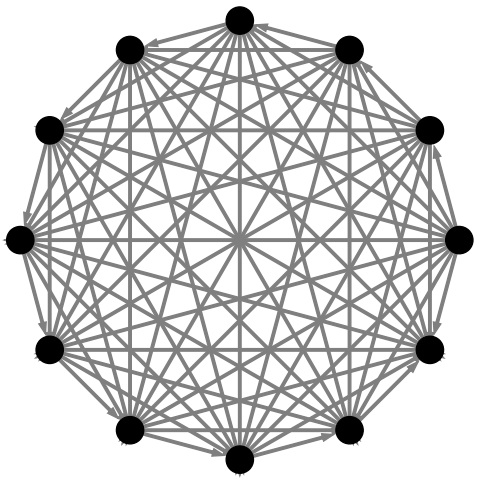


Fig. 3.5.: Digrafo típico representante de la topología $\mathcal{N}^2$ con $N = 12$.

3.3.1. Todas las comparaciones posibles: topología $\mathcal{N}^2$

En este caso el proceso de digitalización se lleva a cabo agotando todas las posibles comparaciones en la matriz de celdas, dando lugar a una palabra binaria $\vec{y}$ de $N_b = \binom{N}{2} = N(N-1)/2$ bits, tal y como se muestra esquemáticamente en el grafo de la figura 3.5 asociado a la topología $\mathcal{N}^2$ con $N = 12$. Tal y como se discutió en la sección 2.3.5, este método de digitalización presentará una serie de restricciones a los posibles valores de algunos bits de la respuesta debido a la propiedad transitiva de la comparación, que podemos formular como:

$$\begin{cases} \psi_i > \psi_j, \psi_i < \psi_k \Longrightarrow \psi_j < \psi_k \\ \psi_i < \psi_j, \psi_i > \psi_k \Longrightarrow \psi_j > \psi_k \end{cases} \tag{3.16}$$

de forma que resulta evidente que, siempre que la comparación de dos celdas (j, k) con una tercera (i) devuelva bits diferentes, entonces el bit de la comparación entre las celdas (j, k) estará predeterminado. Podemos escribir esto en términos de la función diferencial bit definida en (2.115) como:

$$\mathrm{bit}\,(i,j) \neq \mathrm{bit}\,(i,k) \Longrightarrow \mathrm{bit}\,(j,k) = \mathrm{bit}\,(i,k) \tag{3.17}$$

de tal forma que una respuesta binaria de $N(N-1)/2$ bits incompatible con esta restricción tendrá una probabilidad asociada nula en un esquema $\mathcal{N}^2$, constituyendo un espacio de estados inaccesible para este sistema.

Sea $\vec{y}$ una palabra binaria de $N(N-1)/2$ bits compatible con (3.17), podemos calcular explícitamente la probabilidad $p_{\vec{y}}\left(\mathcal{N}^2\right) \equiv \mathrm{Prob}(\mathsf{Y}_{\mathcal{N}^2} = \vec{y})$ de que sea generada por una topología $\mathcal{N}^2$ aplicando el modelo de fabricación descrito en la sección anterior. Ya que sólo es físicamente relevante la diferencia entre las respuestas de cada pareja de celdas (o, equivalentemente, el orden que ocupa cada celda de la matriz $\mathcal{F}^N$ en relación a las demás), dada una construcción de N celdas extraídas del reservorio de M posibles habrá toda una familia de N-tuplas que serán equivalentes en el sentido de que darán lugar a una matriz $\mathcal{F}^N$ cuyo patrón de comparación por pares será idéntico. Denotamos la i-ésima celda de la N-configuración como $m_i^{[k(i)]}$ donde m es un valor entero que identifica la celda del total de M celdas de forma ordenada tal y como se define en (3.7), i indica la posición de la celda en el vector, y $k(i)$ indica la posición que ocupa la i-ésima celda en el ranking de magnitud, siendo "$k=1$" la celda de menor magnitud, y "$k=N$" la de mayor; notar que $k(i)$ es una permutación de $1, \cdots, N$. Dadas las dos celdas de menor magnitud $m^{[1]}$, $m^{[2]}$, la PUF construida aplicando la topología $\mathcal{N}^2$ a la función física $\mathcal{F}^N$ no se vería modificada si el proceso de fabricación hubiera arrojado cualquier otra celda $m^{[1]\prime}$ de magnitud menor a $m^{[2]}$, $1 \leq m^{[1]\prime} < m^{[2]}$. Así mismo, la PUF tampoco se hubiera visto modificada si en lugar de $m^{[2]}$ el proceso de fabricación hubiera arrojado una segunda celda $m^{[2]\prime}$ en el intervalo $2 \leq m^{[2]\prime} < m^{[3]}$ (notar que $m^{[2]\prime} = 1$ es una contradicción, ya que $m=1$ es por definición la menor de todas las celdas fabricables, y en cambio $m^{[2]}$ es sólo la segunda celda más pequeña del vector, *i.e.*, debe haber una celda aún menor, a saber, "$m^{[1]}$"). Este argumento puede repetirse para todas las celdas del vector hasta llegar a la superior $m^{[N]}$, que daría lugar a una PUF $\mathcal{N}^2$ equivalente para cualquier $N \leq m^{[N]\prime} \leq M$ arrojada por el proceso de fabricación. De esta manera, el número de N-tuplas equivalentes $\omega_{\vec{y}}\left(\mathcal{N}^2\right)$ (3.12) puede escribirse como la suma de los casos equivalentes para cada $m^{[i]}$:

$$\omega_{\vec{y}}\left(\mathcal{T}_N = \mathcal{N}^2\right) = \omega\left(\mathcal{N}^2\right) = \sum_{m^{[N]}=N}^{M} \cdots \sum_{m^{[2]}=2}^{m^{[3]}-1} \sum_{m^{[1]}=1}^{m^{[2]}-1} 1 \tag{3.18}$$

donde se ha suprimido la dependencia explícita con el vector de respuesta $\vec{y}$ porque la suma en (3.18) no depende de esta cantidad (notar que el argumento anterior se reproduce de forma idéntica para cualquier vector de respuesta binario). Finalmente, para la distribución de probabilidad de las respuestas en la topología $\mathcal{N}^2$ (3.10) se tiene:

$$p_{\vec{y}}\left(\mathcal{T}_N = \mathcal{N}^2\right) = p\left(\mathcal{N}^2\right) = \frac{\omega\left(\mathcal{N}^2\right)}{\Omega_N} \tag{3.19}$$

con Ω_N dado por (3.9). Definiendo la densidad $\Delta\rho \equiv 1/M$, podemos escribir en general la siguiente suma como una integral de Riemann en el límite $M \ggg N_1, N_2$ (*i.e.*, $N_{1,2}/M \to 0$):

$$\begin{aligned}
\sum_{m=N_1}^{M-N_2} a_m &= \sum_{m=1}^{M} a_m - \sum_{m=1}^{N_1} a_m - \sum_{m=M-N_2}^{M} a_m \\
&= M \left(\sum_{m=1}^{M} a_m \, \Delta\rho - \sum_{m=1}^{N_1} a_m \, \Delta\rho - \sum_{m=M-N_2}^{M} a_m \, \Delta\rho \right) \\
&\xrightarrow[M\to\infty]{} M \left(\int_0^1 f(\rho)\, d\rho - \int_0^{0+\cancel{N_1/M}} f(\rho)\, d\rho - \int_{1-\cancel{N_2/M}}^1 f(\rho)\, d\rho \right) \\
&= M \int_0^1 f(\rho)\, d\rho
\end{aligned} \tag{3.20}$$

donde $f\left(\rho = \frac{m-1}{M-1}\right) \equiv a_m$. En el límite $M \gg N$ podemos escribir Ω_N como su asíntota en M:

$$\begin{aligned}
\Omega_N &= M^N + \mathcal{O}\left(M^{N-1}\right) \\
&\xrightarrow[M\to\infty]{} M^N
\end{aligned} \tag{3.21}$$

Y por (3.20) escribimos la sucesión de sumas (3.18) como:

$$\omega\left(\mathcal{N}^2\right) \xrightarrow[M\to\infty]{} M^N \int_0^1 d\rho_N \cdots \int_0^{\rho_3} d\rho_2 \int_0^{\rho_2} d\rho_1 \tag{3.22}$$

Por (3.21) y (3.22) podemos escribir la probabilidad (3.19) como:

$$\begin{aligned}
p\left(\mathcal{N}^2\right) &\xrightarrow[M\to\infty]{} \frac{M^N \int_0^1 d\rho_N \cdots \int_0^{\rho_3} d\rho_2 \int_0^{\rho_2} d\rho_1}{M^N} \\
&= \int_0^1 d\rho_N \cdots \int_0^{\rho_3} d\rho_2 \int_0^{\rho_2} d\rho_1
\end{aligned} \tag{3.23}$$

Notar que esta expresión no depende de M y, por lo tanto, es perfectamente general. La sucesión de integrales encadenadas en (3.23) puede calcularse iterativamente:

$$\int_0^{\rho_2} d\rho_1 \frac{1}{1} = \frac{1}{1}\rho_2$$

$$\int_0^{\rho_3} d\rho_2 \, \frac{1}{1}\rho_2 = \frac{1}{2} \times \frac{1}{1}\rho_3^2$$
$$\int_0^{\rho_4} d\rho_3 \, \frac{1}{2\times 1}\rho_3^2 = \frac{1}{3} \times \frac{1}{2\times 1}\rho_4^3$$
$$\vdots$$
$$\int_0^1 d\rho_N \, \frac{1}{(N-1)\times(N-2)\times\cdots\times 1}\rho_N^{N-1} = \frac{1}{N} \times \frac{1}{(N-1)\times(N-2)\times\cdots\times 1} = \frac{1}{N!}$$

Obteniendo finalmente para la probabilidad de una PUF de medida compensada de N celdas codificada mediante una topología $\mathcal{N}^2$:

$$p\left(\mathcal{N}^2\right) = \frac{1}{N!} \tag{3.24}$$

y para su entropía y minentropía asociadas (notar que ambas coinciden al ser la distribución plana):

$$H\left(\mathcal{N}^2\right) = H^{\mathrm{m}}\left(\mathcal{N}^2\right) = \log_2 N! \tag{3.25}$$

En la figura 3.5 se ha representado a modo de ejemplo un grafo típico de la topología $\mathcal{N}^2$ con $N = 12$. Este grafo contiene todas las aristas posibles en un sistema de N vértices, cualquier otra topología será de hecho un módulo de $\mathcal{N}^2$, lo cual permite escribir la variable aleatoria correspondiente a esta topología como $\mathsf{Y}_{\mathcal{N}^2} = (\mathsf{Y}_{\mathcal{T}_N}, \Delta)\ \forall \mathcal{T}_N$. Así, se tendrá para la entropía:

$$H\left(\mathcal{N}^2\right) = H\left(\mathcal{T}_N, \Delta\right) \geq H\left(\mathcal{T}_N\right)\ \forall \mathcal{T}_N \tag{3.26}$$

donde la desigualdad se deduce aplicando (2.42) y (2.44). Este resultado prueba que la entropía máxima extraíble en un sistema PUF de medida compensada corresponde a la topología completa $\mathcal{N}^2$. En efecto, cabe notar que la entropía calculada analíticamente en (3.25) coincide con el resultado hallado heurísticamente en (2.116) para la entropía total de una función no-clonable físicamente de medida compensada.

En la figura 3.6 se han representado los histogramas obtenidos para las distribuciones de respuestas binarias codificadas con la topología $\mathcal{N}^2$ en matrices de $N = 3$ (3.6a) y $N = 4$ (3.6b) celdas, tanto por simulación como experimentalmente. Sobre estas imágenes se ha superpuesto la curva de interpolación teórica construida aplicando la distribución de probabilidad (3.24) a las palabras binarias de $N(N-1)/2$ bits compatibles con la restricción (3.17). Así mismo, en la figura 3.7 se muestra la entropía obtenida mediante simulación de la topología $\mathcal{N}^2$ en función del tamaño N de la matriz de celdas (puntos azules), interpolada por la curva obtenida en (3.26) (línea discontinua naranja), mostrando un acuerdo excelente.

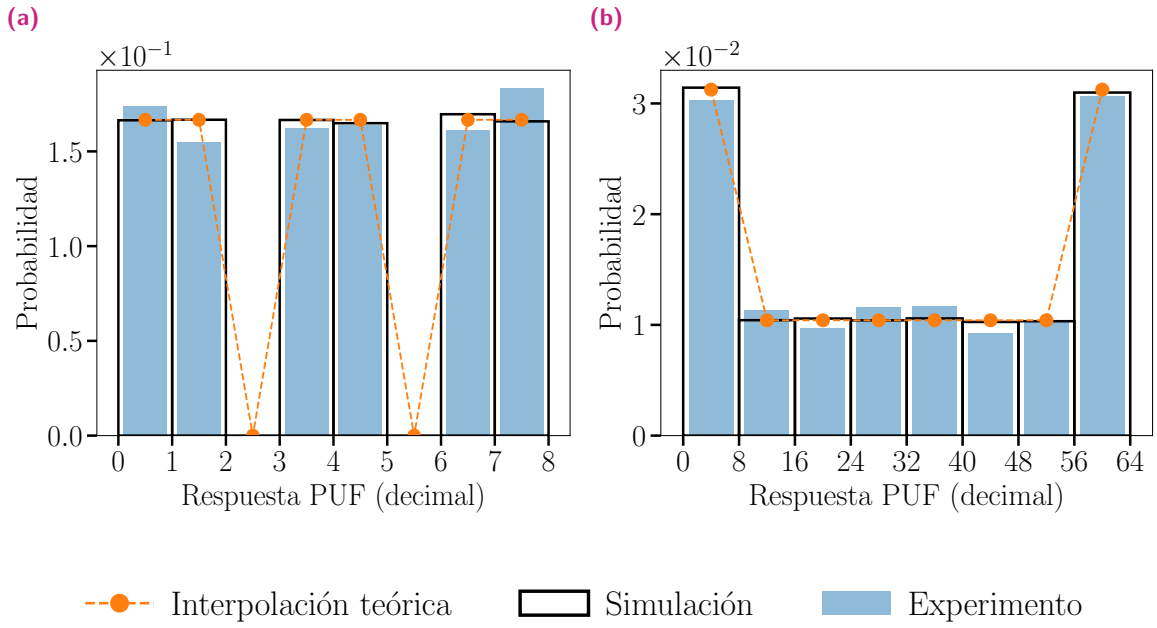


Fig. 3.6.: Histogramas de las respuestas para la topología $\mathcal{N}^2$ con $N = 3$ celdas (a) y $N = 4$ celdas (b), obtenidos mediante simulación y medición experimental, superpuestos sobre la distribución teórica esperada.

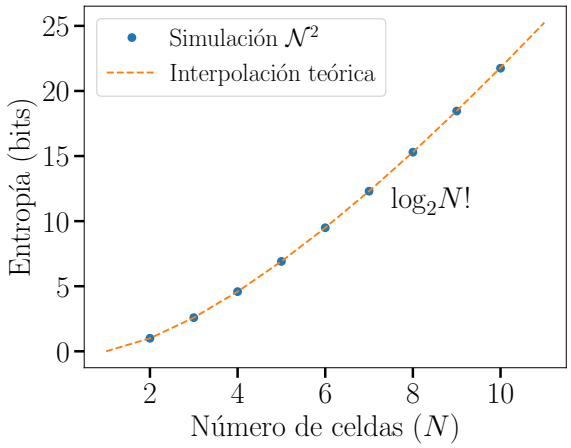


Fig. 3.7.: Entropía entregada por la topología $\mathcal{N}^2$ en función del número N de celdas físicas del sistema, junto con la curva de interpolación teórica.

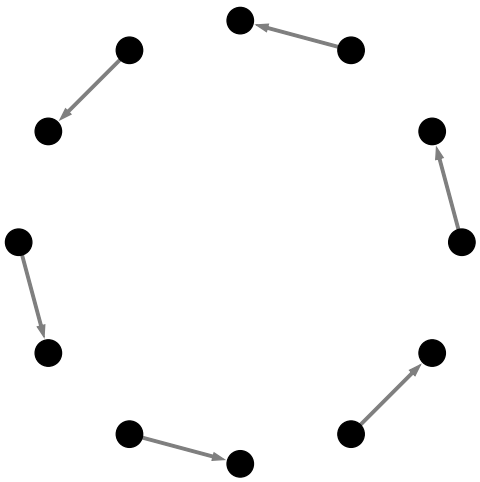


Fig. 3.8.: Grafo típico representante de la topología $\mathcal{N}_{/2}$ con $N = 12$.

3.3.2. Comparaciones sin repetición: topología $\mathcal{N}_{/2}$

En esta topología, un vector de N respuestas físicas $\vec{\psi} = (\psi_1, \cdots, \psi_N)$, N par, es evaluado y codificado en una respuesta binaria $\vec{y}_{\mathcal{N}_{/2}} = (y_1, \cdots, y_{N/2})$ de $N_b = N/2$ bits mediante la comparación de parejas de celdas adyacentes sin repetición:

$$y_i \equiv \text{bit}(2i-1, 2i) = \begin{cases} 1 & \text{si } \psi_{2i-1} - \psi_{2i} > 0 \\ 0 & \text{en otro caso} \end{cases} \tag{3.27}$$

La principal y más evidente consecuencia de esta definición es la inaplicabilidad de la restricción (3.17), dado que ninguna celda se reutiliza para producir un bit. Así, todas las $2^{N/2}$ posibles respuestas serán matemáticamente accesibles, y la correlación entre sus elementos será presumiblemente nula.

Este esquema de comparación, ilustrado en el grafo de la figura 3.8 perteneciente a la topología $(\mathcal{N} = 12)_{/2}$, es, junto a otros que generalizan el enmascaramiento *n-out-of-k*, el método de digitalización más habitual en la práctica del diseño PUF de medida compensada. Esta arquitectura produce respuestas binarias de $N/2$ bits, comprometiendo un gran coste de entropía en aras de respuestas más robustas frente a variaciones ambientales. Desafortunadamente, dado que la actividad de interrupción en un diseño construido sobre tecnología CMOS se puede correlacionar con el número de celdas en la matriz [128], este sistema presentará un mal compromiso entre seguridad y eficiencia energética, lo cual perjudica la integración de esta clase de soluciones en tecnología IoT. No obstante, la ausencia absoluta de reutilización de celdas se traduce en una nula correlación entre bits, lo cual redunda

en una distribución de probabilidad uniforme y en una tasa de entropía por bit H/N_b cercana al 100 % ideal, indicativo de una buena resistencia al criptoanálisis.

Para estudiar la distribución de probabilidad de la topología $\mathcal{N}_{/2}$ aplicamos el mismo modelo de fabricación descrito en la sección 3.2. Ahora, el número de posibles formas de tomar un par de celdas consecutivas (m_{2i-1}, m_{2i}) tales que el i-ésimo bit —definido en (3.27)— no cambia, $\omega_{y_i}(\mathcal{T}_N = 2_{/2})$, estará dado por:

$$\omega_{y_i}(2_{/2}) = \begin{cases} \sum_{m_{2i}=1}^{M} \sum_{m_{2i-1}=1}^{m_{2i}} (1 - \delta_{m_{2i-1},m_{2i}}) & \text{si } y_i = 0 \\ \\ \sum_{m_{2i}=1}^{M} \sum_{m_{2i-1}=m_{2i}}^{M} (1 - \delta_{m_{2i-1},m_{2i}}) & \text{si } y_i = 1 \end{cases} \tag{3.28}$$

donde la delta de Kronecker, $\delta_{m_{2i},m_{2i-1}}$, sustrae aquellos casos imposibles (esto es, $m_{2i} = m_{2i-1}$) que se han contado en la sumas, y el índice natural m_j sustituye al índice continuo λ_j como identificador de cada celda en una matriz ya que únicamente contamos las celdas físicamente fabricables —que son numerables— tal y como se argumentó en la sección 3.2; dado que la variable y_i es una cantidad binaria que sólo puede tomar los valores 0 o 1, esta expresión puede escribirse de forma más compacta como:

$$\omega_{y_i}(2_{/2}) = \sum_{m_{2i-1}=1+y_i(m_{2i}-1)}^{m_{2i}+y_i(M-m_{2i})} (1 - \delta_{m_{2i},m_{2i-1}}) \tag{3.29}$$

Y esta se amplía a un vector de $N/2$ bits $\vec{y} = (y_1, \cdots, y_{N/2})$ producidos al aplicar la topología $\mathcal{N}_{/2}$ sobre una matriz de N celdas como:

$$\begin{aligned} \omega_{\vec{y}}(\mathcal{N}_{/2}) = \sum_{m_N=1}^{M} \sum_{m_{N-1}=1+b_{N/2}(m_N-1)}^{m_N+b_{N/2}(M-m_N)} \cdots \\ \sum_{m_2=1}^{M} \sum_{m_1=1+b_1(m_2-1)}^{m_2+b_1(M-m_2)} (1 - \delta_{m_N,m_{N-1},\cdots,m_2,m_1}) \end{aligned} \tag{3.30}$$

donde la cantidad $\delta_{m_N,\cdots,m_1}$ es la extensión de la delta de Kronecker dada por:

$$\delta_{m_N,\cdots,m_1} \equiv \begin{cases} 1 \text{ si } \exists\, i \neq j \,|\, m_i = m_j \\ 0 \text{ en otro caso} \end{cases} \tag{3.31}$$

que evita sumar aquellas combinaciones en las que cualesquiera dos índices m_j, m_k, $j \neq k$ coinciden. Por la linealidad del sumatorio, (3.30) puede escribirse como la diferencia entre la suma sobre la unidad y la suma sobre la delta de Kronecker

extendida; dado que la cantidad δ es definida positiva, este segundo término estará acotado superiormente por:

$$\sum_{m_N=1}^{M} \cdots \sum_{m_1=1+y_1(m_2-1)}^{m_2+y_1(M-m_2)} \delta_{m_N,\cdots,m_1} \leq \sum_{m_N=1}^{M} \cdots \sum_{m_1=1}^{M} \delta_{m_N,\cdots,m_1} \tag{3.32}$$

ya que para los extremos de los sumatorios se tiene $M \geq m + y(M-m)$, y $1 \leq 1 + y(m-1)$. Utilizando que la cantidad $\delta_{m_N,\cdots,m_1}$ puede interpretarse como una variable booleana, separamos esta utilizando el operador OR lógico, $\vee$:

$$\begin{aligned} \delta_{m_N,\cdots,m_1} &= \delta_{m_N,\cdots,m_2} \vee \delta_{m_2,m_1} \vee \delta_{m_3,m_1} \cdots \vee \delta_{m_N,m_1} \\ &= \delta_{m_N,\cdots,m_2} \vee \bigvee_{i=2}^{N} \delta_{m_i,m_1} \end{aligned} \tag{3.33}$$

y utilizando la identidad $a \vee b = a + b - ab$, reescribimos (3.33) como:

$$\delta_{m_N,\cdots,m_1} = \delta_{m_N,\cdots,m_2} + (1 - \delta_{m_N,\cdots,m_2}) \bigvee_{i=2}^{N} \delta_{m_i,m_1} \tag{3.34}$$

A continuación, utilizamos esta expresión para calcular explícitamente el primer término de la suma (3.32):

$$\begin{aligned} \sum_{m_1=1}^{M} \delta_{m_N,\cdots,m_1} &= \sum_{m_1=1}^{M} \left[\delta_{m_N,\cdots,m_2} + (1 - \delta_{m_N,\cdots,m_2}) \bigvee_{i=2}^{N} \delta_{m_i,m_1} \right] \\ &= \delta_{m_N,\cdots,m_2} \sum_{m_1=1}^{M} 1 + (1 - \delta_{m_N,\cdots,m_2}) \sum_{m_1=1}^{M} \bigvee_{i=2}^{N} \delta_{m_i,m_1} \\ &= M\delta_{m_N,\cdots,m_2} + (1 - \delta_{m_N,\cdots,m_2}) \sum_{m_1=1}^{M} \bigvee_{i=2}^{N} \delta_{m_i,m_1} \end{aligned} \tag{3.35}$$

Dada la definición de la delta extendida (3.31), el coeficiente del segundo término en (3.35) únicamente será distinto de cero cuando todos los índices $m_N, \cdots, m_2$ sean distintos entre sí, en cuyo caso valdrá la unidad. Bajo este supuesto, el sumando $\bigvee_{i=2}^{N} \delta_{m_i,m_1}$ valdrá la unidad cada vez que el valor del índice m_1 coincida con el valor de cada uno de los $N-1$ índices $\{m_i\}_{i=2}^{N}$, de modo que la suma $\sum \bigvee \delta_{m_i,m_1} = N-1$. Introduciendo este resultado en (3.35) escribimos:

$$\begin{aligned} \sum_{m_1=1}^{M} \delta_{m_N,\cdots,m_1} &= M\delta_{m_N,\cdots,m_2} + (1 - \delta_{m_N,\cdots,m_2})\,(N-1) \\ &= (M - N + 1)\delta_{m_N,\cdots,m_2} + N - 1 \\ &= \mathrm{pol}\,(M)\,\delta_{m_N,\cdots,m_2} + N - 1 \end{aligned} \tag{3.36}$$

donde $\mathrm{pol}(M)$ representa un polinomio de primer orden en M; así mismo, introduciendo este resultado en la suma sobre el índice m_2 en (3.32):

$$\begin{aligned}\sum_{m_2=1}^{M}\left[\mathrm{pol}(M)\,\delta_{m_N,\cdots,m_2}+N-1\right] &= \mathrm{pol}(M)\,\left[\mathrm{pol}(M)\,\delta_{m_N,\cdots,m_3}+N-1\right]+M(N-1)\\ &= \mathrm{pol}(M^2)\,\delta_{m_N,\cdots,m_3}+\left[\mathrm{pol}(M)+M\right](N-1)\\ &= \mathrm{pol}(M^2)\,\delta_{m_N,\cdots,m_3}+\mathrm{pol}(M)(N-1) \qquad (3.37)\end{aligned}$$

Es sencillo probar por inducción que, en general, la suma encadenada sobre el k-ésimo índice será:

$$\sum_{m_k=1}^{M}\ldots\,\delta_{m_N,\cdots,m_1}=\mathrm{pol}(M^k)\,\delta_{m_N,\cdots,m_{k+1}}+\mathrm{pol}(M^{k-1})(N-1) \qquad (3.38)$$

Aplicando esta expresión a $k=N-2$ se tiene:

$$\sum_{m_{N-2}=1}^{M}\ldots\,\delta_{m_N,\cdots,m_1}=\mathrm{pol}(M^{N-2})\,\delta_{m_N,m_{N-1}}+\mathrm{pol}(M^{N-3})(N-1) \qquad (3.39)$$

De forma que la suma total:

$$\begin{aligned}&\sum_{m_N=1}^{M}\sum_{m_{N-1}=1}^{M}\left(\sum_{m_{N-2}=1}^{M}\ldots\delta_{m_N,\cdots,m_1}\right)\\ &=\sum_{m_N=1}^{M}\sum_{m_{N-1}=1}^{M}\mathrm{pol}(M^{N-2})\,\delta_{m_N,m_{N-1}}+\mathrm{pol}(M^{N-3})(N-1)\\ &=\mathrm{pol}(M^{N-2})\,M+\mathrm{pol}(M^{N-3})\,M^2(N-1)\\ &=\mathrm{pol}(M^{N-1})\in\mathcal{O}\left(M^{N-1}\right) \qquad (3.40)\end{aligned}$$

Y aplicando este resultado a (3.32):

$$\sum_{m_N=1}^{M}\cdots\sum_{m_1=1+y_1(m_2-1)}^{m_2+y_1(M-m_2)}\delta_{m_N,\cdots,m_1}\in\mathcal{O}\left(M^{N-1}\right) \qquad (3.41)$$

De este modo, podemos escribir (3.30) como:

$$\omega_{\vec{y}}(\mathcal{N}_{/2})=\sum_{m_N=1}^{M}\cdots\sum_{m_1=1+y_1(m_2-1)}^{m_2+y_1(M-m_2)}1+\mathcal{O}\left(M^{N-1}\right) \qquad (3.42)$$

Por otra parte, en el límite $M \gg 1$ podemos sustituir las sumas del término de la izquierda en (3.42) por las integrales de Riemann, utilizando la noción de densidad "ρ" dada en (3.20):

$$\omega_{\vec{y}}(\mathcal{N}_{/2}\,) \xrightarrow[M\to\infty]{} M^N \int_0^1 d\rho_N \int_{\rho_N y_{N/2}}^{\rho_N+(1-\rho_N)y_{N/2}} d\rho_{N-1}\ldots \int_0^1 d\rho_2 \int_{\rho_2 y_1}^{\rho_2+(1-\rho_2)y_1} d\rho_1 + \mathcal{O}\left(M^{N-1}\right) \tag{3.43}$$

Y aplicando ahora (3.19) y (3.21) en el límite continuo:

$$p_{\vec{y}}(\mathcal{N}_{/2}\,) = \int_0^1 d\rho_N \int_{\rho_N y_{N/2}}^{\rho_N+(1-\rho_N)y_{N/2}} d\rho_{N-1}\ldots \int_0^1 d\rho_2 \int_{\rho_2 y_1}^{\rho_2+(1-\rho_2)y_1} d\rho_1 \tag{3.44}$$

donde se ha utilizado $\lim_{M\to\infty} \mathcal{O}\left(M^{N-1}\right)/M^N = 0$. Cada par de integrales encadenadas $\int d\rho_{2i} \int d\rho_{2i-1}$ correspondientes al bit y_i es independiente del resto, de modo que podemos calcular explícitamente:

$$\int_0^1 d\rho_{2i} \int_{\rho_{2i} y_i}^{\rho_{2i}+(1-\rho_{2i})y_i} d\rho_{2i-1} = \frac{1}{2} \tag{3.45}$$

Y sustituyendo este resultado en (3.44):

$$p_{\vec{y}}(\mathcal{N}_{/2}\,) = p(\mathcal{N}_{/2}\,) = \frac{1}{2^{N/2}} \tag{3.46}$$

Este resultado demuestra la uniformidad en la distribución de probabilidad de las respuestas, confirmando la conjetura de una nula correlación entre bits debido a la ausencia de reutilización de celdas en la topología $\mathcal{N}_{/2}$. Utilizando (3.46) podemos calcular la entropía y minentropía, que coincidirán dada la uniformidad de la distribución:

$$H\left(\mathcal{N}_{/2}\right) = \mathrm{H}^{\mathrm{m}}\left(\mathcal{N}_{/2}\right) = -\log_2 \frac{1}{2^{N/2}} = N/2 \text{ bits} \tag{3.47}$$

En las figuras 3.9a y 3.9b se representan los histogramas correspondientes a la topología $\mathcal{N}_{/2}$ para matrices de $N = 6$ y $N = 12$ celdas respectivamente, obtenidos mediante simulación así como medidos experimentalmente. Así mismo, superpuesta al histograma se representa la distribución teórica hallada en (3.46). Por otra parte, en la figura 3.10 hemos representado la entropía para matrices simuladas

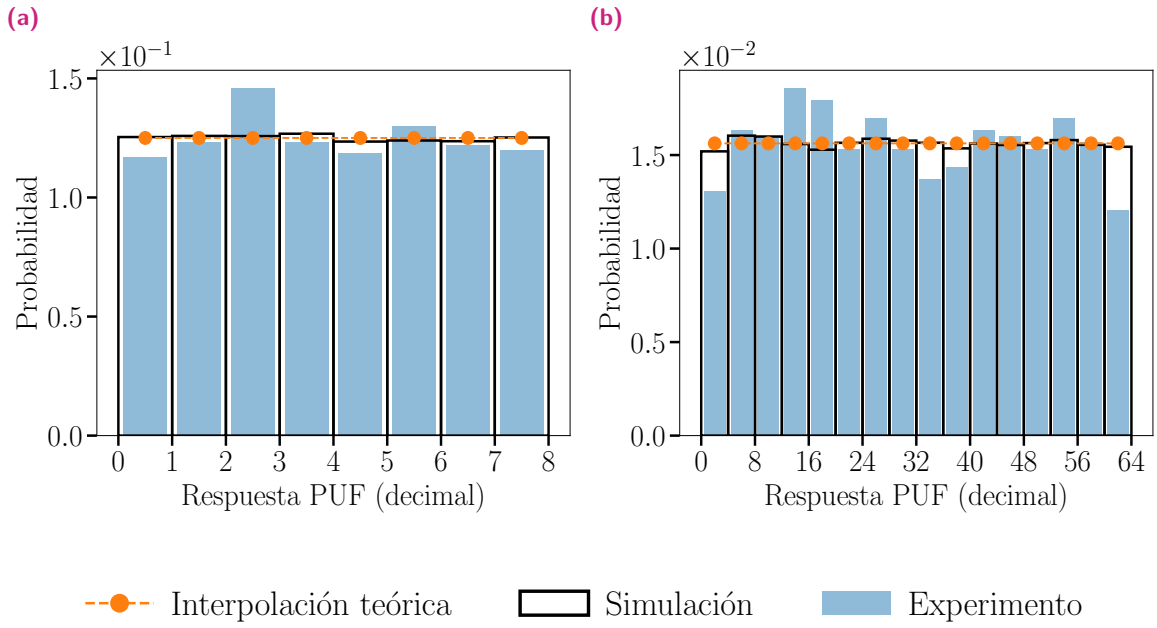


Fig. 3.9.: Histogramas de las respuestas para la topología $\mathcal{N}_{/2}$ con $N = 6$ celdas (a) y $N = 12$ celdas (b), obtenidos mediante simulación y medición experimental, superpuestos sobre la distribución teórica esperada.

de diferentes tamaños, cuyas respuestas han sido codificadas utilizando la topología $\mathcal{N}_{/2}$; superpuesta se muestra la curva de interpolación dada en (3.47).

La propiedad de que la entropía asociada a una topología $\mathcal{T}$ escale linealmente con el tamaño de la matriz de celdas N, $H(\mathcal{T}_N) \sim N$, es una característica general de las topologías modulares en las que el número de bits crece de forma lineal con el tamaño de la matriz, $N_b(N) \sim N$. En efecto, de acuerdo con la definición de modularidad dada en la sección 3.1, podemos escribir un sistema modular de $N+M$ celdas como un sistema conjunto de las variables $\mathsf{Y}_{\mathcal{T}_N}$, $\mathsf{Y}_{\mathcal{T}_M}$, Δ:

$$\begin{aligned} H\left(\mathcal{T}_{N+M}\right) &= H(\Delta, \mathcal{T}_N, \mathcal{T}_M) \\ &= H(\mathcal{T}_N|\mathcal{T}_M) + H(\mathcal{T}_M) + H(\Delta|\mathcal{T}_N, \mathcal{T}_M) \\ &= H(\mathcal{T}_N) + H(\mathcal{T}_M) + H(\Delta|\mathcal{T}_N, \mathcal{T}_M) \\ &\geq H(\mathcal{T}_N) + H(\mathcal{T}_M) \end{aligned} \tag{3.48}$$

donde se ha utilizado la regla de la cadena, *i.e.*, la aplicación sucesiva de (2.44), para expandir la entropía del sistema conjunto, así como el hecho de que las variables $\mathsf{Y}_{\mathcal{T}_N}$, $\mathsf{Y}_{\mathcal{T}_M}$ son independientes. Utilizando esta expresión, que es formalmente similar a la

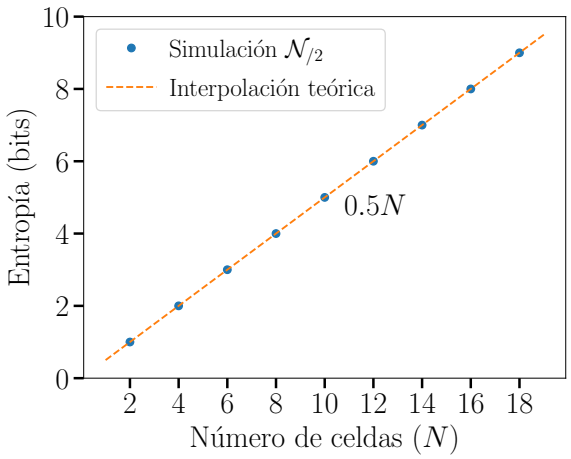


Fig. 3.10.: Entropía entregada por la topología $\mathcal{N}_{/2}$ en función del número N de celdas físicas del sistema, junto con la curva de interpolación teórica.

ecuación funcional de Cauchy [74], y la hipótesis de inducción $H(\mathcal{T}_{kN}) \geq kH(\mathcal{T}_N)$ se tiene:

$$\begin{aligned} H[\mathcal{T}_{(k+1)N}] = H(\mathcal{T}_{kN+N}) &\geq H(\mathcal{T}_{kN}) + H(\mathcal{T}_N) \\ &\geq kH(\mathcal{T}_N) + H(\mathcal{T}_N) = (k+1)H(\mathcal{T}_N) \end{aligned}$$

Lo cual junto con la identidad $H(\mathcal{T}_{1N}) \geq 1H(\mathcal{T}_N)$ prueba:

$$H(\mathcal{T}_{kN}) \geq kH(\mathcal{T}_N),\ \forall k \in \mathbb{N}^+ \tag{3.49}$$

Tomando ahora $N = 2$ en (3.49) se tiene:

$$H(\mathcal{T}_{2k}) \geq k\cancelto{1}{H(\mathcal{T}_2)} \tag{3.50}$$

donde se ha hecho la sustitución $H(\mathcal{T}_2) = 1$ porque cualquier topología de dos celdas es de hecho un sistema $\mathcal{N}_{/2}$ con $N = 2$, $\mathcal{T}_{N=2} = (\mathcal{N} = 2)_{/2}$, y por (3.47) se tiene $H(2_{/2}) = 1$. Renombrando la cantidad $2k \to N$ en (3.50) se obtiene el resultado:

$$H(\mathcal{T}_N) \geq N/2 \tag{3.51}$$

el cual tiene dos implicaciones relevantes: (*i*) demuestra que la tasa mínima de incremento de la entropía con el tamaño N de la matriz de celdas en un sistema modular es $N/2$, lo cual corresponde precisamente a la topología $\mathcal{N}_{/2}$; y (*ii*) dado

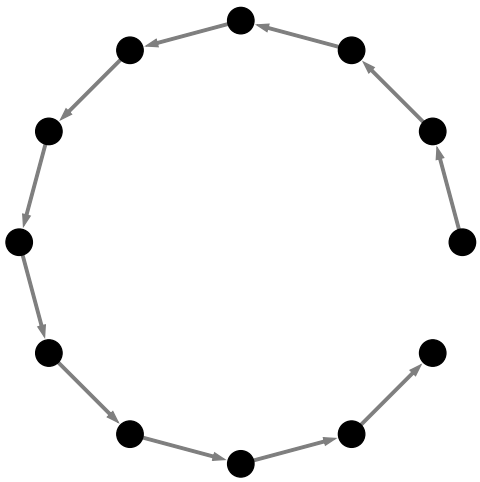


Fig. 3.11.: Grafo típico representante de la topología $\mathcal{N}_{-1}$ con $N = 12$.

que la cantidad $N/2 \sim N$ escala linealmente con N, permite escribir una cota mínima general para la entropía en topologías modulares:

$$H(\mathcal{T}_N) \gtrsim N \tag{3.52}$$

Así mismo, dado que la entropía está acotada superiormente por el número de bits producidos por una topología $H(\mathcal{T}_N) \leq N_b(N)$, si la función $N_b(N)$ es lineal con N, $N_b \sim N$, se tiene:

$$N \lesssim H(N) \lesssim N \Longrightarrow H(N) \sim N \tag{3.53}$$

3.3.3. Comparaciones con una repetición: topología $\mathcal{N}_{-1}$

En este esquema de codificación propuesto por Maiti *et al.* en [127], cada respuesta binaria de $N_b = N - 1$ bits, $\vec{y} = (y_1, \cdots, y_{N-1})$, se construye a partir de un vector de N mediciones físicas $\vec{\psi} = (\psi_1, \cdots, \psi_N)$ realizadas en una matriz de N celdas, comparando osciladores sucesivos con una repetición:

$$y_i \equiv \text{bit}(i, i+1) = \begin{cases} 0 & \text{si } \psi_i > \psi_{i+1} \\ 1 & \text{si } \psi_i < \psi_{i+1} \end{cases} \tag{3.54}$$

de tal forma que cada bit involucra la utilización de una nueva celda física —en el sentido de no-utilizada previamente—, véase en la figura 3.11 un grafo característico de esta topología para una matriz de $N = 12$ celdas, $(\mathcal{N} = 12)_{-1}$. Esta solución permite mantener el sistema bajo una buena resistencia ambiental a la par que proporciona una alta densidad de entropía cercana a los N bits (*i.e.*, una densidad

de entropía por bit próxima al $100\,\%$ ideal). En particular, dado que cada nueva comparación utiliza al menos una celda no empleada previamente, la restricción (3.16) se satisface automáticamente (esto es, resulta imposible de violar) ya que por definición de la topología $\mathcal{N}_{-1}$, si se han extraído los bits: $\mathrm{bit}(i,j)$ y $\mathrm{bit}(i,k)$; entonces $\mathrm{bit}(j,k)$ no estará presente en la respuesta puesto que exigiría comparar dos celdas utilizadas previamente. Consecuentemente, los bits de la respuesta no estarán correlacionados debido a la transitividad de la comparación sobre grupos de tres celdas. Sin embargo, resulta notable que todavía existe una cierta correlación entre bits, la cual surge como un artificio debido a la estrategia de digitalización [129]: dada una matriz de N celdas, existirán algunas permutaciones de estas tales que la comparación entre elementos inmediatamente sucesivos dará lugar a una misma respuesta binaria. Para ilustrar este escenario de colisión y la manera en que emerge la correlación entre bits, examinamos el comportamiento de una tupla de tres celdas $(\lambda_1, \lambda_2, \lambda_3)$, que de acuerdo con el modelo definido por (3.3) determinan unívocamente sendas respuestas $[\psi_1(\lambda_1), \psi_2(\lambda_2), \psi_3(\lambda_3)]$. Dado que cada bit depende únicamente del resultado de la comparación entre estas cantidades, el conjunto total de posibles configuraciones diferentes para el vector $\vec{\psi}$ será el conjunto de permutaciones de los elementos ψ_1, ψ_2 y ψ_3; conviniendo en asignar un subíndice a cada respuesta atendiendo a su magnitud relativa, $\psi_1 < \psi_2 < \psi_3$, las posibles realizaciones físicas de la matriz y sus correspondientes respuestas binarias son:

Tupla de respuestas físicas	Repuesta PUF
$\psi_1 < \psi_2 < \psi_3$	$(0,0)$
$\psi_1 < \psi_3 > \psi_2$	$(0,1)$
$\psi_2 > \psi_1 < \psi_3$	$(1,0)$
$\psi_2 < \psi_3 > \psi_1$	$(0,1)$
$\psi_3 > \psi_1 < \psi_2$	$(1,0)$
$\psi_3 > \psi_2 > \psi_1$	$(1,1)$

En esta tabla se expone de manera evidente que la probabilidad de extraer un segundo símbolo y_2 igual al primero, $y_2 = y_1$ es del $33\,\%$, frente a un $66\,\%$ de obtener el símbolo opuesto, $y_2 \neq y_1$. Este comportamiento, que es independiente de las propiedades físicas de la matriz de celdas y emerge únicamente como consecuencia en la elección del codificador, redunda en una sensibilidad incrementada a ataques de diccionario, ya que dota a un potencial adversario de un criterio objetivo para seleccionar respuestas PUF candidatas óptimas durante un intento de suplantación.

Este fenómeno de correlación dará lugar a distribuciones de respuestas no uniformes. Podemos calcular la distribución de probabilidad asociada a la variable

aleatoria $\mathsf{Y}_{\mathcal{N}_{-1}}$ de respuestas $\vec{y}$ dadas por una topología $\mathcal{N}_{-1}$ de N celdas utilizando el modelo de fabricación física expuesto en 3.2: de acuerdo con la función bit dada en (3.54) para la topología $(\mathcal{N}=2)_{-1}$, el número de distintas configuraciones de celdas (del total de M fabricables) tales que el bit no cambia, $\omega_y(2_{-1})$, será el número de parejas $\psi_1=\mathcal{F}(m_1), \psi_2=\mathcal{F}(m_2)$ que cumplan $y=\mathrm{bit}(1,2)=\text{cte}$; esto corresponde con el conjunto de celdas $m_1>m_2$ si $y=0$, o recíprocamente $m_1<m_2$ si $y=1$. Del total de M celdas ordenadas esto es:

$$\omega_y(2_{-1})=\begin{cases}\sum_{m_2=1}^{M}\sum_{m_1=1}^{m_2}(1-\delta_{m_1,m_2}), & \text{si } y=1\\[2ex] \sum_{m_2=1}^{M}\sum_{m_1=m_2}^{M}(1-\delta_{m_1,m_2}), & \text{si } y=0\end{cases} \tag{3.55}$$

donde la delta de Kronecker, δ_{m_1,m_2}, sustrae el número de casos imposibles (*i.e.*, $m_1=m_2$) que hemos contado en las sumas; dado que la variable y es una cantidad booleana que sólo puede tomar los valores 0 / 1, esta expresión puede escribirse de forma más compacta como:

$$\omega_y(2_{-1})=\sum_{m_2=1}^{M}\sum_{m_1=1+y(m_2-1)}^{m_2+y(M-m_2)}(1-\delta_{m_1,m_2}) \tag{3.56}$$

Y esta se amplía a todo un vector de respuesta $\vec{y}=(y_1,\cdots,y_{N-1})$ sumando iterativamente el número de configuraciones equivalentes de las palabras binarias sucesivamente menores:

$$\omega_{\vec{y}}(\mathcal{N}_{-1})=\sum_{m_N=1}^{M}\sum_{m_{N-1}=1+y_{N-1}(m_N-1)}^{m_N+y_{N-1}(M-m_N)}\cdots \tag{3.57}$$

$$\sum_{m_2=1+y_2(m_3-1)}^{m_3+y_2(M-m_3)}\sum_{m_1=1+y_1(m_2-1)}^{m_2+y_1(M-m_2)}(1-\delta_{m_N,m_{N-1},\cdots,m_2,m_1}) \tag{3.58}$$

Pasando al continuo en el límite $M\gg 1$ (3.20), y utilizando (3.41), podemos escribir (3.57) como:

$$\omega_{\vec{y}}(\mathcal{N}_{-1})\xrightarrow[M\to\infty]{} M^N\int_0^1 d\rho_N\int_{\rho_N y_{N-1}}^{\rho_N+(1-\rho_N)y_{N-1}}d\rho_{N-1}\cdots \int_{\rho_2 y_1}^{\rho_2+(1-\rho_2)y_1}d\rho_1+\mathcal{O}\left(M^{N-1}\right) \tag{3.59}$$

Y aplicando (3.19) y (3.21):

$$p_{\vec{y}}(\mathcal{N}_{-1})=\int_0^1 d\rho_N\int_{\rho_N y_{N-1}}^{\rho_N+(1-\rho_N)y_{N-1}}d\rho_{N-1}\cdots\int_{\rho_2 y_1}^{\rho_2+(1-\rho_2)y_1}d\rho_1 \tag{3.60}$$

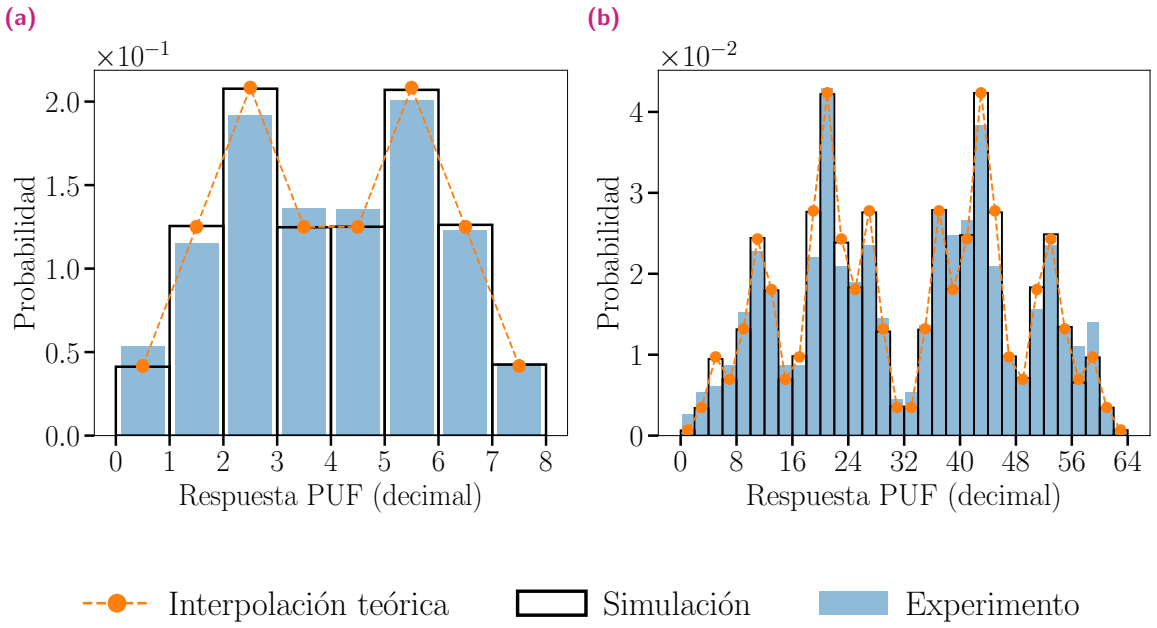


Fig. 3.12.: Histogramas de las respuestas para la topología $\mathcal{N}_{-1}$ con $N = 4$ celdas (a) y $N = 7$ celdas (b), obtenidos mediante simulación y medición experimental, superpuestos sobre la distribución teórica esperada.

En las figuras 3.12a y 3.12b se han representado simultáneamente los histogramas obtenidos por simulación y evaluación experimental para la distribución de la variable $Y_{\mathcal{N}_{-1}}$ sobre matrices de $N = 4$ y $N = 7$ celdas respectivamente; así mismo, se ha superpuesto la distribución de probabilidad deducida en (3.60) para la topología $\mathcal{N}_{-1}$.

En este caso, la evaluación general de (3.60) no es sencilla, de modo que resulta impracticable deducir constructivamente una ley de escala para la entropía con el tamaño de la matriz de celdas; en su lugar, recurrimos al resultado (3.53) para construir una recta de interpolación,

$$H(\mathcal{N}_{-1}) = aN + b \tag{3.61}$$

donde las constantes a, b estarán dadas por:

$$a = H(\mathcal{3}_{-1}) - H(\mathcal{2}_{-1}) \tag{3.62}$$

$$b = 3H(\mathcal{2}_{-1}) - 2H(\mathcal{3}_{-1}) \tag{3.63}$$

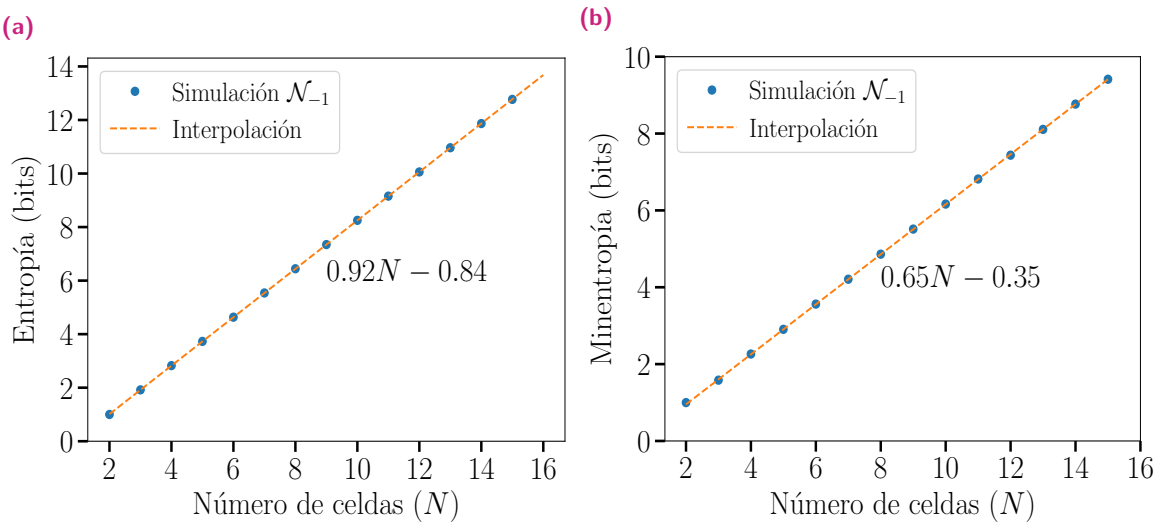


Fig. 3.13.: Entropía (a) y minentropía (b) entregada por la topología $\mathcal{N}_{-1}$ en función del número N de celdas físicas del sistema, junto con las curvas de interpolación teórica correspondientes.

Para los casos $N = 2$, $N = 3$ las distribuciones de probabilidad (3.60) explícitas para cada respuesta binaria son:

$$p_y(2_{-1}) = 1/2 \tag{3.64}$$

$$p_{(y_1,y_2)}(3_{-1}) = 1/6 + y_1/6 + y_2/6 - y_1 y_2/3 \tag{3.65}$$

Y las entropías asociadas:

$$H(2_{-1}) = 1\,\text{bit} \tag{3.66}$$

$$H(3_{-1}) = \log_2(3) + 1/3\,\text{bits} \tag{3.67}$$

Sustituyendo (3.66) y (3.67) en (3.62) y (3.63) se obtiene para la curva de interpolación de la entropía:

$$H(\mathcal{N}_{-1}) \approx 0{,}92N - 0{,}84\,\text{bits} \tag{3.68}$$

la cual se muestra en la figura 3.13a (línea naranja) junto a los valores de entropía hallados mediante simulación (puntos azules) en función del número de celdas N de la matriz.

Dado que en esta topología la distribución de probabilidad sobre las respuestas posibles no es uniforme, la minentropía no coincidirá en general con la entropía. No obstante, puede deducirse una curva de interpolación lineal de modo similar al

caso de la entropía, utilizando el hecho de que la topología $\mathcal{N}_{/2}$ es un módulo de $\mathcal{N}_{-1}$. Esto permite escribir para las variables aleatorias asociadas a ambas topologías $\mathsf{Y}_{\mathcal{N}_{-1}} = (\mathsf{Y}_{\mathcal{N}_{/2}}, \Delta)$, y por (2.52) $H^{\mathrm{m}}(\mathcal{N}_{-1}) \geq H^{\mathrm{m}}(\mathcal{N}_{/2})$, lo cual por (3.47) se puede escribir:

$$H^{\mathrm{m}}\left(\mathcal{N}_{/2}\right) \geq N/2 \Longrightarrow H^{\mathrm{m}}\left(\mathcal{N}_{/2}\right) \gtrsim N \tag{3.69}$$

Así mismo, dado que la minentropía está acotada superiormente por la entropía, por (3.68) se tiene:

$$H^{\mathrm{m}}\left(\mathcal{N}_{/2}\right) \lessapprox 0{,}92N - 0{,}84 \Longrightarrow H^{\mathrm{m}}\left(\mathcal{N}_{/2}\right) \lesssim N \tag{3.70}$$

Combinando (3.69) y (3.70):

$$N \lesssim H^{\mathrm{m}}\left(\mathcal{N}_{-1}\right) \lesssim N \Longrightarrow H^{\mathrm{m}}\left(\mathcal{N}_{-1}\right) \sim N \tag{3.71}$$

Ahora utilizamos la distribución de probabilidad (3.60) para construir una curva de interpolación lineal de modo análogo al caso de la entropía, obteniendo:

$$H^{\mathrm{m}}\left(\mathcal{N}_{-1}\right) \approx 0{,}65N - 0{,}35 \,\text{bits} \tag{3.72}$$

En 3.13b se ha representado la minentropía frente al número de celdas N obtenida por simulación para el sistema $\mathcal{N}_{-1}$, junto con la curva (3.72).

3.3.4. Comparaciones en grupos de K celdas: topología $\mathcal{N}^2_{/K}$

La entropía máxima extraíble de una matriz de N celdas corresponde a la topología $\mathcal{N}^2$, tal como se indica en (3.26), lo cual convierte a esta opción en la más eficiente desde el punto de vista de los recursos *hardware* implicados en la matriz, al ser la estructura capaz de proporcionar una mayor entropía y minentropía con una menor cantidad de celdas físicas. El número de bits generados por un diseño $\mathcal{N}^2$ es máximo con respecto a cualquier otra topología que emplee el mismo número de celdas, lo cual obliga a implementar *in silico* algoritmos comparativamente más complejos para generar cada respuesta, así como protocolos de comunicación capaces de transmitir y gestionar un gran número de bits altamente sensibles. Estos inconvenientes impactan negativamente en el rendimiento potencial de un diseño, y aumenta la probabilidad de errores. Por otra parte, el resultado (3.51) a propósito de la topología $\mathcal{N}_{/2}$ sitúa a esta como la alternativa más ineficiente

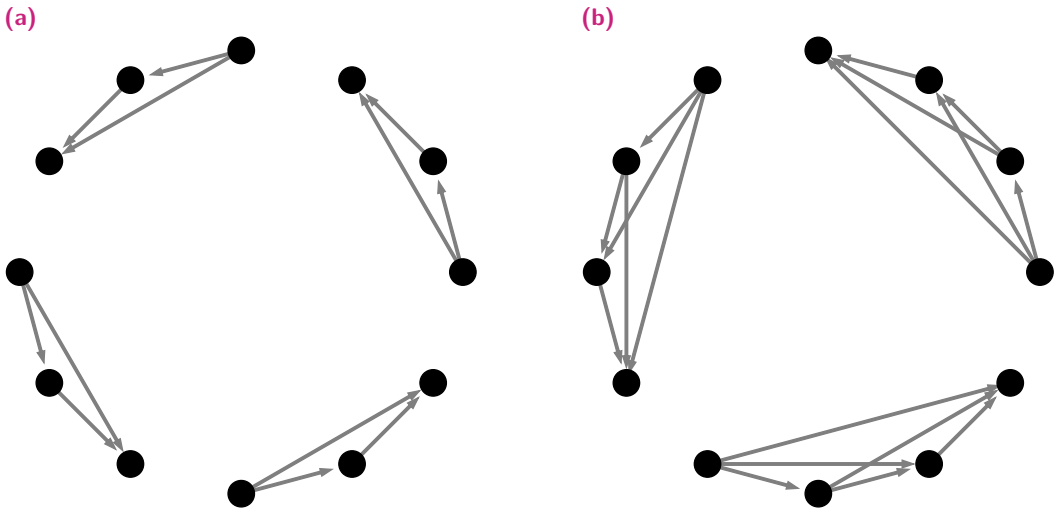


Fig. 3.14.: Grafo típico representante de la topología $\mathcal{N}^2_{/K}$ para los casos $K = 3$ (a) y $K = 4$ (b), con $N = 12$.

desde el punto de vista de los recursos *hardware* empleados en la matriz, siendo esta la arquitectura que proporciona una menor cantidad de entropía por celda. No obstante, las respuestas binarias devueltas por este esquema de comparación son máximamente compactas, con una razón de entropía por bit predicha del 100 %. En esta sección buscamos una manera sistemática de aproximarnos a topologías que proporcionen un compromiso sucesivamente mejor en cuanto a entropía por número de celdas, y entropía por número de bits. Para ello, presentamos una familia novedosa original de topologías que hemos denominado "K-modular", $\mathcal{N}^2_{/K}$, y que constituyen un intento de combinar la propiedad de no-correlación característica de la topología $\mathcal{N}_{/2}$, y la alta densidad de entropía por celda de $\mathcal{N}^2$. Una topología $\mathcal{N}^2_{/K}$ se construye dado un conjunto de N celdas separando estas en N/K grupos de K elementos. Cada uno de estos módulos inconexos es evaluado comparando todas las parejas posibles, siguiendo una topología completa $(\mathcal{N} = \mathcal{K})^2$, dando lugar a respuestas binarias de $K(K-1)/2$ bits. La respuesta completa de una matriz de N celdas se construye mediante la concatenación de cada respuesta parcial, generando una palabra binaria de $N_b = N(K-1)/2$ bits. Dado que cada módulo está desconectado del resto, la entropía y minentropía total entregada por este sistema será la suma aritmética de las cantidades de cada módulo,

$$H(\mathcal{N}^2_{/K}) = H^{\mathrm{m}}(\mathcal{N}^2_{/K}) = N/K \times \log_2 K! \tag{3.73}$$

Nótese que las topologías referidas como $\mathcal{N}^2$ y $\mathcal{N}_{/2}$ son de hecho casos particulares de la topología $\mathcal{N}^2_{/K}$ tomando $K = N$ y $K = 2$ respectivamente. En este capítulo

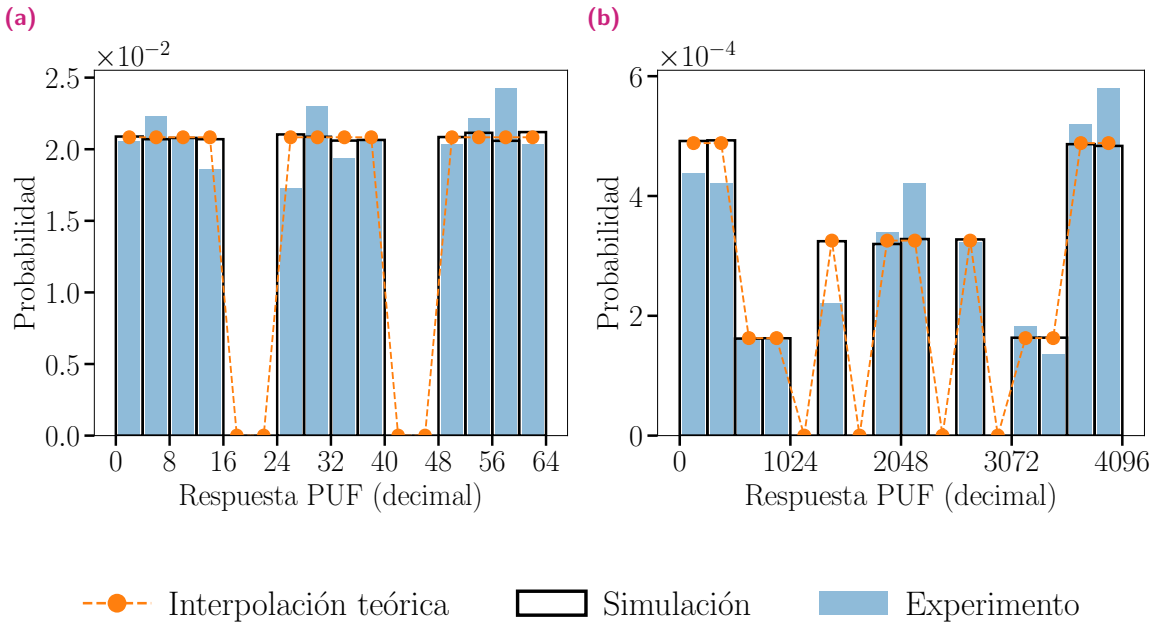


Fig. 3.15.: Histogramas de las respuestas para la topología $\mathcal{N}^2_{/K}$ con $K = 3$ celdas por módulo, $N = 6$ celdas (a) y $K = 4$ celdas por módulo, $N = 8$ celdas (b), obtenidos mediante simulación y medición experimental, superpuestos sobre la distribución teórica esperada.

nos centramos en los casos de estudio $K = 3$ y $K = 4$, para cada una de las cuales se ha representado un grafo típico en las figuras 3.14a y 3.14b, correspondientes a las topologías $(\mathcal{N} = 12)^2_{/3}$ y $(\mathcal{N} = 12)^2_{/4}$ respectivamente. En la figura 3.15 se muestran los histogramas obtenidos experimentalmente y mediante simulación para las distribuciones de probabilidad de las variables aleatorias $\mathsf{Y}_{\mathcal{6}^2_{/3}}$ y $\mathsf{Y}_{\mathcal{8}^2_{/4}}$, correspondientes a las topologías K-modular con $K = 3$, $K = 4$ respectivamente, junto con las distribuciones de probabilidad deducidas mediante la aplicación de (3.24) y la restricción (3.17) a cada módulo.

Finalmente, en la figura 3.16 se muestra la entropía obtenida por simulación para las topologías $\mathcal{N}^2_{/K}$ con $K = 3$ y $K = 4$, junto con la curva de interpolación (3.73).

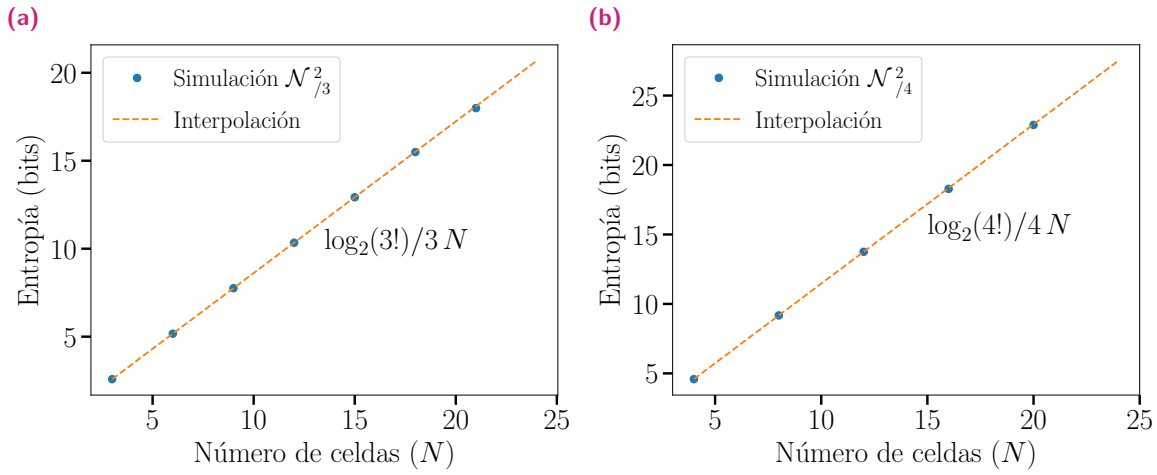


Fig. 3.16.: Entropía entregada por la topología $\mathcal{N}^2_{/K}$ con $K = 3$ (a) y $K = 4$ (b), en función del número N de celdas físicas del sistema, junto con las curvas de interpolación teórica correspondientes.

3.4. Comparación entre topologías

Definimos las siguiente cantidades como figuras de mérito para evaluar el desempeño de cada topología estudiada sobre PUF de medida compensada de N celdas, cuyas respuestas constan de N_b bits:

1. Densidad de entropía por celda, $\varrho(\mathcal{T}_N) \equiv H(\mathcal{T}_N)/N$, y minentropía por celda, $\varrho^{\mathrm{m}}(\mathcal{T}_N) \equiv H^{\mathrm{m}}(\mathcal{T}_N)/N$; estas cantidades son una medida del rendimiento del sistema en relación al consumo de recursos *hardware*, toda vez que un mayor número de celdas supone una mayor superficie de silicio, lo que se relaciona directamente con una mayor actividad de interrupción, implicando un mayor consumo de potencia.

2. Tasa de entropía por bit, $h(\mathcal{T}_N) \equiv H(\mathcal{T}_N)/N_b$, y tasa de minentropía por bit, $h^{\mathrm{m}}(\mathcal{T}_N) \equiv H^{\mathrm{m}}(\mathcal{T}_N)/N_b$. Mayores tasas de entropía permiten alcanzar cotas superiores de seguridad utilizando respuestas binarias de menor longitud, lo cual a la postre impacta positivamente en el rendimiento del diseño físico al reducir la cantidad de lógica necesaria para producir, procesar y almacenar las respuestas. Una complejidad reducida en el diseño de la electrónica auxiliar, además, minimiza la probabilidad de errores tanto en el proceso de diseño como en fase de ejecución.

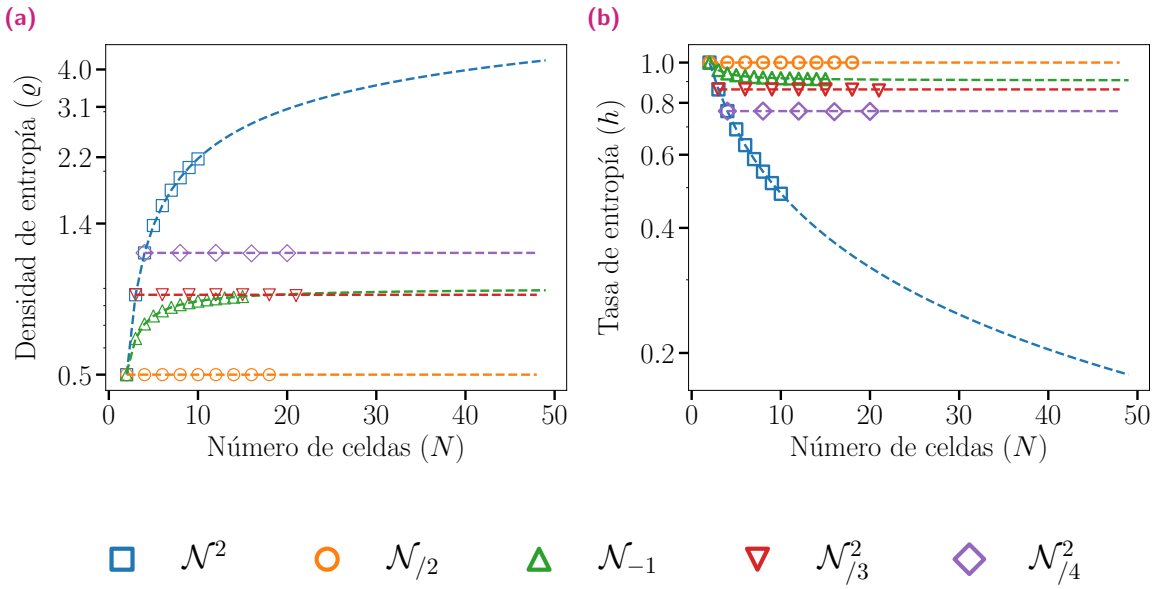


Fig. 3.17.: (a) Densidad de entropía y (b) tasa de entropía para cada topología estudiada (marcadores), junto con sus curvas de interpolación (líneas discontinuas).

En la figura 3.17 se ha representado la densidad de entropía frente al número de celdas (a) y tasa de entropía frente al número de bits (b) de las respuestas extraídas de un sistema PUF para cada una de las topologías estudiadas. El caso de interés para la aplicación de estas topologías PUF en un criptosistema radica en el límite asintótico $N_b \gg 1$; esta región resulta accesible mediante las curvas de interpolación detalladas en la sección 3.3 para cada topología. En este extremo, la arquitectura $\mathcal{N}_{/2}$ proporciona los valores mínimo de $0{,}5\,{}^{\text{bit}}/_{\text{celda}}$ y máximo del $100\,\%$ posibles para la densidad de entropía por celda y tasa de entropía por bit, respectivamente. Por otra parte, la topología $\mathcal{N}_{-1}$ en el límite asintótico de N elevado conserva una densidad de entropía $\varrho \simeq 0{,}92\,{}^{\text{bit}}/_{\text{celda}}$, y una tasa de entropía $h \simeq 92\,\%$. Para la topología completa $\mathcal{N}^2$, el comportamiento de la entropía en el límite $N \to \infty$ se puede estudiar aplicando la aproximación de Stirling al logaritmo del factorial, $\log N! \sim N \log N$, lo cual anticipa un comportamiento extremo en el cual la densidad de entropía diverge, mientras la tasa de entropía por bit tiende a anularse. Finalmente, a propósito de las nuevas topologías propuestas, las cantidades asintóticas para $\mathcal{N}^2_{/3}$ son $\varrho = 0{,}86\,{}^{\text{bit}}/_{\text{celda}}$ y $h = 86\,\%$, mientras que las mismas magnitudes en el caso $\mathcal{N}^2_{/4}$, se estiman en $\varrho = 1{,}15\,{}^{\text{bit}}/_{\text{celda}}$ y $h = 76\,\%$. Estas curvas muestran la existencia de una relación inversa entre la densidad de entropía y la tasa de entropía por bit. En este contexto, la propuesta $\mathcal{N}^2_{/4}$ proporciona el mejor compromiso entre ambas cantidades.

Tab. 3.1.: Comparativa del coste ($\mathcal{C}$) y mincoste ($\mathcal{C}^{\mathrm{m}}$) para cada topología estudiada en el límite asintótico $N \to \infty$, en función del parámetro de diseño α.

Topología	$\mathcal{C}(\boldsymbol{N \to \infty})$	$\mathcal{C}^{\mathrm{m}}(\boldsymbol{N \to \infty})$
$\mathcal{N}^2$	$\mathcal{O}(N/\log N) \to \infty$	$\mathcal{O}(N/\log N) \to \infty$
$\mathcal{N}_{/2}$	$2 + \alpha$	$2 + \alpha$
$\mathcal{N}_{-1}$	$1{,}09 + 1{,}09\alpha$	$1{,}54 + 1{,}54\alpha$
$\mathcal{N}^2_{/3}$	$1{,}16 + 1{,}16\alpha$	$1{,}16 + 1{,}16\alpha$
$\mathcal{N}^2_{/4}$	$0{,}87 + 1{,}31\alpha$	$0{,}87 + 1{,}31\alpha$
Ideal	$\mathcal{O}(N^{-1}) + \alpha \to \alpha$	$\mathcal{O}(N^{-1}) + \alpha \to \alpha$

Además de estas métricas, resulta conveniente proporcionar una figura de mérito que sea capaz de capturar simultáneamente el impacto del número de celdas así como la longitud de las respuestas en el consumo de recursos *hardware*, en función de un factor de proporcionalidad α que mida el consumo de recursos para procesar cada bit en unidades de celdas físicas. Así, $\alpha = 1\,{}^{\mathrm{celda}}/_{\mathrm{bit}}$ representa que la cantidad de recursos —superficie y/o potencia— necesarios para procesar un bit es igual a los recursos consumidos por una celda física. Esta cantidad toma valores típicos del orden de la unidad, por ejemplo en [130] Merli *et al.* implementan una RO-PUF sobre FPGA utilizando $N = 129$ osciladores para construir una respuesta de $N_b = 128$ bits mediante una topología $\mathcal{N}_{-1}$, empleando en total 512 *slice*, lo que supone un factor de diseño $\alpha = 128/(512 - 129) = 0{,}334\,{}^{\mathrm{celda}}/_{\mathrm{bit}}$. Este cálculo arroja valores similares para otros trabajos, *e.g.*, [131] ($\alpha = 0{,}67\,{}^{\mathrm{celda}}/_{\mathrm{bit}}$) o [132] ($\alpha = 1{,}25\,{}^{\mathrm{celda}}/_{\mathrm{bit}}$). Utilizando el parámetro de diseño α podemos definir una función "coste":

$$\mathcal{C}(\mathcal{T}_N) \equiv \frac{1}{\varrho\,(\mathcal{T}_N)} + \frac{\alpha}{h\,(\mathcal{T}_N)} \tag{3.74}$$

y análogamente, la función "mincoste":

$$\mathcal{C}^{\mathrm{m}}(\mathcal{T}_N) \equiv \frac{1}{\varrho^{\mathrm{m}}\,(\mathcal{T}_N)} + \frac{\alpha}{h^{\mathrm{m}}\,(\mathcal{T}_N)} \tag{3.75}$$

En la tabla 3.1 se han desglosado los límites asintóticos $N \to \infty$ de estas cantidades en función del parámetro α, el cual es específico de la plataforma y del diseño donde se implemente la solución PUF. Se ha indicado también el valor mínimo ideal de las funciones coste y mincoste, utilizando el hecho de que idealmente la entropía y

minentropía coincidirán y estarán acotadas por el número de bits, que escala cuadráticamente con el número de celdas, $N_b \sim N^2$. De nuevo, en estos resultados destaca la alternativa $\mathcal{N}^2_{/4}$, particularmente en cuanto a su comportamiento minentrópico, la cual resulta superada únicamente por $\mathcal{N}^2_{/3}$ para valores de $\alpha > 1{,}93$, y $\mathcal{N}_{/2}$ para $\alpha > 3{,}65$; que tal y como se ha discutido, suponen valores elevados del parámetro de diseño en relación a sus valores típicos.

3.5. Conclusión

En este capítulo hemos analizado de una manera exhaustiva las distribuciones de probabilidad características de una función no-clonable físicamente de medida compensada, destacando las propiedades relevantes para el diseño de una primitiva criptográfica y el impacto que estas tienen en la seguridad de un sistema criptográfico. Para ello se ha propuesto y elaborado un formalismo capaz de describir una PUF de medida compensada de manera general, estructurado en torno a la noción de "topología" característica una PUF. Este concepto se reconoce como un factor relevante en el diseño de PUF de medida compensada al determinar completamente la técnica de codificación digital que transforma cada medida física en una respuesta binaria, lo cual se ha demostrado que tiene importantes implicaciones para las propiedades de seguridad del sistema, y ha sido utilizado para clasificar las diferentes arquitecturas de PUF basadas en medida compensada. Así mismo, se ha propuesto un modelo de fabricación física cuya validez se suscribe a la evidencia experimental, y que ha sido utilizado para deducir la forma de las distribuciones de respuestas características de cada topología estudiada. También han sido derivadas algunas de sus propiedades más relevantes para las aplicaciones de seguridad de la información, específicamente la entropía y minentropía, y sus correspondientes magnitudes intensivas derivadas, *i.e.*, densidad de entropía en el espacio físico y tasa de entropía.

Estas soluciones han sido contrastadas con resultados de simulación, así como medidas experimentales realizadas sobre un numeroso conjunto de osciladores de anillo implementados en una FPGA Artix 7 , para cada uno de los casos estudiados. Una vez destacados algunos de los principales inconvenientes de las estructuras estándar se ha propuesto una familia alternativa de topologías "K-modulares", la cual proporciona una manera sistemática, modular e iterativa de diseñar una topología con las propiedades de seguridad adecuadas para satisfacer los requerimientos

exigidos por una determinada aplicación, minimizando el sobrecoste en recursos *hardware*.

4 Funciones no-clonables físicamente basadas en osciladores de anillo

En este capítulo analizamos la arquitectura de funciones no-clonables físicamente basadas en osciladores de anillo (*ring oscillator* PUF, RO-PUF). Esta es una propuesta versátil para extraer entropía del *hardware* subyacente, adecuada para su implementación en FPGA, la cual fue descrita originalmente por Gassend *et al.* en 2002 [24]. No obstante, la arquitectura estándar de RO-PUF se basa en la modificación propuesta por Suh y Devadas en 2007 [118]; desde entonces se han llevado a cabo numerosos desarrollos para incrementar el número de bits de las respuestas, *e.g.*, RO-PUF reconfigurable, o propuestas más modernas como *Transformer-PUF* [50], PUF dinámica [133] o PUF interpuesta (*iPUF*) [134]. El elemento fundamental de RO-PUF es un banco de N osciladores de anillo diseñados para ser idénticos, junto con un elemento de medida capaz de capturar las frecuencias de oscilación de cada anillo con una cierta resolución (figura 4.1). Idealmente, la medición de la frecuencia a la que oscila un anillo implementado en una localización concreta de una FPGA determinada será reproducible en el tiempo y única para cada dispositivo, permitiendo su identificación. Sin embargo, en la práctica, estos osciladores son altamente sensibles al ruido global causado por fluctuaciones de tensión y temperatura en el chip, de modo que es una práctica común aplicar una técnica de medida compensada (sección 2.3.5) mediante una función diferencial entre las medidas de frecuencia. Una función empleada a tal efecto ha sido propuesta por Gassend *et al.*, la cual consiste en utilizar la diferencia de frecuencia medida entre parejas de osciladores, tomando el signo de la diferencia como el bit de respuesta.

4.1. Osciladores de anillo

Un oscilador de anillo es un circuito digital astable (*i.e.*, oscilante entre dos estados inestables) construido mediante una sucesión de inversores lógicos (“fases”

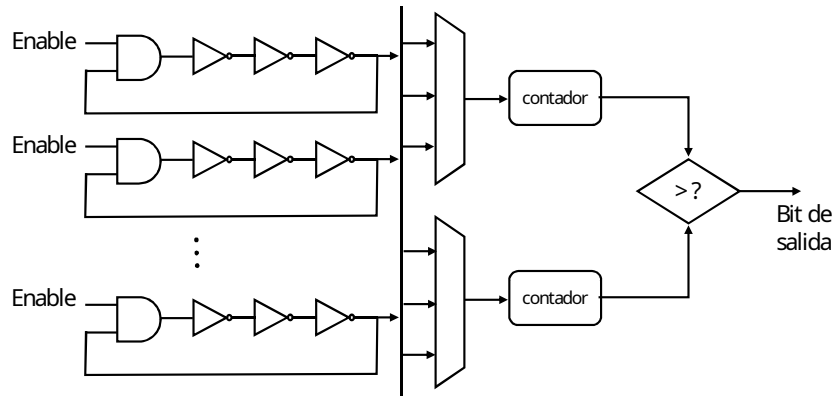


Fig. 4.1.: Esquema de una función no-clonable físicamente basada en osciladores de anillo.

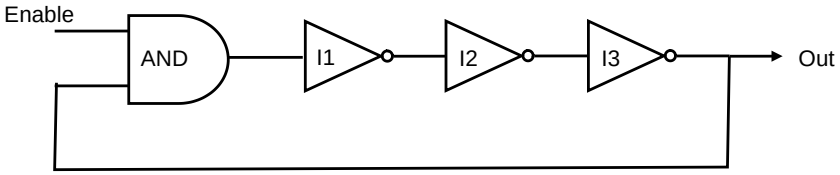


Fig. 4.2.: Esquema de un oscilador de anillo de tres etapas.

o "etapas") y un bucle de realimentación, de tal forma que la salida del último inversor alimente la entrada del primero, tal y como se muestra en la figura 4.2. Esta arquitectura oscila entre dos estados cuando el número de etapas inversoras en el anillo es impar y mayor o igual a tres[1]. En el nivel de abstracción de la lógica, una puerta lógica puede caracterizarse mediante su función booleana (o tabla de verdad), así como un par de constantes $\tau_{\uparrow}$ (retardo de propagación $0 \rightarrow 1$), y $\tau_{\downarrow}$ (retardo de propagación $1 \rightarrow 0$). Si llamamos $\tau \equiv (\tau_{\downarrow} + \tau_{\uparrow})/2$ al retardo promedio de una puerta, y llamando τ_i al retardo total de cada puerta en el oscilador (siendo $i = 0$ la puerta AND inicial utilizada para habilitar el oscilador), el periodo de oscilación del anillo será la suma de los retardos de cada uno de sus elementos, y su frecuencia característica (ν) se podrá escribir como:

$$\nu \equiv \frac{1}{\sum_{i=0}^{N} 2\tau_i} \tag{4.1}$$

donde N es el número de fases del oscilador. Para medir experimentalmente esta cantidad, comparamos el número de ciclos M^{ref} completados por un reloj de referencia de frecuencia ν^{ref} con el número de ciclos M realizados por el oscilador en un mismo intervalo temporal, $M^{\text{ref}}/\nu^{\text{ref}} = M/\tilde{\nu}$, donde $\tilde{\nu}$ representa la estimación

[1]Un único inversor realimentado no oscila porque no satisface el criterio de Barkhausen para la existencia de oscilaciones sostenidas en un sistema realimentado [135].

experimental de la frecuencia del oscilador. De este modo, para el número de ciclos observados se tendrá:

$$M = \frac{\tilde{\nu}}{\nu^{\text{ref}}} M^{\text{ref}} \tag{4.2}$$

Esta expresión permite escribir la variación del número de ciclos M en función de la estimación $\tilde{\nu}$ como:

$$\Delta M = \frac{\delta\tilde{\nu}}{\nu^{\text{ref}}} M^{\text{ref}} \tag{4.3}$$

Y definiendo la resolución de la medida como la variación de frecuencia mínima $\delta\tilde{\nu}^{\text{min}}$ tal que resulta discernible, *i.e.*, provoca una variación de al menos una unidad en la cantidad de flancos medidos M, se puede escribir la resolución en frecuencia de una medida experimental como:

$$\Delta M^{\text{min}} = 1 = \frac{\delta\tilde{\nu}^{\text{min}}}{\nu^{\text{ref}}} M^{\text{ref}} \Longrightarrow \delta\tilde{\nu}^{\text{min}} = \frac{\nu^{\text{ref}}}{M^{\text{ref}}} \tag{4.4}$$

Lo cual indica que la frecuencia de oscilación puede medirse con precisión arbitraria incrementando el tiempo de medida (*i.e.*, el número de ciclos de referencia M^{ref}).

Podemos particularizar el modelo lineal descrito en la sección 2.3.5 para una magnitud física arbitraria ψ al caso de la frecuencia de oscilación ν de un oscilador de anillo, separando la contribución de las componentes locales/globales y aleatorias/deterministas en la frecuencia total:

$$\nu = \nu^0 + \Delta\nu^{gD} + \Delta\nu^{lD} + \Delta\nu^{gA} + \Delta\nu^{lA} \tag{4.5}$$

de manera que la frecuencia de oscilación se puede considerar como una variable aleatoria distribuida normalmente con media igual a la suma de las componentes deterministas, $\mu = \nu^0 + \Delta\nu^{gD} + \Delta\nu^{lD}$, y varianza $\sigma = \sigma^{\text{rep}}$, donde el superíndice "rep" hace referencia a la dispersión de la medida entre repeticiones para una misma instancia. Típicamente, esta variabilidad determina la resolución óptima para una medición, definida como que la componente aleatoria de la frecuencia sea dominante, $\delta\tilde{\nu}^{\text{min}} \lesssim \sigma^{\text{rep}}$.

4.2. Diseño e implementación de matrices de osciladores en FPGA

Para implementar una función no-clonable físicamente resulta primordial controlar tantas variables de diseño como sea posible, de tal modo que el sesgo sistemático debido al diseño de la respuesta PUF sea mínimo. La herramienta Vivado utilizada para realizar los diseños permite un control fino de los recursos empleados a nivel de LUT, sin embargo, este se lleva a cabo a través de un archivo de restricciones y no como parte del código HDL compilable. Esto impide parametrizar los diseños de forma general, de modo que se ha optado por el desarrollo de una *suite software* auxiliar orientada a facilitar el diseño de los módulos implementados en esta tesis. Para el caso específico de matrices de osciladores de anillo sobre la FPGA Artix 7, se ha escrito un módulo de Python "ring_osc", el cual genera todos los archivos necesarios para el diseño dada una serie de datos introducidos en el programa a través de una consola (apéndice E). Este programa se organiza en torno a una clase de objetos "StdMatrix", que se inicializan con los siguientes parámetros de diseño:

- El número de inversores ("N_inv") de que consta cada anillo en la matriz.

- Lista de variables "dominio", cada una de las cuales contiene una lista de coordenadas X,Y referidas a la matriz FPGA, y que indican las posiciones de cada anillo (*i.e.*, la posición de la primera LUT del anillo) donde se implementará físicamente el oscilador. Todos los osciladores en la matriz se diseñan siguiendo el esquema dado en la figura 4.2 (una puerta AND seguida de un número impar de inversores).

- Algunas restricciones físicas relacionadas con la configuración de cada LUT participante en el anillo (*e.g.*, pines físicos empleados, posición BEL de cada LUT del anillo en la *slice* FPGA, etcétera). Esto permite controlar la disposición y ruteado de cada anillo de manera que sean implementados de forma idéntica por las herramientas de síntesis digital, así como configurar los inversores de cada oscilador para utilizar un ruteado de número de nodos mínimo en caso de que la aplicación lo requiera (tabla 2.3).

- Número de puertos de entrada utilizados por cada inversor para la configuración de líneas de retardo programables (sección 4.5).

El objeto “StdMatrix” contiene la función “implement()”, la cual genera las fuentes de diseño, esto es, una lista de archivos en formato Verilog, así como el código C para programar el procesador ARM con la interfaz de comunicación (apéndice D), y archivos de restricciones en formato “XDC”, en su caso. Así mismo, la función también produce una serie de *scripts* en formato “TCL” que pueden ser ejecutados por la herramienta Vivado para automatizar el flujo de diseño digital (apéndice B). Una vez el archivo *bitstream* ha sido volcado sobre la FPGA y programado el ARM, el diseño puede caracterizarse utilizando la función “medir()” del objeto “StdMatrix”. Entre las opciones aceptadas por esta función destacan:

1. Sucesión de osciladores de anillo a medir, enumerados desde 0 hasta $N-1$, donde N es el número de anillos implementados en la matriz.

2. Lista de PDL a medir, en caso de que los inversores hayan sido implementados en configuración PDL, enumerados de 0 a 31.

3. Número de repeticiones de cada medida.

4. Intervalo temporal, medido en unidades del periodo del reloj de referencia, durante el cual se mide el número de ciclos ejecutados por cada inversor, *i.e.*, resolución de la medida de frecuencia.

La función “medir()” devuelve un objeto “NumpyArray” con tres ejes: índice de oscilador, índice PDL y número de repetición. Una vez se dispone de las medidas de una matriz de osciladores, esta puede procesarse para construir la respuesta mediante la comparación de pares de anillos (sección 2.3.5).

4.3. Estrategias de selección de osciladores de anillo en FPGA

A pesar de que un oscilador de anillo puede modelarse como un proceso estocástico, tal como se ha destacado en la sección 4.1, es conocido que sus correspondientes medidas de frecuencia están sujetas a diversos fenómenos físicos los cuales inducen correlaciones en las frecuencias características de los osciladores [136]. Uno de estos efectos, de influencia capital, es la correlación debida a la posición del anillo dentro de la FPGA; las causas de estas correlaciones de naturaleza espacial pueden

ser diversas, desde procesos sistemáticos en la fase de fabricación del semiconductor, hasta la alimentación efectiva de los recursos que componen el oscilador debido a la distancia y la cantidad de *hardware* interpuesto entre los recursos del anillo y los pines de alimentación del chip. Este fenómeno conduce a que las comparaciones de muchas parejas de osciladores resulten estériles para la producción de bits con buena calidad criptográfica, ya que el resultado de la comparación estará dominado por la diferencia sistemática debido a las posiciones de los anillos a comparar. Este fenómeno de correlación es conocido desde los trabajos iniciales con PUF basadas en osciladores de anillo, sin embargo, su mitigación se ha limitado típicamente a restringir la producción de bits a parejas de osciladores físicamente próximos en la FPGA, lo cual no está exento de inconvenientes ya que los osciladores de anillo exhiben un fenómeno de "bloqueo frecuencial", por el cual osciladores cercanos tienden a la sincronización debido al acoplo inductivo y capacitivo entre sus componentes [137], [138]. Una posible solución alternativa es la elección cuidadosa de las localizaciones donde se implementarán físicamente los osciladores dentro de la FPGA. En este apartado proponemos y analizamos varios métodos sistemáticos para seleccionar las posiciones en las que implementar los osciladores, entre ellos una propuesta novedosa que minimiza las probabilidades de correlación espacial entre los osciladores de anillo. Esta estrategia es utilizada para generar un experimento PUF, el cual evaluamos en términos de unicidad, reproducibilidad y robustez frente a variaciones ambientales.

4.3.1. Montaje experimental

Para llevar a cabo el análisis descrito anteriormente, se implementa un conjunto de osciladores de anillo sobre un SoC Zynq-7000 montado en una placa de desarrollo PYNQ-Z2, utilizando el procesador ARM Cortex-A9 del chip para sintetizar tanto la señal de reloj como la interfaz de comunicación externa. Así mismo, dispondremos de un ordenador para recolectar las medidas (apéndice D). La arquitectura de cada oscilador consta de tres etapas inversoras realimentadas, junto con una puerta AND que permite activar/desactivar la oscilación de cada anillo (sección 4.2). La frecuencia de oscilación $\tilde{\nu}$ se calcula midiendo el número de ciclos completados por el anillo, M, durante un intervalo de tiempo fijo t^{ref}. Este intervalo temporal se mide intra-chip, utilizando el reloj interno sintetizado por el procesador ARM, el cual oscila a una frecuencia conocida $\nu^{\text{ref}} = 100\,\text{MHz}$. Así, dicho reloj completará $M^{\text{ref}} = t^{\text{ref}}\nu^{\text{ref}}$ ciclos para medir la frecuencia característica de oscilación de un anillo, la cual se puede calcular haciendo $\tilde{\nu} = \nu^{\text{ref}} M/M^{\text{ref}}$. De acuerdo con (4.4),

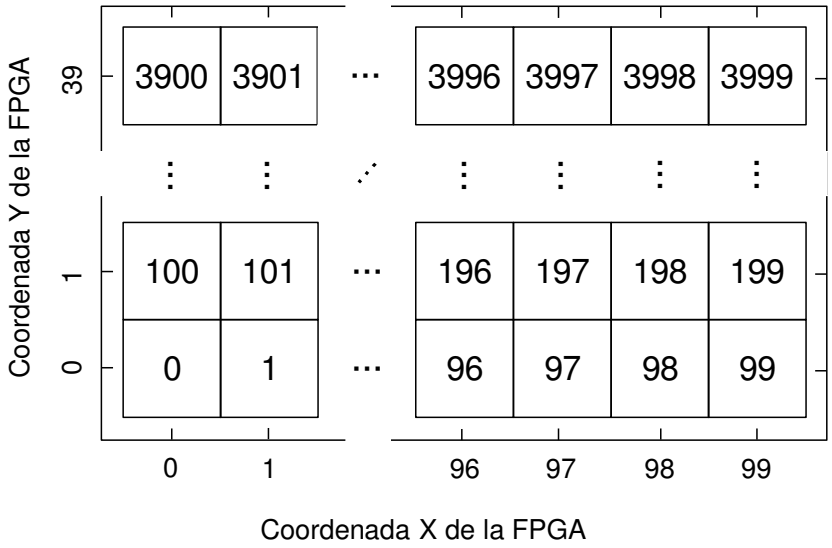


Fig. 4.3.: Definición de los índices de los osciladores y su localización esquemática en la FPGA.

el error máximo posible debido al conteo de ciclos decrecerá con el tiempo de medida (expresado en función del número de ciclos del reloj de referencia) como $\sim 1/M^{\text{ref}}$, sin embargo, en algún momento el ruido gaussiano se hará dominante. En los experimentos descritos a continuación hemos tomado $M^{\text{ref}} = 2^{17}$ ciclos de referencia (función "StdMatrix.medir()" en el apéndice E), de modo que la resolución de la medida será $\sim 7,6 \times 10^{-4}$ MHz, mientras que la menor de las desviaciones típicas encontradas en la matriz de anillos implementada fue de $\pm 0{,}01$ MHz. Esto indica que, en efecto, el tiempo de referencia utilizado es apropiado para medir las características deterministas de los anillos diseñados.

Tal y como se introdujo en la sección 2.4.1, se pueden reconocer al menos tres grados de libertad en la implementación de elementos digitales sobre los recursos de la FPGA, a saber:

1. Tipo de *slice*: (0/1)

2. Tipo de LUT: *logic* (L)/*memory* (M)

3. Orientación del CLB: izquierda (I)/derecha (D)

Para este primer experimento hemos diseñado una matriz de 100 columnas por 40 filas de anillos (figura 4.3). Cada oscilador se implementa ocupando un único *slice* utilizando tres LUT (B, C y D) para los inversores, y una LUT A para la puerta

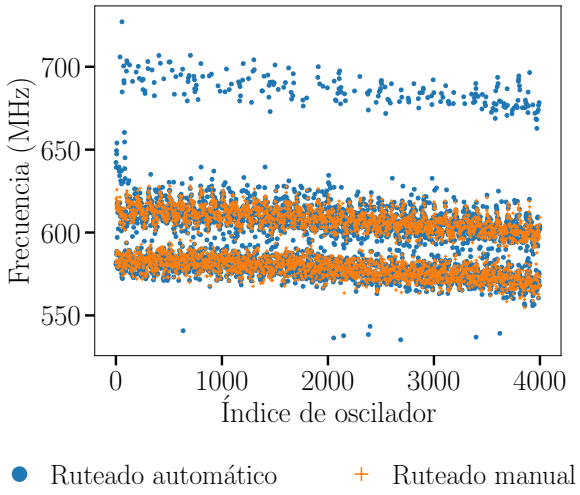


Fig. 4.4.: Frecuencia de los osciladores frente al índice asignado con el ruteado automático seleccionado por la herramienta de diseño (azul), y fijado de manera manual (naranja).

AND que permite activar o desactivar el oscilador. La implementación fina de estos elementos ocupando exactamente los recursos deseados se lleva a cabo utilizando las herramientas *software* desarrolladas a tal efecto para este trabajo, descritas en la sección 4.2. De este modo, se puede identificar cada oscilador con un punto en los ejes "tipo de *slice*, tipo de LUT, orientación del CLB" enumerados anteriormente, permitiendo analizar de manera sistemática el impacto que tiene cada grado de libertad de diseño en los osciladores de anillo, así como en las propiedades de seguridad de la PUF. Este diseño se implementa en 40 dispositivos FPGA de idéntico modelo con el fin de analizar la variabilidad sistemática de la matriz de osciladores de anillo.

4.3.2. Ruteado de los osciladores

Una de las principales ventajas de la PUF basada en osciladores de anillo es que el ruteado de cada oscilador no tiene que ser necesariamente simétrico; sin embargo, todos los osciladores utilizados para generar la respuesta sí deben ser idénticos por diseño entre sí, de forma que debe asegurarse que todos los anillos emplean recursos del mismo tipo; la herramienta EDA Vivado permite automatizar el posicionamiento fino de elementos lógicos a través de un fichero de restricciones (sección 4.3.2). En la figura 4.4 se han representado las frecuencias de los osciladores frente al índice de

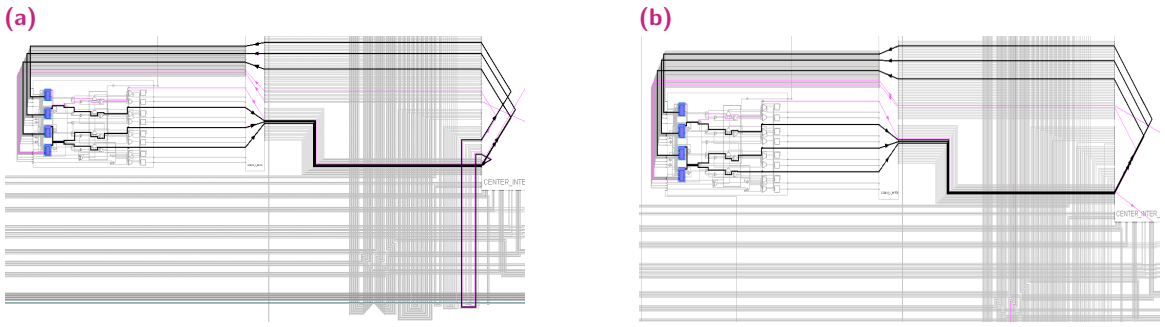


Fig. 4.5.: (a) Oscilador de anillo implementado próximo al borde de la matriz FPGA. (b) Oscilador implementado en una posición alejada del borde. Notar la diferencia en el ruteado.

la posición, calculado como Índice $= 40 \times X + Y$. De esta curva se pueden extraer las siguientes conclusiones:

1. Se aprecian claramente dos dominios de frecuencia: los osciladores implementados en *slice* (1) presentan frecuencias más altas que aquellos sobre *slice* (0); esta discrepancia podría ser atribuible a que cada uno de estos tipos de *slice* emplean recursos de ruteado diferentes.

2. Se observa una correlación negativa en la curva de frecuencia frente al índice de posición del oscilador: la frecuencia característica disminuye de forma aproximadamente monótona con este índice, *i.e.*, los osciladores en la proximidad de X0Y0 tienden a oscilar más rápidamente que sus homólogos en el extremo opuesto, X99Y39.

3. Algunos osciladores tienden a presentar frecuencias mucho más altas que el resto ($\sim 700\,\mathrm{MHz}$); así mismo, algunos anillos presentan frecuencias extraordinariamente inferiores ($\sim 540\,\mathrm{MHz}$). Un examen minucioso de los recursos *hardware* empleados en estos casos concretos reveló que la herramienta de diseño habría introducido diferencias significativas en el ruteado de los osciladores.

4. Los osciladores implementados próximos a los bordes de la FPGA presentan frecuencias más altas que el resto de anillos. Análogamente, los osciladores implementados en localizaciones próximas a algunos de los recursos embebidos en la FPGA (*e.g.*, bloques de RAM) tienden a presentar un ruteado diferente del resto de anillos, produciendo pequeñas diferencias sistemáticas en sus frecuencias características (figura 4.5).

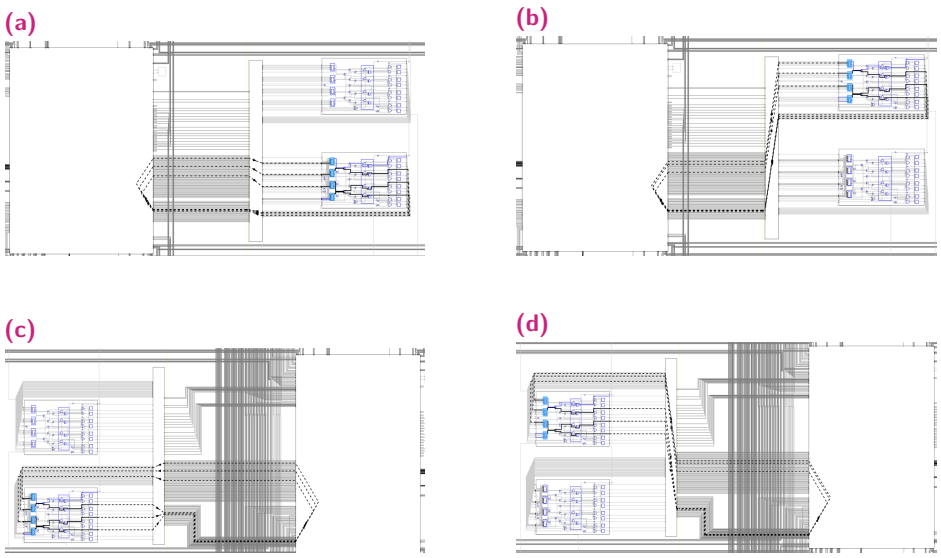


Fig. 4.6.: Posibles configuraciones para el tipo de *slice* y orientación del CLB: (a) 0, D; (b) 1, D; (c) 0, I; (d) 1, I.

En trabajos previos ya se ha destacado la importancia de realizar un ruteado uniforme entre los elementos de cada oscilador para obtener PUF con buena unicidad [130], [139], [140]. Para minimizar estos efectos se procede a fijar el ruteado de los anillos implementados.

Fijado de los recursos de ruteado

Tal y como se expuso en la sección 2.4.1, la plataforma FPGA permite un control detallado de los recursos de ruteado empleados para interconectar los elementos lógicos programables. En la figura 4.6 se muestran los esquemas de ruteado mínimo (de acuerdo con la tabla 2.3) para cada una de las posiciones relativas al CLB en las cuales es posible implementar un oscilador de anillo de tres etapas, tal y como han sido descritos en la sección 4.3.1. A continuación, repetimos la medida anterior utilizando este esquema de ruteado, empleando el lenguaje de *script* "TCL" para forzar al *software* de diseño Vivado a ignorar la optimización del diseño completo, restringiendo el enrutado de cada anillo a su correspondiente CLB. Los nuevos resultados de frecuencia obtenidos se muestran superpuestos sobre la figura 4.4 (puntos de color naranja). En este caso, se observa que muchas de las variaciones debidas a efectos de borde y proximidad con bloques de memoria RAM desaparecen. Sin embargo, todavía se distinguen claramente dos dominios

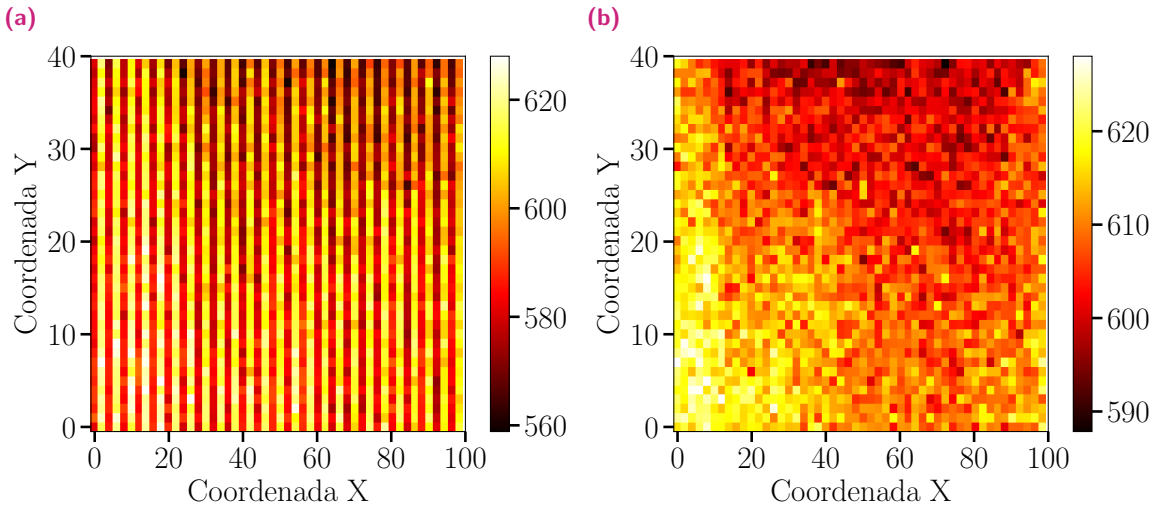


Fig. 4.7.: Frecuencia de los osciladores en función de su posición sobre la FPGA para: (a) cualquier tipo de CLB ocupado en orden; (b) sólo *slice* (1).

frecuenciales, correspondientes a osciladores implementados en *slice* (0) y (1), así como una pendiente de correlación negativa con respecto del índice asignado a cada localización. Más aún, algunas de las discrepancias observadas se pueden atribuir a las diferencias entre LUT (L/M); discernir el efecto de estas estructuras es el objeto de la siguiente sección.

4.3.3. Diferencias entre LUT L/M

Dado que la FPGA está estructurada por columnas (a lo largo de la coordenada Y), los recursos disponibles en cada una de estas son análogos, *i.e.*, dada una coordenada X constante, todos los CLB de la columna correspondiente compartirán un mismo tipo de *slice* (0/1) y una misma orientación I/D. Esta característica se aprecia claramente en la figura 4.7, donde se observa que la distribución de frecuencias presenta una cierta estructura a lo largo de la coordenada X. Para estudiar sistemáticamente el impacto que tienen los detalles de fabricación del chip FPGA en las propiedades de los osciladores de anillo, se han medido las frecuencias características de oscilación a lo largo del eje X, distinguiendo entre cada grado de libertad: tipo de *slice* (0/1), tipo de LUT (L/M) y orientación del CLB (I/D) [141]. En la figura 4.8 se han representado estas frecuencias promediadas a lo largo de la coordenada Y, de donde pueden extraerse algunas conclusiones:

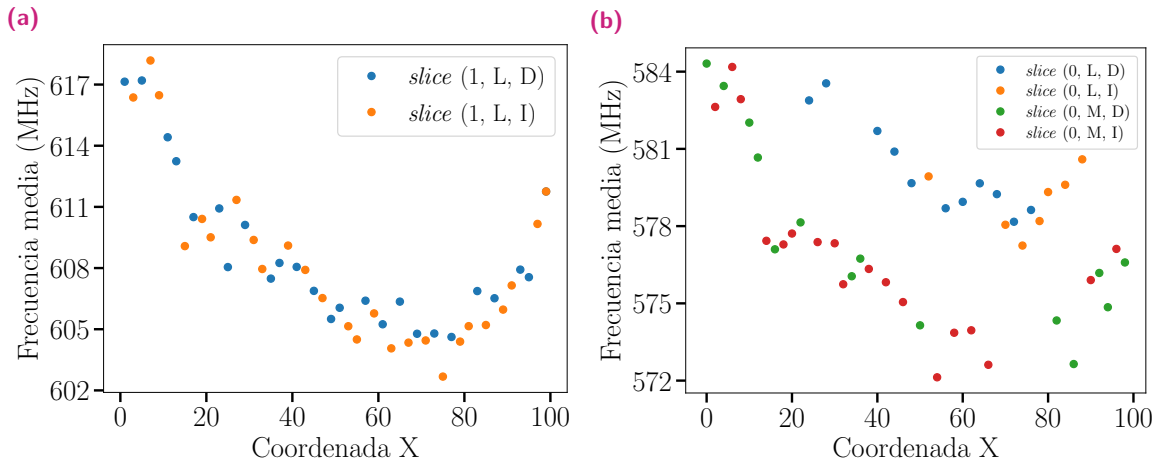


Fig. 4.8.: Frecuencia promedio de los osciladores en función del tipo de *slice* en la cual se ubican: (a) *slice* (0), (b) *slice* (1). Los resultados se han destacado también atendiendo al tipo de LUT (L/R) y su orientación (I/D).

1. El tipo de *slice* (0/1) introduce un componente sistemático en la frecuencia, de tal modo que los osciladores implementados sobre *slice* (1) presentan una frecuencia característica sistemáticamente mayor que sus homólogos implementados en *slice* (0) (figura 4.8).

2. Dentro del subgrupo de anillos construidos sobre *slice* (0), las frecuencias de aquellos implementados utilizando LUT (L) presentan una frecuencia sistemáticamente mayor que aquellos implementados utilizando LUT (M) (figura 4.8b).

3. El impacto de la orientación del CLB (I/D) es despreciable en todos los casos.

Así mismo, en la tabla 4.1 se muestran las frecuencias promedio obtenidas para cada grado de libertad de diseño estudiado. Estos resultados sugieren que, en una misma matriz de anillos, debe evitarse mezclar osciladores implementados en distintos tipos de *slice* (0/1), así como distinto tipo de LUT (M/L), mientras que resulta aceptable combinar elementos implementados en distintas orientaciones (I/D).

Tab. 4.1.: Frecuencia promedio en función del tipo de *slice* (0/1), el tipo de LUT (L/M) y la orientación del CLB (I/D) de los osciladores implementados.

Tipo de LUT	Orientación	Frecuencia (MHz)	
		slice (0)	*slice* (1)
M	D	$577{,}66 \pm 0{,}26$	–
M	I	$576{,}97 \pm 0{,}24$	–
L	D	$580{,}18 \pm 0{,}28$	$609{,}81 \pm 0{,}20$
L	I	$579{,}00 \pm 0{,}29$	$605{,}77 \pm 0{,}22$

4.3.4. Estrategias de selección de osciladores

Para este experimento seleccionamos un conjunto de 200 osciladores del repositorio completo, y utilizamos sus medidas de frecuencia para construir una respuesta PUF basada en osciladores de anillo. La digitalización del resultado se lleva a cabo mediante la comparación de pares de osciladores sin repetición (2-*masking compensated measuring*, referenciada como topología $\mathcal{N}_{/2}$ en la sección 3.3.2), obteniendo una respuesta binaria de longitud $N_b = 100$ bits. Esta topología permite eliminar los efectos artificiales que emergen como resultado de la estrategia de comparación, y que fueron discutidos en el capítulo 3. Esto es conveniente porque facilita el análisis de los resultados debidos exclusivamente a la estructura física de los anillos, incluso aunque las propiedades de seguridad de este sistema como PUF pudieran beneficiarse de las topologías novedosas introducidas anteriormente. En este caso, analizamos las propiedades de la PUF sintetizada para diferentes casos en los que los 200 osciladores son seleccionados atendiendo a diversos criterios:

1. *Naif*, se toman los primeros 200 osciladores del repositorio, indexados de 0 a 199 (*i.e.*, *slice* X0Y0 a X4Y39).

2. *Aleatoria*, se seleccionan 200 osciladores aleatoriamente de entre todos los disponibles en el repositorio.

3. *Naif-específica*, se seleccionan los primeros 200 osciladores implementados utilizando la misma configuración de hardware: *slice* (1), LUT L.

4. *Aleatoria-específica*, se seleccionan 200 osciladores aleatoriamente de entre aquellos implementados sobre una misma configuración *hardware*: *slice* (1), LUT L.

5. *Óptima*, para esta estrategia se seleccionan 200 osciladores atendiendo a que sus frecuencias características sean lo más próximas posible, evitando la comparación entre anillos con respuestas sistemáticas que permitan predecir el resultado de la evaluación de la PUF. Para ello, se calcula la frecuencia promedio, $\overline{\nu}$, sobre la superficie de la matriz de anillos para el conjunto de osciladores implementados sobre *slice* (1) y LUT (L), y se seleccionan los 200 anillos cuyas frecuencias están en el intervalo $\overline{\nu} \pm c_{200}$, donde c_{200} es un factor de cobertura que se toma para que el número de elementos dentro del intervalo sea igual a 200. Notar que el número de osciladores cuyas frecuencias están comprendidas dentro de este intervalo, $n(c)$, es una función monótonamente creciente de c y, por tanto, existirá un tal $c = c_{200}$ que haga $n(c_{200}) = 200$.

4.3.5. Resultados

Cada una de estas estrategias de selección de osciladores ha sido utilizada para implementar una PUF basada en osciladores de anillo sobre 40 chips Zynq-7000, y se han llevado a cabo sendos experimentos PUF repitiendo cada medida 100 veces empleando como único reto el propio proceso de medida. A continuación, se detallan los resultados de unicidad y reproducibilidad obtenidos para cada estrategia, evaluados mediante las estimaciones de inter/intra-distancia de Hamming tal y como fueron definidas en (2.94) y (2.99) respectivamente, junto con las estimaciones de los parámetros $\tilde{p}^{\text{inter}}$ y $\tilde{p}^{\text{intra}}$ que mejor ajustan los histogramas de inter/intra-distancia a una distribución binomial. Este ajuste se ha contrastado con la hipótesis de que la distribución de respuestas PUF subyacente sigue el modelo cuasi-ideal presentado en la sección 2.3.4, el cual predice efectivamente una distribución binomial para las intra/inter-distancias de Hamming en un experimento PUF. El estadístico utilizado para dicho test de hipótesis ha sido la distancia de Kolmogorov-Smirnov D_{KS}, que definimos para una distribución discreta binomial de parámetros n, p y el conjunto de distancias de Hamming obtenidas experimentalmente a partir de una serie de n-vectores binarios como la mayor distancia "vertical" entre el

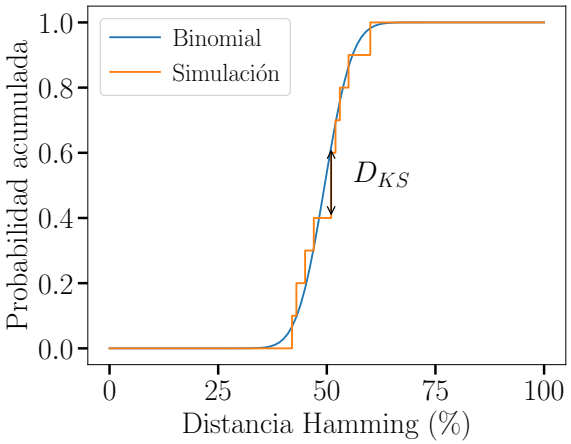


Fig. 4.9.: Distancia de Kolmogorov-Smirnov (D_{KS}) entre la distribución binomial acumulada de parámetros $n = 100$, $p = 0{,}5$, y una muestra aleatoria binomial.

histograma acumulado de las distancias de Hamming experimentales y la curva binomial acumulada:

$$D_{KS} \equiv \max_{0 \le j \le n} \left[\sum_{i=0}^{j} \left| f_i - \underset{n,p}{\mathrm{Bin}}(i) \right| \right] \tag{4.6}$$

siendo f_i la frecuencia de aparición del valor "i" en el conjunto de los datos experimentales; notar que esta cantidad está acotada por los extremos $[0, 1]$. En la figura 4.9 se ilustra el cálculo de la distancia D_{KS} entre el histograma acumulado de 10 muestras extraídas de una variable aleatoria binomial con parámetros $n = 100$, $p = 0{,}5$, y la correspondiente distribución de probabilidad binomial. Dado que no existe una forma cerrada predicha para la distribución del estadístico de Kolmogorov-Smirnov en el caso de modelos discretos, se ha optado por realizar las pertinentes simulaciones para obtener la distancia D_{KS} en cada uno de los casos experimentales analizados. Estas simulaciones se han llevado a cabo tal y como se describe en la sección 2.3.4, y la distribución de distancias D_{KS} esperada en el caso hipotético de que efectivamente una PUF se ajuste al modelo cuasi-ideal se ha obtenido repitiendo cada simulación 10^4 veces para obtener estadística. Esto se ha ilustrado en la figura 4.10, donde se muestran las distribuciones de distancias D_{KS} obtenidas para un experimento simulado con parámetros $p^{\text{intra}} = 0{,}02$ y $p^{\text{inter}} = 0{,}45$. Esta distribución se ha ajustado *ad hoc* a una variable aleatoria S_B de Johnson [142], la cual es una modificación multiparamétrica de la distribución normal adecuada para modelar variables aleatorias acotadas, como es el caso de la distancia de Kolmogorov-Smirnov. A efectos de este trabajo, adoptaremos el enfoque estándar de considerar que la hipótesis cuasi-ideal prevalece (*i.e.*, es no-descartable) si la distancia $\tilde{D}_{KS}$ estimada

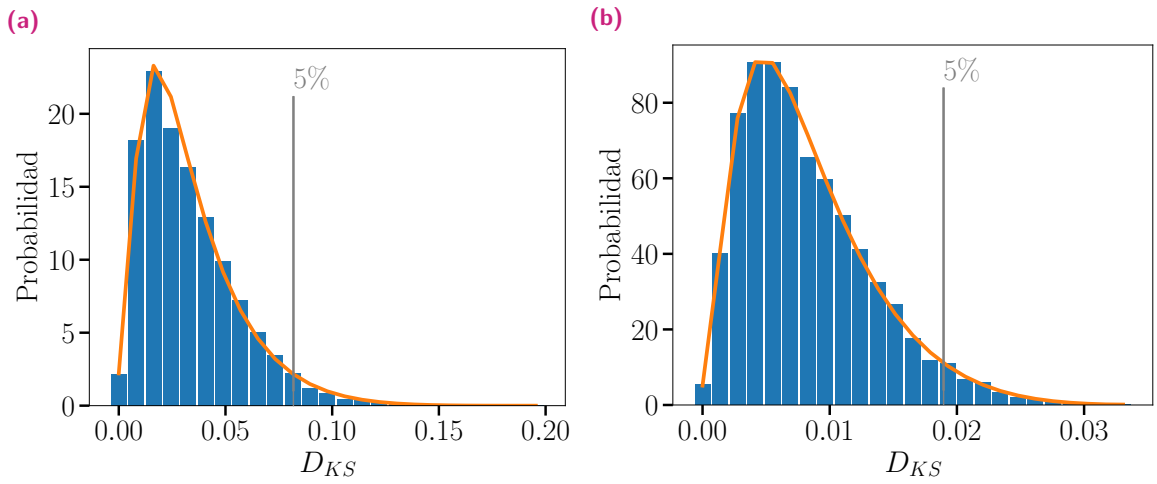


Fig. 4.10.: Distribuciones de la distancia de Kolmogorov-Smirnov para: (a) la inter-distancia y (b) la intra-distancia en un experimento PUF simulado siguiendo un modelo cuasi-ideal. En las figuras se han superpuesto las densidades de probabilidad utilizadas para interpolar los histogramas, y se han marcado los umbrales de significancia $\alpha = 5\,\%$.

experimentalmente está dentro de los primeros 95 percentiles de la distribución D_{KS} hallada mediante simulación, $\tilde{D}_{KS} < D_{KS}|_{5\,\%}$.

Unicidad

En la tabla 4.2 se muestran los resultados de unicidad obtenidos para cada una de las estrategias de selección de osciladores enumeradas, los cuales muestran variaciones muy significativas:

1. *Naif*, tal y como cabía esperar se obtiene una pobre unicidad, dado que se están comparando sistemáticamente osciladores pertenecientes a dominios frecuenciales diferentes, haciendo que las respuestas sean perfectamente predecibles para cualquier instancia.

2. *Aleatoria*, como en el caso anterior, estadísticamente la mitad de las comparaciones realizadas habrán tenido lugar entre anillos de dominios diferentes, lo cual suprime una gran parte de la aleatoriedad de la respuesta.

3. *Aleatoria-específica*, la unicidad se incrementa hasta el $\sim 39\,\%$, claramente más próximo al valor teórico de $50\,\%$. Sin embargo, al seleccionar osciladores de

Tab. 4.2.: Inter-distancia de Hamming promedio para cada estrategia estudiada de selección de osciladores.

Estrategia	$\tilde{\mu}^{\text{inter}}$ (%)	$\tilde{p}^{\text{inter}}$	$\tilde{D}_{KS}$	$D_{KS}\|_{5\,\%}$	Test
Naif	0	0	0	0	–
Aleatoria	19,90	0,199	0,062	0,155	✓
Aleatoria-específica	39,27	0,393	0,149	0,113	✗
Naif-específica	49,64	0,496	0,019	0,041	✓
Óptima	49,54	0,495	0,011	0,042	✓

forma aleatoria, existe la posibilidad de comparar anillos físicamente alejados entre sí, con el consiguiente impacto de la correlación espacial, que introduce un término sistemático en la frecuencia característica tal y como se mostró en la figura 4.4. Este fenómeno impacta negativamente en la aleatoriedad de la respuesta y, por tanto, en la métrica de unicidad. Además, esta estrategia de selección falla el test de hipótesis, obteniendo un valor para la distancia de Kolmogorov-Smirnov $\tilde{D}_{DK} = 0{,}149$ superior al límite para una significancia del $5\,\%$, $D_{KS}\_5\,\% = 0{,}113$.

4. *Naif-específica*, en este caso se observa un notable incremento de la inter-distancia promedio, que se sitúa en el $\sim 49{,}64\,\%$, muy próxima al valor ideal. Esto se debe a la comparación de anillos pertenecientes a un mismo dominio, al mismo tiempo que se mitiga el efecto negativo de la correlación espacial que se observaba en el caso anterior.

5. *Óptima*, en este caso se obtiene un resultado de unicidad similar a la estrategia de selección *Naif-específica*, con una inter-distancia Hamming promedio de $\sim 49{,}54\,\%$, próxima al $50\,\%$ ideal. Sin embargo, este método de selección tiene el inconveniente de requerir un estudio de cada instancia FPGA previo a su despliegue en campo para estimar las localizaciones óptimas en las que implementar los anillos.

Tab. 4.3.: Intra-distancia de Hamming promedio para cada estrategia de selección de osciladores estudiada.

Estrategia	$\tilde{\mu}^{\text{intra}}$ (%)	$\tilde{p}^{\text{intra}}$	$\tilde{D}_{KS}$	$D_{KS}\|_{5\,\%}$	Test
Naif	0	0	0	0	–
Aleatoria	1,732	0,0173	0,2010	0,0189	✗
Aleatoria-específica	0,748	0,0075	0,0224	0,0202	✗
Naif-específica	1,482	0,0148	0,0141	0,0192	✓
Óptima	1,464	0,0146	0,0108	0,0192	✓

Reproducibilidad

En la tabla 4.3 se muestran los resultados para la intra-distancia de Hamming promedio obtenidos, desglosados por estrategia de implementación, junto con los parámetros $\tilde{p}^{\text{intra}}$ que mejor ajustan la distribución binomial, así como la distancia de Kolmogorov-Smirnov obtenida experimentalmente y el correspondiente umbral $D_{KS}\_5\,\%$ calculada mediante simulación para un modelo cuasi-ideal. Tal y como se discutió en 2.3.2, una PUF es tanto más reproducible cuanto más próxima al $0\,\%$ es su intra-distancia estimada. En este sentido, todos los resultados obtenidos para la reproducibilidad son buenos en el contexto del diseño de funciones no-clonables físicamente para FPGA, sin embargo, el test de hipótesis para las alternativas *Aleatoria* y *Aleatoria-específica* devuelven un resultado negativo, indicando una baja fiabilidad respecto de los correspondientes ajustes binomiales. Por otra parte, para las estrategias de selección *Naif-específica* y *Óptima*, el comportamiento de la reproducibilidad es opuesto al de la unicidad en tanto que la segunda presenta un resultado de intra-distancia promedio ($\sim 1{,}46\,\%$) marginalmente mejor que la primera ($\sim 1{,}48\,\%$), siendo en todo caso ambos buenos resultados en relación a otras arquitecturas PUF [29], [143].

Identificabilidad

El análisis de la identificabilidad aúna los resultados previos en una única métrica que permite valorar las estrategias propuestas atendiendo a un criterio

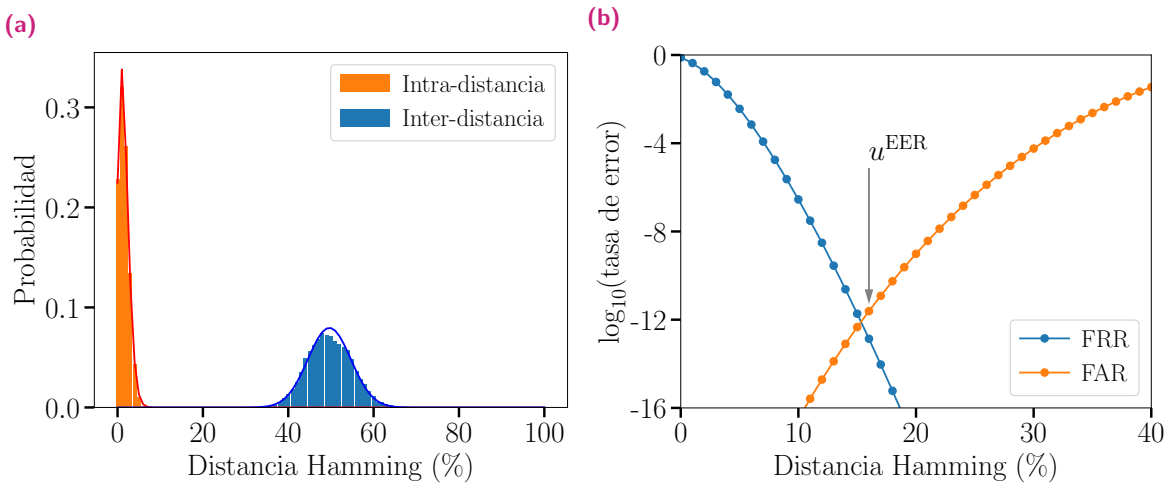


Fig. 4.11.: Métricas de la estrategia de selección *Óptima*: (a) distribuciones de la intra/inter-distancias de Hamming, (b) curvas FAR y FRR, donde se ha destacado el umbral de identificación u^{EER}.

objetivo. Tal y como se detalló en la sección 2.3.3, para estudiar esta magnitud se han construido las curvas FAR y FRR correspondientes a cada estrategia de selección. Estas curvas han sido utilizadas para extraer el umbral de identificación, u^{EER}, correspondiente a un mismo error de falso rechazo y falsa aceptación, EER. Esta magnitud constituye una figura de mérito que permite evaluar cada estrategia. En la figura 4.11b se muestran a modo de ejemplo las curvas FAR, FRR para el caso *Óptima* (correspondientes a los histogramas de intra/inter-distancia de Hamming mostrados en la figura 4.11a), y en la tabla 4.4 se presentan los resultados para la tasa EER y sus correspondientes umbrales u^{EER} hallados. En esta tabla se hace evidente que las estrategias *Naif* y *Aleatoria* ofrecen unas malas propiedades de identificabilidad; con la única restricción de implementar osciladores en un mismo dominio frecuencial (*Aleatoria-específica*), la tasa de error medido se reduce en cuatro órdenes de magnitud hasta los $3{,}15 \times 10^{-10}$, y al restringirnos además a anillos adyacentes (*Naíf-específica*) este resultado mejora a $2{,}13 \times 10^{-12}$. En cambio, la ganancia de la estrategia *Óptima* respecto de *Naif-específica*, si bien positiva, resulta marginal. Este hecho, sumado a la dificultad práctica de implementar la selección *Óptima* de osciladores debido al estudio específico previo que se requiere para cada instancia permite destacar la estrategia de selección de osciladores *Naif-específica* como la más adecuada para la implementación de una PUF basada en osciladores de anillo en FPGA. Por ello, en subsiguientes análisis y salvo que se especifique lo contrario, se utilizará esta estrategia como prototipo ideal.

Tab. 4.4.: Umbral de identificación (u^{EER}) y tasas de error (EER) para cada estrategia estudiada.

Estrategia	u^{EER} (bits)	$\log_{10}(\mathrm{EER})$
Naif	0	0
Aleatoria	7	$-3{,}44$
Aleatoria-específica	11	$-9{,}50$
Naif-específica	16	$-11{,}67$
Óptima	15	$-11{,}73$

Variación de temperatura

Para estudiar el impacto de la temperatura en el rendimiento de una RO-PUF de 200 anillos y topología $\mathcal{N}_{/2}$ implementada siguiendo la estrategia *Naíf-específica* discutida en la sección previa, hemos utilizado una cámara térmica "Aralab Fitoterm 22E", que permite una variación térmica de $-40\,^{\circ}$C a $160\,^{\circ}$C. Para caracterizar térmicamente el prototipo llevamos a cabo un análisis "V" tal y como se ha detallado en la sección 2.3.5, donde utilizamos como condición de referencia la temperatura $T_0 = 20\,^{\circ}$C; realizamos cien repeticiones de cada medida a lo largo de una rampa de temperaturas $-20\,^{\circ}\mathrm{C} \leq T \leq 80\,^{\circ}$C, en intervalos $\Delta T = 10\,^{\circ}$C. El resultado obtenido se muestra en la figura 4.12, donde además de la temperatura medida por el termómetro de la cámara térmica se ha representado la temperatura real del núcleo FPGA, accesible a través de un termómetro integrado en el chip de Xilinx. Como cabría esperar, los resultados de menor intra-distancia Hamming corresponden a las proximidades de la temperatura de referencia; sin embargo, sí resulta notable la resistencia ambiental en cuanto a cambios de temperatura exhibida por este sistema, que proporciona una muy aceptable discrepancia máxima de $\sim 4\,\%$ para el caso extremo de un entorno de operación a $80\,^{\circ}$C.

Variación de la tensión del núcleo

La potencia que alimenta el chip FPGA es gestionada por la placa de desarrollo a través de un módulo de gestión de potencia (*Power Management Unit*, PMU),

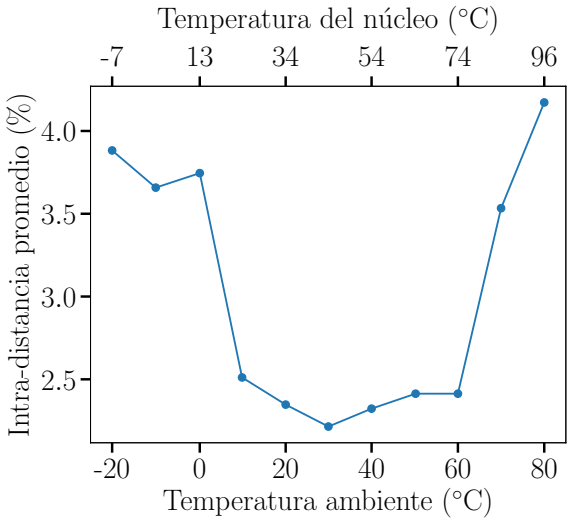


Fig. 4.12.: Evolución de la intra-distancia de Hamming promedio frente a la temperatura ambiente y del núcleo, con temperatura ambiente de referencia $T_0 \equiv 20\,°\text{C}$.

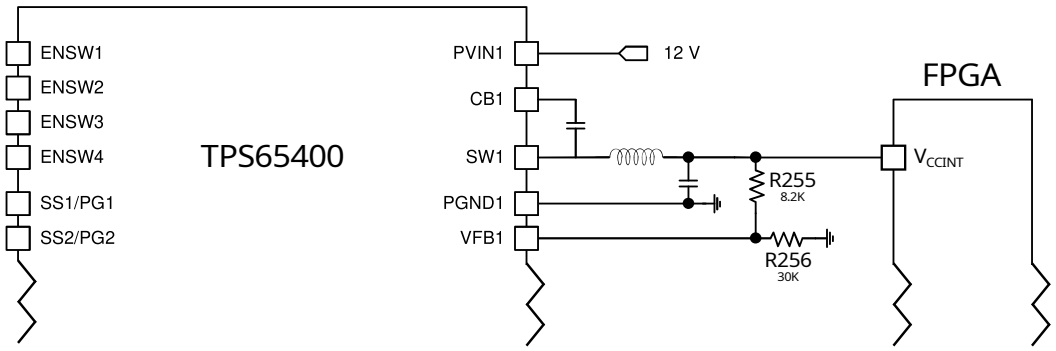


Fig. 4.13.: Detalle del módulo de control de la alimentación (PMU) utilizado en la placa PYNQ-Z2 para alimentar el chip FPGA, destacando la red resistiva que acopla la salida SW1 del PMU con la entrada de potencia V_{CCINT} al núcleo FPGA.

el cual se encarga de mantener unos niveles de voltaje estables para los distintos dominios de tensión que se puedan dar en el chip. En el caso de la placa de desarrollo PYNQ-Z2 que hemos utilizado para estos experimentos, este PMU es un integrado "TPS65400" fabricado por *Texas Instruments* (TI), que dispone de cuatro pines de tensión, denominados "SW1", "SW2", "SW3" y "SW4", capaces de proporcionar niveles estables entre los $0{,}6$ y $1{,}87$ V de forma independiente, lo cual resulta particularmente conveniente para emplear un único PMU cuando hay varios dominios de voltaje, *e.g.*, la FPGA Artix y el procesador Cortex ARM, a pesar de estar integrados en el mismo chip, trabajan a valores distintos de alimentación. El pin de entrada de potencia del núcleo FPGA se denomina "V_{CCINT}", y está acoplado con la salida SW1 del PMU a través de una red resistiva formado por dos resistencias R255 ($8{,}2\,$K) y R256 ($30\,$K), tal y como se muestra esquemáticamente en la figura 4.13. De

acuerdo con la documentación del regulador proporcionada por TI [144], la tensión de salida en el pin V_{CCINT} está dada por:

$$\begin{aligned} V_{\text{CCINT}} &= V_{\text{ref}} \left(1 + \frac{\text{R255}}{\text{R256}}\right) \\ &= V_{\text{ref}} \left(1 + \frac{8{,}2}{30}\right) \\ &\approx 1{,}273 \times V_{\text{ref}} \end{aligned}$$

donde V_{ref} es una tensión de referencia del regulador, la cual puede modificarse a través de una interfaz de comunicación digital I2C implementada en el módulo TPS65400, que permite la introducción de algunos comandos en forma de palabras binarias de 8 bits. En particular, este acepta un comando *VREF_COMMAND* que modifica el valor de V_{ref} en el rango de $0{,}6$ a $1{,}87$ V con una resolución de 10 mV, lo que se traduce en una variación discreta de la tensión de alimentación para el núcleo FPGA en escalones de aproximadamente $12{,}7$ mV.

Finalmente, estudiamos el impacto que tiene esta variación sobre la identificabilidad del prototipo RO-PUF. Para ello, accedemos al regulador de tensión de la placa PYNQ-Z2 y efectuamos un análisis "V" sobre una rampa de tensiones de alimentación que varían entre $0{,}959$ y $1{,}072$ V, respetando un margen de seguridad del $\pm 10\,\%$ respecto de la tensión nominal de alimentación $V_{\text{CCINT}} = 1{,}00$ V referida en la hoja de especificaciones de la FPGA Artix 7 para evitar daños en el chip [145]. Como en el caso anterior, en la figura 4.14 se han representado las intra-distancias de Hamming medidas a distintos valores de tensión respecto de las respuestas PUF obtenidas a una condición ambiental de referencia $V_0 = 1{,}01$ V. En este caso, el impacto de la tensión de alimentación sobre las propiedades de la PUF es mayor que en el análisis previo de la variación térmica, no obstante las discrepancias alrededor del $6\,\%$ u $8\,\%$ halladas para los valores extremos de voltaje siguen siendo aceptables en la práctica del diseño PUF, donde existen propuestas que exhiben intra-distancias de más del $10\,\%$ en condiciones estándar de operación, *i.e.*, temperatura ambiente y tensión nominal de referencia [143].

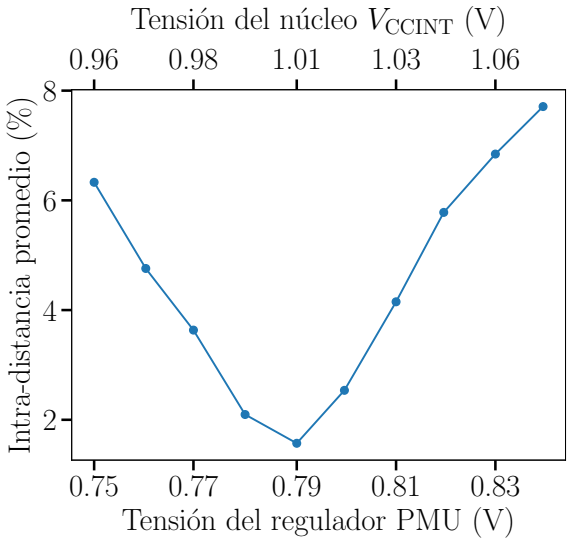


Fig. 4.14.: Evolución de la intra-distancia de Hamming promedio con respecto la tensión del regulador y del núcleo, con una tensión de referencia del núcleo $V_0 = 1{,}01\,\text{V}$.

4.4. Medida compensada de orden superior aplicada a RO-PUF

El objetivo de esta sección es estudiar la posibilidad de extraer más de un bit de cada comparación, multiplicando de manera efectiva la entropía de las respuestas PUF cuya digitalización está basada en la medida compensada de los parámetros físicos que la definen. Operativamente, la técnica de compensación (sección 2.3.5) se puede describir de forma general mediante el siguiente algoritmo:

1. Medir las respuestas físicas de una pareja de celdas, ψ_a, ψ_b.

2. Calcular la diferencia de las medidas anteriores, $\Delta\psi \equiv \psi_a - \psi_b$.

3. Expresar el valor anterior en formato binario, *i.e.*, como un vector de símbolos binarios: $\vec{y}(\Delta\psi) \equiv \text{ADC}(\Delta\psi) = [\text{signo}(\Delta\psi), \text{ADC}(|\Delta\psi|)]$.

4. Extraer el 0-ésimo elemento del vector binario $\vec{y}(\Delta\psi)$ como respuesta de la medida, $y_0(\Delta\psi) = \text{signo}(\Delta\psi)$.

En este apartado analizaremos las propiedades de una PUF cuya respuesta se construye modificando el último paso en el algoritmo anterior para incluir, además del bit

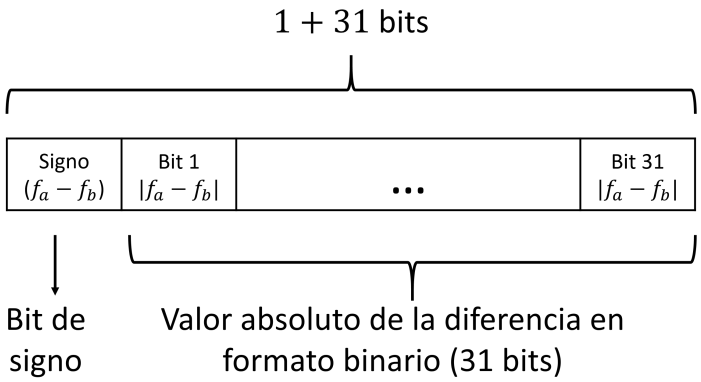


Fig. 4.15.: Representación binaria de la diferencia entre dos medidas de frecuencia utilizando 32 bits.

de signo, bits adicionales procedentes de la representación binaria de la diferencia entre las cantidades medidas [146], [147]. Para llevar a cabo este experimento tomamos 200 osciladores del grupo *Naif-específico* estudiado en la sección anterior, midiendo sus frecuencias características en las mismas condiciones descritas, esto es, utilizando un reloj de referencia de $100\,\text{MHz}$ y tomando datos de cada anillo durante 2^{17} ciclos de referencia ($\sim 1{,}3\,\text{ms}$)[2]. Sin embargo, para construir la respuesta PUF calculamos la diferencia total entre las frecuencias de parejas de la matriz de anillos, en lugar de realizar la mera comparación de los valores obtenidos. La frecuencia se mide del mismo modo descrito en la sección 4.3.1, utilizando ahora 31 bits de resolución para cada medida, de tal modo que el resultado será un número entero en $[0, 2^{31}]$. Así, el intervalo de posibles valores diferenciales es $[-2^{31}, 2^{31}]$, para lo cual serán necesarios como máximo 32 bits. Construimos las respuestas como un vector binario de 32 bits, del cual se utiliza el primer bit para representar el signo de la comparación, y los 31 bits restantes para representar el módulo de la diferencia: dadas dos medidas de frecuencia f_a y f_b, la respuesta será el vector binario $[\text{signo}(f_a - f_b), \text{abs}(f_a - f_b)]$ (figura 4.15). El valor absoluto de la diferencia se representa en formato *little endian*, *i.e.*, el elemento 1-ésimo del vector corresponde al bit menos significativo (reservamos la posición "0" para el bit de signo), y el bit 31-ésimo al más significativo; además, para homogeneizar el formato de las respuestas todos los vectores se rellenan con ceros "a la derecha" hasta completar los "1+31" bits.

[2]Por diseño, el número de ciclos de referencia a utilizar se pasa al módulo FPGA en tiempo de ejecución mediante una palabra de 5 bits, cuyo valor representa el logaritmo en base 2 del número de ciclos, *i.e.*, el número de ciclos factibles es cualquier potencia de 2 entre $2^{2^0-1} = 1$ y $2^{2^5-1} = 2^{31}$ (función "medir()" en apéndice E).

Tab. 4.5.: Intra/inter-distancias de Hamming más significativas para cada uno de los 32 canales bit estudiados.

Bit(s)	μ^{intra} (%)	μ^{inter} (%)	$\log_{10}(\text{EER})$
0	1,47	48,39	−11,654
1 - 16	0,00	0,00	–
17	0,04	1,44	−0,631
18	1,16	27,09	−5,567
19	3,76	46,05	−8,216
20	8,85	48,80	−6,032
21	20,07	49,49	−3,104
22 a 31	38,82 a 50,00	50,00 a 49,98	−0,868 a −0,267

A continuación, analizamos las propiedades de cada uno de los bit que forman parte del vector diferencial de manera independiente, construyendo sendas respuestas PUF. Este análisis apoyará la percepción intuitiva de que, en efecto, el bit de signo es la alternativa preferible para construir una respuesta en caso de emplear un único bit, al presentar este una menor volatilidad y ser, por lo tanto, significativamente repetible. Sin embargo, el análisis sistemático de cada canal individual permitirá seleccionar los bit adicionales que pueden utilizarse conjuntamente con el bit de signo para enriquecer la PUF de osciladores de anillo y mejorar sus métricas de seguridad. El conjunto de vectores diferenciales se construye utilizando una topología $\mathcal{N}_{/2}$ para evitar efectos artificiales como consecuencia del algoritmo de digitalización, tal y como se argumentó en la sección anterior. Así mismo, cada medida se ha repetido 100 veces para obtener estadística ($N^{\text{rep}} = 100$), y estas se han llevado a cabo sobre 45 instancias diferentes ($N^{\text{inst}} = 45$). En la tabla 4.5 se muestran los valores de intra/inter-distancia promedio obtenidos para cada uno de los 32 experimentos realizados (*i.e.*, para cada canal bit del vector diferencial). Tal y como cabría esperar, los bits más significativos de la diferencia varían con muy poca probabilidad dadas dos medidas de frecuencia (corresponden al "orden de magnitud" de las frecuencias características de los anillos) y, por lo tanto, las respuestas construidas con estos son constantes entre repeticiones, pero también entre diferentes instancias (bajos valores de intra/inter-distancias). En el extremo opuesto, los bits menos significativos capturan las fluctuaciones aleatorias debidas

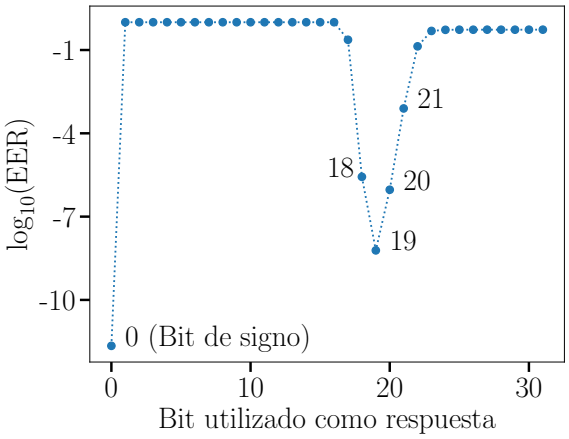


Fig. 4.16.: Tasa de igual error (EER) para las respuestas utilizando cada uno de los 32 canales bit estudiados.

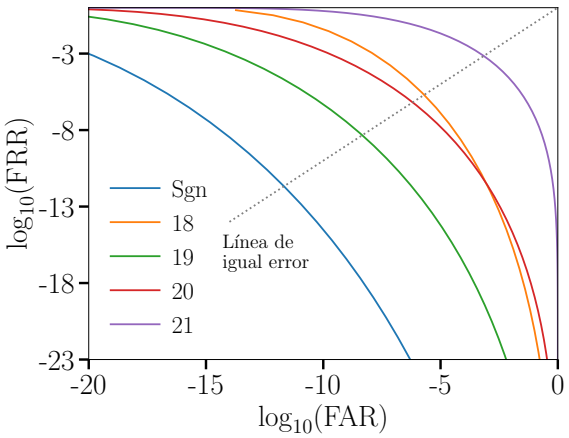


Fig. 4.17.: Curvas ROC para las respuestas generadas por los bits más prometedores, junto con la identidad (*equal error line*), que corta cada curva ROC en su correspondiente tasa de error EER. Un valor EER menor implica una mayor identificabilidad.

al ruido térmico de los anillos y, por lo tanto, su variación es máxima entre distintas instancias, pero también entre diferentes repeticiones de una misma instancia (altos valores de intra/inter-distancia). En el rango intermedio del vector binario existe una región de transición entre ambas situaciones donde los bits capturan el "ruido de fabricación", exhibiendo un buen comportamiento PUF. En la figura 4.16 se han representado las tasas de error EER para cada caso. Aquí se aprecia que la tasa de error mínima obtenida corresponde al bit de signo, sin embargo destacan al menos otros cuatro bits que exhiben buenos valores EER. En virtud de este resultado, restringiremos subsiguientes análisis a los bit correspondientes a las posiciones 18, 19, 20 y 21 del vector binario, además del bit de signo. En la figura 4.17 se han representado las curvas ROC para cada caso individual, junto con la recta identidad

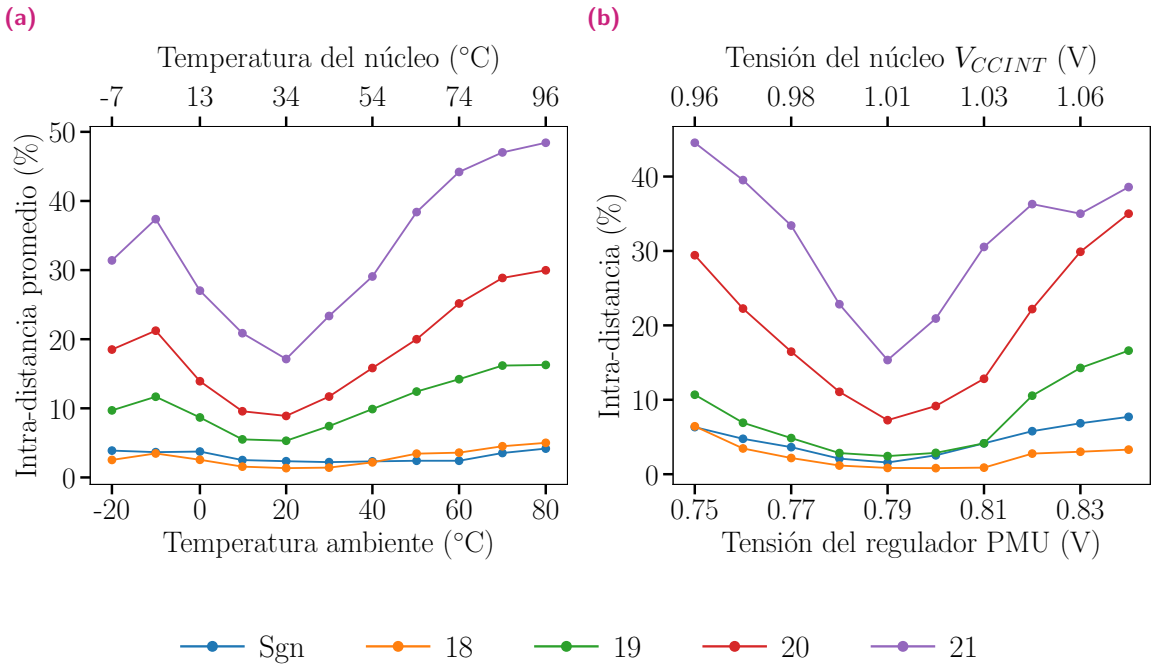


Fig. 4.18.: Análisis "V" para las respuestas generadas por los bits más prometedores: (a) variaciones de temperatura, (b) variaciones en la tensión de alimentación.

("*equal error line*"), la cual interseca cada curva ROC en la tasa EER. Las curvas características de recepción-operación caracterizan completamente las propiedades de identificabilidad de un sistema PUF, y permiten comparar rápidamente sistemas alternativos a través de la regla heurística de que un sistema es preferible cuanto más "abajo y a la izquierda" se encuentre su correspondiente ROC. También se ha llevado a cabo un experimento para estimar la robustez de las propuestas construidas con estos bits frente a variaciones del entorno de operación (temperatura y tensión), en los mismos términos descritos en la sección 4.3.5. Las curvas "V" para cada caso se muestran respectivamente en las figuras 4.18a y 4.18b. En estas se observa una degradación muy notable para las propuestas de los canales bit 20 y 21, que superan ampliamente las intra-distancias promedio del 20 % frente a variaciones de temperatura y/o tensión. Por el contrario, los bits 18 y 19 se postulan como buenos candidatos para un análisis más exhaustivo, para lo cual realizamos una nueva batería de experimentos donde construimos las respuestas PUF de forma conjunta, *i.e.*, combinando bits en parejas: "signo+18", "signo+19", "18+19", o el trío "signo+18+19". En general, la identificabilidad mejorará monótonamente con el uso combinado de un número sucesivamente mayor de bits, sin embargo, para la combinación de más de tres bits dicha mejora es sólo marginal, y no justifica la complejidad del sistema de medida asociado a una respuesta PUF excesivamente larga (sección 3.4). En la figura 4.19 se muestran las curvas ROC para estas

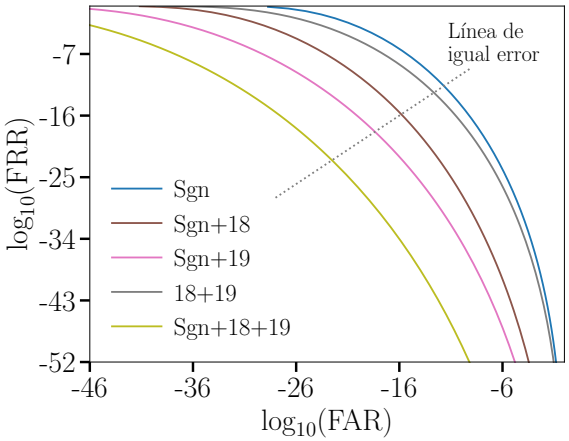


Fig. 4.19.: Curvas ROC para las respuestas generadas por las combinaciones de bits más prometedoras, junto con la identidad (*equal error line*).

combinaciones de bits, donde se pone de manifiesto la notable mejora en las propiedades de identificabilidad de este sistema, destacando en particular las propuestas "signo+19" y "signo+18+19", que con unas tasas EER respectivas de $\sim 10^{-19}$ y $\sim 10^{-23}$ mejoran los resultados de la PUF basada en osciladores de anillo estándar (EER $\sim 10^{-12}$) entre 7 y 11 órdenes de magnitud. Finalmente, este estudio se ha complementado con un análisis de la sensibilidad a las condiciones ambientales de la RO-PUF conjunta; en la figura 4.20 se muestran las curvas "V" para las variaciones de temperatura y tensión exhibidas por este sistema. Como cabría esperar, estas curvas muestran una degradación de la reproducibilidad para valores sucesivamente más alejados de las condiciones ambientales de referencia, con especial incidencia en las propuestas "signo+19" y "18+19" en el caso de temperaturas altas, y de "signo+19" en el caso de tensiones elevadas: en todos estos casos las intra-distancias medidas superan el $10\,\%$, lo cual puede comprometer la fiabilidad de estos sistemas en condiciones donde la variabilidad ambiental sea un factor determinante. En el extremo diametralmente opuesto se encuentra la solución "signo+18", que presenta una alta resistencia a la variabilidad ambiental, incluso superior al bit de signo utilizado individualmente en el caso de variaciones en la tensión de alimentación; por ello, esta será la opción preferible en entornos de operación con grandes variaciones ambientales, o en aplicaciones donde una alta reproducibilidad sea crucial (por ejemplo, si resulta imposible destinar recursos a un sistema de corrección de errores). En cualquier caso, la respuesta combinada "signo+18+19" presenta unos buenos resultados de resistencia ambiental, con intra-distancias que no superan el $10\,\%$ en ningún caso respecto de las condiciones de referencia.

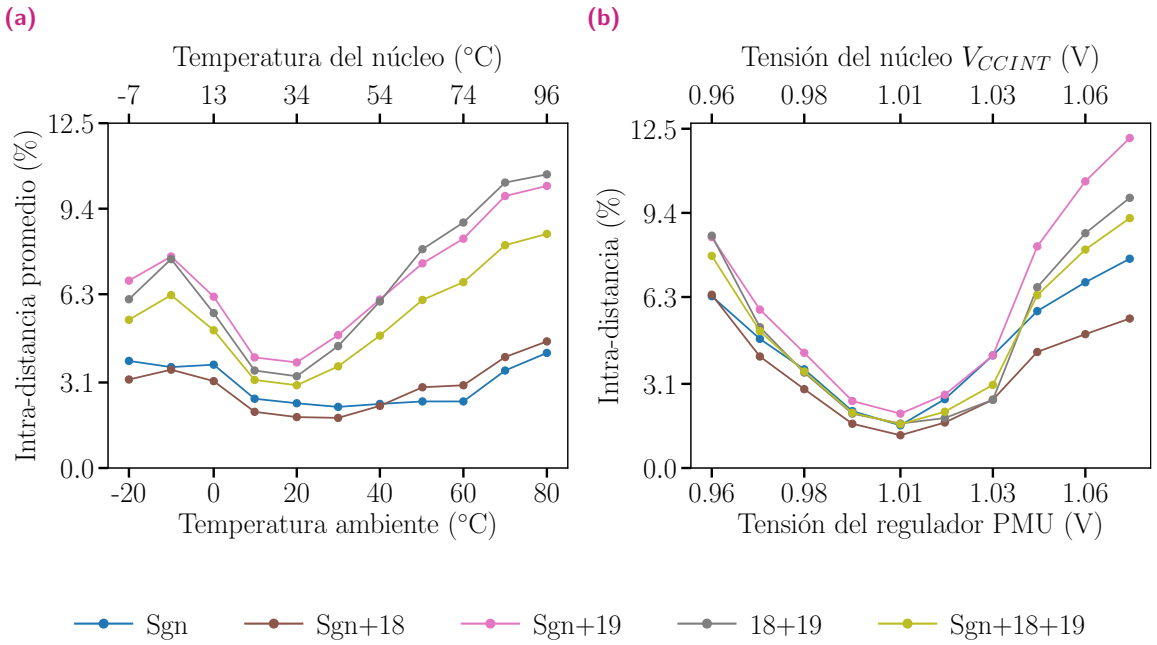


Fig. 4.20.: Análisis "V" para las respuestas generadas por las combinaciones de bits más prometedoras: (a) variaciones de temperatura, (b) variaciones en la tensión de alimentación.

4.5. Líneas de retardo programables aplicadas a RO-PUF

En la figura 4.21 se ilustra el concepto de "líneas de retardo programables" (*Programmable Delay Lines*, PDL) en FPGA. En este esquema se representa una LUT de tres entradas y una salida tal que los bits de la memoria SRAM han sido seleccionados para implementar la inversión de la entrada I_0 (función lógica NOT); los valores binarios en los puertos I_2 e I_1 son irrelevantes desde el punto de vista lógico, sin embargo, físicamente la elección de estos bits sí introduce una diferencia ya que determina la celda SRAM física a la que accede la LUT, así como la ruta que conecta dicha celda con la salida del bloque. Esto dará lugar a variaciones físicas que serán de hecho lo suficientemente significativas como para ser medidas y discernidas del ruido térmico, dando lugar a un potencial comportamiento PUF. En [132], Habib *et al.* fueron pioneros en el uso de esta técnica para explotar la estructura interna de las tablas de búsqueda de una FPGA, mejorando la entropía (*i.e.*, el número de bits independientes) de una PUF basada en osciladores de anillo.

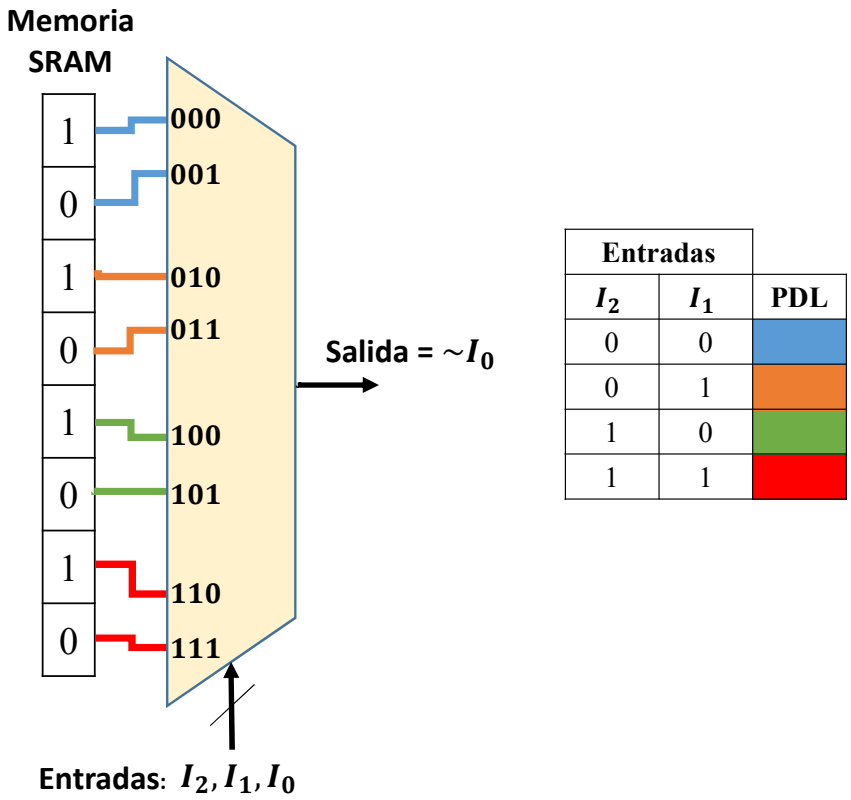


Fig. 4.21.: Representación esquemática de una LUT de tres entradas actuando como inversor de la entrada I_0. Se han destacado en diferentes colores cada una de las distintas PDL configurables físicamente mediante los pines I_2 e I_1.

El modelo FPGA Artix 7 de Xilinx utilizado en esta tesis consta de seis entradas, etiquetadas como A1 a A6, lo cual deja un máximo de cinco puertos sin utilizar para generar $2^5 = 32$ respuestas diferentes, en un esquema de oscilador de anillo como el mostrado en la figura 4.22, donde las i-ésimas entradas de cada etapa se conectan entre sí para maximizar la variabilidad inducida en cada LUT por medio de las líneas de retardo [148]. El *software* Vivado permite al diseñador asignar manualmente la relación entre puertos de entrada lógicos (I_i, $0 \leq i \leq 5$) y físicos (A_j, $1 \leq j \leq 6$) a nivel de LUT mediante el comando "LOCK_PINS". Para estudiar sistemáticamente el impacto de los puertos utilizados como PDL en las propiedades de seguridad de la RO-PUF emergente, utilizaremos los mismos cinco puertos en todos los inversores de cada anillo para generar las líneas de retardo, de forma que existen seis posibles diseños diferentes correspondientes a utilizar cada uno de los A1 - A6 pines de entrada de cada etapa para capturar la salida del inversor precedente. Llamaremos "propagador" a este sexto pin utilizado para transmitir la señal a lo largo del oscilador de anillo. Estos casos de diseño han sido ilustrados en las figuras 4.23a y 4.23b, donde se representan esquemáticamente sendas LUT utilizando respectivamente los puertos A1 y A6 como propagadores, así como se

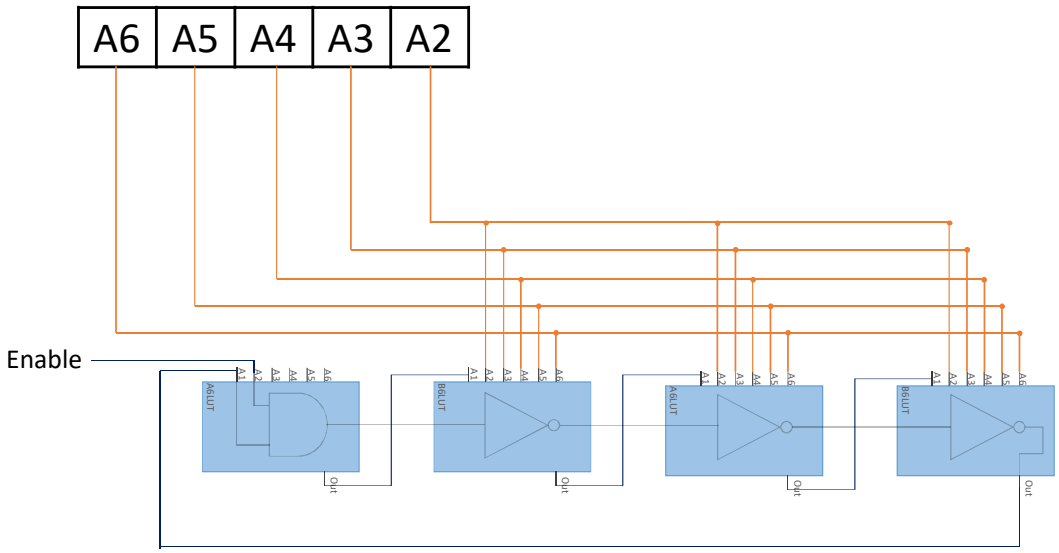


Fig. 4.22.: Esquema conceptual de oscilador de anillo de tres etapas en FPGA, donde se ilustra la disposición de las entradas para la configuración de PDL en cada una de las LUT inversoras del oscilador.

destacan los cinco puertos restantes empleados para configurar cada PDL. Dado uno de estos diseños, construimos las curvas de respuesta de un oscilador midiendo sus frecuencias características para cada configuración PDL (figura 4.24). De este modo, cada oscilador proporcionará 32 medidas de frecuencia diferentes en lugar de un único valor, lo cual permite obtener 32 bits de cada pareja de anillos mediante la comparación de las frecuencias para cada PDL respectivo. No obstante, se puede dar el fenómeno de que las variaciones de frecuencia inducidas por las líneas de retardo no sean lo bastante grandes como para generar bits diferentes durante la comparación, *e.g.*, en la figura 4.25a se han representado las curvas de frecuencia frente al índice PDL para dos osciladores diferentes: en esta imagen se aprecia que la medida de frecuencia para cualquier PDL es sistemáticamente superior en uno de los osciladores, de modo que la comparación siempre dará lugar al mismo valor binario. Esta diferencia se debe a una componente constante en las curvas que eleva una respecto a la otra, y puede eliminarse tomando la diferencia entre puntos sucesivos de la curva (*i.e.*, la derivada), dando lugar a un comportamiento mucho más rico, tal y como se muestra en la figura 4.25b, donde se representan las diferencias de frecuencia entre puntos sucesivos con condiciones periódicas (esto es, considerando que los puntos “0” y “31” son contiguos). Esta aproximación fue propuesta por Zhang *et al.* [149]; sin embargo, desde el punto de vista del diseño, el rendimiento de la propuesta de Zhang (y otros trabajos donde se emplean PDL para mejorar las PUF de osciladores de anillo configurables) es extremadamente sensible al puerto de entrada de la LUT utilizado para construir los inversores. En la figura 4.26 hemos

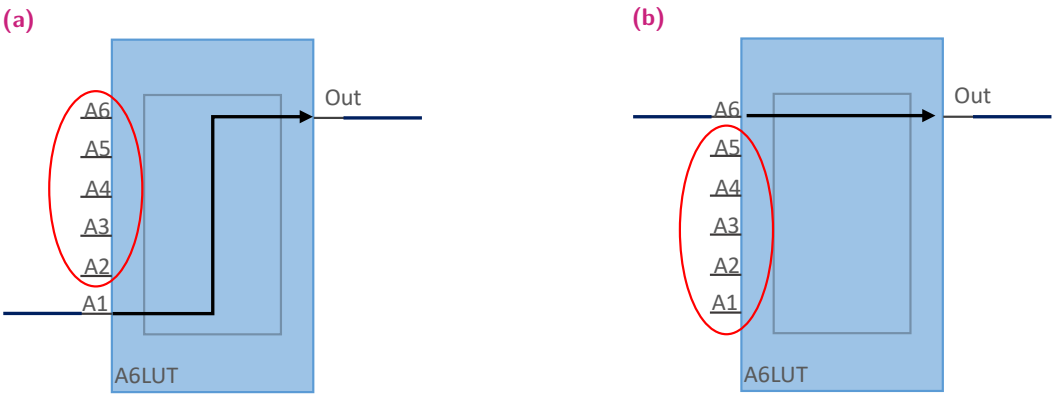


Fig. 4.23.: (a) Inversor implementado en una LUT utilizando el puerto A1 como propagador. (b) Ídem, utilizando el puerto A6 como propagador. En rojo se han destacado los puertos utilizados para configurar las PDL en cada caso.

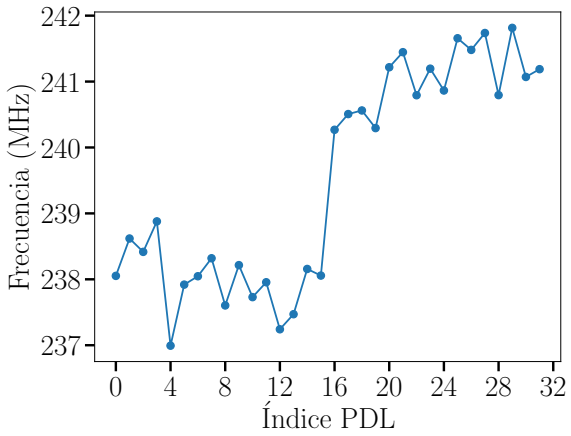


Fig. 4.24.: Curva frecuencia-PDL obtenida al medir la frecuencia característica de un oscilador de anillo para cada posible línea PDL.

representado la derivada de la frecuencia respecto del índice PDL para los casos extremos en los que se utilizan los puertos A1 y A6 como propagadores de la señal a la hora de implementar los inversores de cada oscilador, de forma que se utilizan los puertos A2 a A6 y A1 a A5 respectivamente para configurar las PDL [150]. Estas curvas muestran una variación radicalmente diferente en cada caso.

A continuación, analizaremos las prestaciones de RO-PUF configurables mediante PDL para cada uno de los seis posibles puertos de entrada LUT que pueden utilizarse como propagador en los inversores que constituyen los anillos.

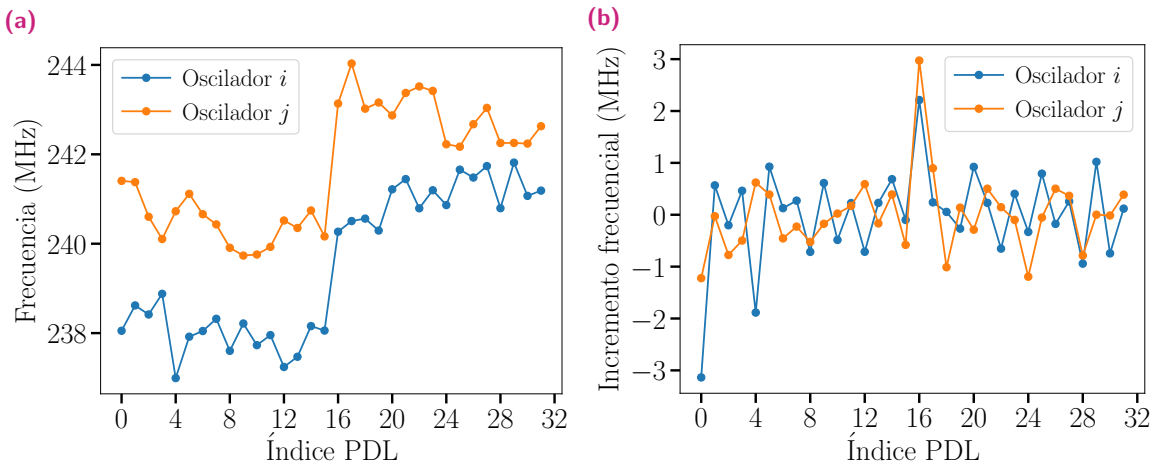


Fig. 4.25.: Medida de dos osciladores de anillo empleando la entrada LUT A1 y configurados mediante PDL: (a) frecuencia de oscilación, (b) derivada de la frecuencia respecto del índice PDL.

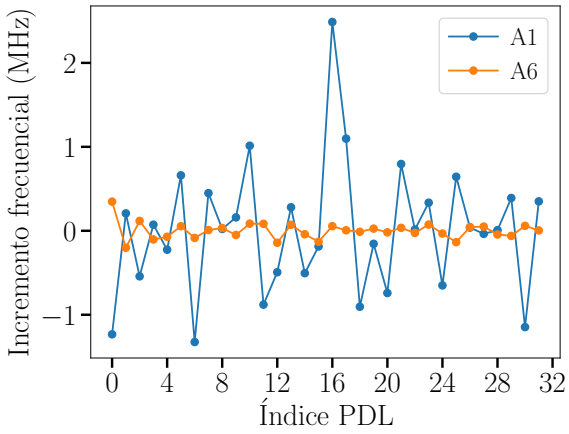


Fig. 4.26.: Incremento frecuencial respecto del índice de selección PDL para un oscilador de anillo. Se muestran conjuntamente dos diseños diferentes para ilustrar la magnitud de las variaciones inducidas en cada caso.

4.5.1. Experimentos

De acuerdo con las conclusiones alcanzadas en la sección 4.3, nos restringimos a un dominio *Naif-específico* constituido por *slice* (1), LUT (L) para implementar un pequeño conjunto de 16 osciladores de anillo de tres etapas, repitiendo este diseño en 28 instancias FPGA diferentes. Si la respuesta PUF se obtuviera digitalizando las medidas de frecuencia de acuerdo con la topología $\mathcal{N}_{/2}$, se producirían respuestas binarias de 8 bits (de escaso interés criptográfico) en el caso de una PUF basada

en osciladores de anillo estándar. A pesar del pequeño tamaño de la matriz de osciladores, la entropía total extraída puede maximizarse mediante esquemas de comparación más complejos tal y como se discutió en el capítulo 3, pero como hemos demostrado en la sección anterior, esto daría lugar a correlaciones entre bits indeseadas, de forma que nos atendremos al esquema simple de comparación sin repetición. En este caso, dadas las distintas configuraciones de cada oscilador mediante PDL, el tamaño de las respuestas se incrementará en $2^5 \times 8 = 256$ bits. El *hardware* para la interfaz entre la matriz de anillos y el sistema de medida es idéntico al utilizado en los experimentos anteriores (apéndice D), donde ahora hemos ampliado el vector de datos de entrada con cinco bits adicionales para seleccionar cada una de las 2^5 posibles líneas PDL. Para garantizar que los osciladores de cada pareja son idénticos entre sí por diseño, estos se implementan en bloques lógicos configurables inmediatamente sucesivos, de forma que los recursos *hardware* pueden replicarse de forma exacta, minimizando así las posibles correlaciones espaciales entre anillos. Así mismo, el ruteado y posicionamiento de las celdas que componen cada anillo ha sido llevado a la práctica por medio de *scripts* en lenguaje TCL, lo cual garantiza que todos los osciladores son efectivamente idénticos por diseño. Para estudiar la calidad de las PUF resultantes en función de qué salida LUT se utiliza como propagador para los inversores (o equivalentemente, qué cinco de las seis posibles entradas se utilizan para configurar cada PDL), llevamos a cabo la evaluación de un experimento PUF caracterizado por $N^{\text{retos}} = 1$, $N^{\text{inst}} = 28$, $N^{\text{rep}} = 100$ sobre seis diseños diferentes, cada uno de los cuales se distingue por utilizar una entrada LUT distinta como propagador, y que etiquetaremos utilizando el nombre de dicha entrada: A1, A2, A3, A4, A5 o A6.

A continuación, se exponen y discuten los resultados obtenidos en estos experimentos, en el que medimos la respuesta de 28 instancias diferentes, repitiendo cada medida 100 veces para obtener estadística.

4.5.2. Resultados

Como en los casos anteriores, la respuesta PUF se construye en posprocesado a partir de las medidas de frecuencia, en este caso comparando la derivada de la curva frecuencia-PDL para parejas de osciladores contiguos. Las respuestas resultantes constan de $2^5 \times 8 = 256$ bits, lo cual supone una notable densidad de información dada la limitada cantidad de recursos *hardware* empeñados en el diseño. Téngase en

Tab. 4.6.: Comparativa de la desviación estándar promedio del incremento frecuencial debida al ruido aleatorio frente a la selección PDL.

Puerto de entrada	σ^{gauss} (MHz)	σ^{PDL} (MHz)	Q
A1	0,019	0,89	46,84
A2	0,021	1,01	48,10
A3	0,024	0,79	32,92
A4	0,026	0,93	35,77
A5	0,040	1,80	45,00
A6	0,041	0,13	3,17

Tab. 4.7.: Intra/inter-distancia de Hamming promedio.

Puerto de entrada	μ^{intra} (%)	μ^{inter} (%)
A1	1,90	50,04
A2	1,71	50,12
A3	23,67	50,18
A4	23,72	49,73
A5	27,79	49,75
A6	28,43	50,03

cuenta que una clave de 128 bits se considera un estándar criptográfico, adecuado para muchas aplicaciones de seguridad [151].

En la tabla 4.6 se muestran las variaciones del incremento frecuencial debidas a diferentes PDL, σ^{PDL}, en comparación con el ruido aleatorio, σ^{gauss}, junto con el cociente entre ambas cantidades, $Q \equiv \sigma^{\text{PDL}}/\sigma^{\text{gauss}}$; estos resultados evidencian el hecho de que modificar la configuración de la LUT induce grandes variaciones en la frecuencia (hasta 48 veces mayores que el ruido aleatorio), excepto en el caso de utilizar la entrada A6 como propagador, donde las variaciones debidas a PDL pueden resultar imposibles de discernir del ruido. Esto queda confirmado por los resultados mostrados en la tabla 4.7, donde hemos calculado la intra/inter-distancia de Hamming promedio (*i.e.*, reproducibilidad y unicidad) para cada uno de los

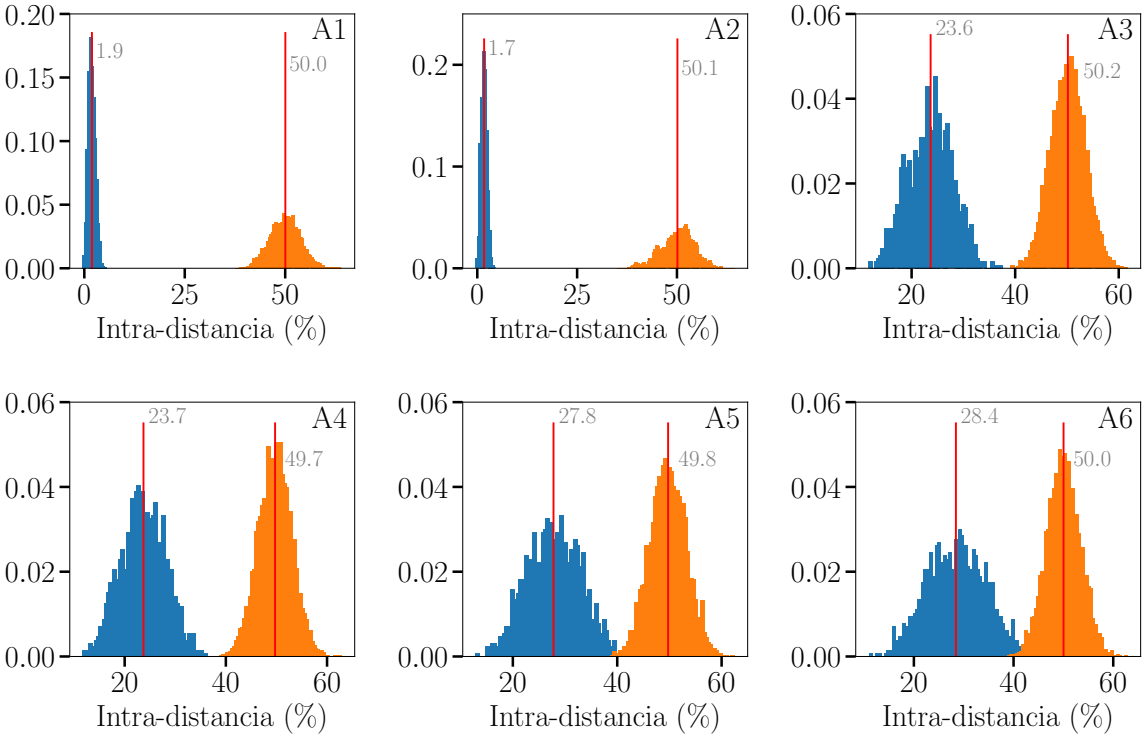


Fig. 4.27.: Distribución de la intra- (azul) e inter- (naranja) distancias de Hamming para cada una de las implementaciones estudiadas A1 - A6.

seis experimentos llevados a cabo. Este resultado muestra que las inter-distancias se encuentran todas próximas al 50 % ideal esperado en un experimento PUF; sin embargo, sólo los casos donde se utilizan las entradas A1 o A2 muestran una intra-distancia comparable al caso ideal del 0 %. En la figura 4.27 se representan los histogramas de intra/inter-distancia para cada caso estudiado A1 a A6; estas gráficas evidencian que el comportamiento PUF está fuertemente influenciado por la elección de diseño de qué entrada LUT emplear para construir el oscilador de anillo: los casos A1 y A2 exhiben excelentes propiedades de unicidad y reproducibilidad, mientras que los restantes casos A3 a A6 muestran una mala calidad en las respuestas y una alta probabilidad de cometer errores de falso rechazo y/o falsa aceptación.

Los resultados que arroja este estudio permiten concluir que la entrada LUT utilizada para construir los anillos influye de forma significativa en la calidad de la RO-PUF resultante, principalmente en cuanto a su medida de reproducibilidad, ya que esta se ve determinada por la magnitud de las variaciones inducidas al elegir diferentes líneas de retardo en comparación con el ruido térmico. En este sentido, utilizar las entradas A1 o A2 de la LUT para rutear los osciladores de anillo, manteniendo el resto de entradas libres para configurar las PDL, proporciona

un resultado óptimo en cuanto a unicidad y reproducibilidad. Por otra parte, esta sección pone de manifiesto una clara degradación de las métricas PUF al utilizar las entradas A3, A4, A5 o A6 como propagadores; esto obliga a extremar la precaución para evitar que el *software* EDA emplee estos puertos de manera automática en el diseño de RO-PUF, particularmente en un flujo de diseño idiosincráticamente ligado a la automatización como es el diseño digital.

4.6. Conclusión

En este capítulo se han propuesto varias estrategias de diseño e implementación de funciones no-clonables físicamente basadas en osciladores de anillo que mejoran significativamente el estado de la técnica actual. Estas han sido caracterizadas experimentalmente en prototipos FPGA, con especial atención a sus propiedades de identificabilidad. Específicamente, en primer lugar se ha demostrado que la calidad de la PUF puede verse muy afectada dependiendo de ciertos parámetros del diseño como la ubicación de los osciladores en la FPGA, su ruteado, o el tipo de recursos *hardware* utilizados para su implementación. En este sentido, se han identificado los grados de libertad de diseño más relevantes para las métricas de calidad PUF, destacando una configuración espacial óptima para el sistema RO-PUF sobre FPGA. Además, se han explorado dos vías alternativas para la extracción de entropía en una matriz de osciladores de anillo. Por un lado, la propuesta de un esquema de medida compensada ampliado, denominado “medida compensada de orden superior”, el cual es capaz de aumentar el número de bits de la respuesta sin generar un impacto significativo en el consumo de recursos *hardware* empleados por la solución estándar. Por otra parte, se ha llevado a cabo un análisis exhaustivo de las funciones no-clonables físicamente basadas en osciladores de anillo configurables mediante líneas de retardo programables, con especial énfasis en su diseño e implementación sobre la plataforma FPGA de bajo consumo empleada en este trabajo, y que resulta adecuada para el despliegue en un ecosistema IoT. En este sentido, se ha demostrado que, mediante la configuración adecuada de los puertos físicos de la LUT, esta técnica permite obtener respuestas binarias de al menos 256 bits empleando una matriz reducida de 16 osciladores de anillo.

5 Funciones no-clonables físicamente basadas en osciladores de anillo de Galois

En este capítulo estudiamos la posibilidad de utilizar osciladores de anillo de Galois para construir funciones no-clonables físicamente (*Galois Ring Oscillator PUF*, GARO-PUF) sobre FPGA [152]. Un oscilador de anillo de Galois es una arquitectura digital que puede ser utilizada en el diseño de generadores de números verdaderamente aleatorios (TRNG), adecuada para su implementación en plataformas FPGA. Un defecto característico de estos sistemas al actuar como TRNG es la existencia de un sesgo sistemático, *i.e.*, una cierta tendencia a generar bits con una razón $1/0$ distinta del $50\,\%$ [153], [154]. En este capítulo se propone la hipótesis de que la sistematicidad de este defecto es debida a variaciones físicas microscópicas en los anillos y, por lo tanto, resulta en una magnitud única, repetible y no-clonable físicamente, adecuada para la construcción de un sistema PUF, cuya principal ventaja respecto de los osciladores de anillo estándar radica en una menor correlación espacial. Además, estas estructuras PUF han sido diseñadas para generar secuencias binarias aleatorias como parte de su proceso de evaluación, lo cual es una etapa habitual en la ejecución de muchos protocolos criptográficos que incluyen PUF [155], [156], permitiendo combinar potencialmente ambas primitivas PUF y TRNG en una única estructura [157].

5.1. Osciladores de anillo de Galois

En 2006, J. D. Golić propuso un método para la generación de números verdaderamente aleatorios utilizando unas estructuras digitales oscilantes novedosas: anillos de Fibonacci (FIRO), y anillos de Galois (GARO) [158]. Estas se basan en el diseño del oscilador de anillo estándar, el cual se modifica incorporando una serie de

puertas lógicas XOR al lazo de realimentación de modo análogo a las configuraciones de Fibonacci y Galois en un registro de desplazamiento realimentado linealmente (*Linear Feedback Shift Register*, LFSR) (figura 5.1). Tanto para los osciladores de

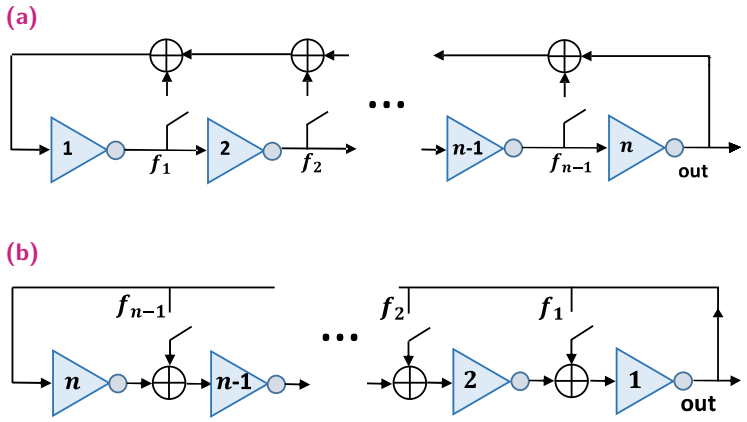


Fig. 5.1.: Esquema de: (a) oscilador de anillo de Fibonacci, (b) oscilador de anillo de Galois.

Fibonacci como de Galois, las conexiones de realimentación se especifican mediante unos coeficientes binarios $f_i \in \{0, 1\}$, de tal forma que la configuración de cada anillo se define unívocamente mediante un polinomio binario $f(x) = 1 + \sum_{i=1}^{n} f_i x^i$, $f_n = 1$. De este modo, $f_i = 1$ representa el i-ésimo interruptor cerrado de los osciladores de la figura 5.1, mientras que $f_i = 0$ representa el mismo interruptor abierto.

5.1.1. Implementación en FPGA

Los osciladores de Galois (figura 5.1b) tienen la ventaja práctica respecto de sus homólogos de Fibonacci (figura 5.1a) de que los primeros se pueden implementar en FPGA utilizando n LUT para un polinomio de orden n, mientras que los segundos necesitan $2n - 1$ elementos para una estructura equivalente. En la figura 5.2 se ha representado esquemáticamente un anillo GARO tal y como han sido diseñados en FPGA: la n-ésima etapa se implementa como un inversor, y las $n - 1$ fases restantes están constituidas por LUT de tres entradas configuradas para realizar la función lógica:

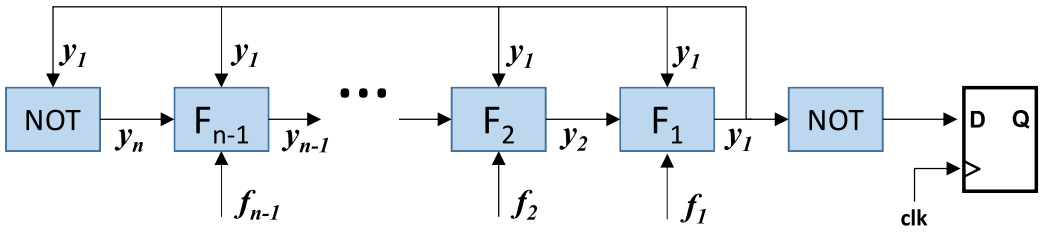


Fig. 5.2.: Esquema de un oscilador de anillo de Galois implementado en FPGA, utilizando $n+1$ LUT y un *flip-flop*.

f_i	$y_i = \mathrm{F}_i\,(f_i, y_{i+1}, y_1)$
0	**NOT**(y_{i+1})
1	**XNOR**(y_{i+1}, y_1)

las cuales implementan una función NOR-exclusiva (XNOR) si $f_i = 1$, o un inversor (NOT) si $f_i = 0$. A continuación de la última etapa del anillo se coloca un inversor adicional para mitigar la diafonía (*crosstalk*) entre osciladores cercanos, y la salida de este conjunto se lleva a la entrada de datos de un *flip-flop*, que se encarga de muestrear la señal procedente del anillo con cada flanco ascendente del reloj de muestreo, "clk". De esta manera, en la salida "Q" del registro se obtiene un bit que varía de forma verdaderamente aleatoria con cada flanco ascendente de "clk", siempre que se observen algunas restricciones tanto del anillo como de la tasa de muestreo:

- El periodo de muestreo debe ser lento en comparación con el tiempo de propagación característico del bucle de realimentación, de forma que no se lleven a cabo varios muestreos sucesivos en un intervalo de tiempo inferior al requerido por el anillo de Galois para modificar el bit de salida.

- El polinomio LFSR no debe coincidir con un punto fijo que conduzca a un bit constante a la salida del anillo.

En [158] se demuestra que un anillo de Galois carece de un punto fijo si n es impar y su polinomio característico se puede escribir como $f(x) = (1+x)h(x)$, y consta de un ciclo de periodo máximo si además el polinomio $h(x)$ es un polinomio primitivo[1] del cuerpo $\mathrm{GF}(2^n)$, *i.e.*, la extensión a orden n del cuerpo binario $\mathbb{B} \equiv \mathrm{GF}(2)$. Utilizando una tabla de polinomios primitivos para el cuerpo finito $\mathbb{B}$ [160], podemos

[1] No obstante, Su *et al.* demuestran en [159] que este criterio es superfluo para la generación de secuencias verdaderamente aleatorias en un oscilador de Galois; en su lugar, estas se deberían suficientemente al ruido aleatorio (*jitter*) de los propios osciladores que forman el anillo.

escribir algunos de los polinomios característicos para anillos de 3, 5 y 7 etapas como:

n	$h(x)$	$f(x) = (1+x)h(x)$
3	$1 + x + x^2$	$1 + x^3$
5	$1 + x + x^4$	$1 + x^2 + x^4 + x^5$
7	$1 + x + x^6$	$1 + x^2 + x^6 + x^7$

Donde se debe notar que los coeficientes de estos polinomios pertenecen al cuerpo $\mathbb{B}$, *e.g.*, $1 + 1 = 0$.

Para implementar cada uno de estos osciladores empleamos bloques lógicos configurables sucesivos en una misma columna de la FPGA, utilizando ambas celdas *slice* (0) y *slice* (1) del bloque lógico configurable (2.4.1) cuando el tamaño del anillo lo requiera. Cada etapa del oscilador se implementa sobre una LUT diferente, y la salida del conjunto se envía al *flip-flop* más próximo presente en la misma celda (véase la figura 2.12 de la sección 2.4.1). A excepción de la primera etapa, que implementa una función lógica NOT e invierte la señal procedente del bucle de realimentación, las $n - 1$ fases restantes se diseñan como LUT de tres entradas: la primera se conecta con la salida de la LUT inmediatamente precedente para formar el anillo, la segunda se cortocircuita con el bucle de realimentación, y la entrada restante se hace accesible externamente para enviar el bit f_i a la i-ésima etapa, permitiendo configurar el polinomio de Galois en tiempo de ejecución.

Con el fin de estimar una frecuencia de muestreo óptima, en la figura 5.3 se ha representado la autocorrelación normalizada $R(L) \equiv \sum_i y_i y_{i+L}$, siendo y_i el i-ésimo bit de la secuencia y L la longitud de correlación[2], para diferentes frecuencias de muestreo ν_s, utilizando un GARO de 7 etapas configurado con el polinomio $f(x) = 1 + x + x^6 + x^7$. En esta figura se aprecia cómo las curvas de autocorrelación evolucionan hacia un valor estacionario tanto más rápidamente cuanto menor es la frecuencia de muestreo. Utilizando esta propiedad, diseñamos una matriz de 100 anillos GARO de cada uno de los cuales extraemos una secuencia aleatoria y calculamos su curva de autocorrelación como una función de la frecuencia de muestreo,

[2]Notar que $R(L = 0)$ coincide con el sesgo de la secuencia, $R(0) = \mathrm{Prob}(y = 1)$. En efecto, si denotamos el sesgo $b \equiv \mathrm{Prob}(y = 1)$, para una secuencia suficientemente grande podemos reordenar la suma $R(0)$ en dos términos, correspondientes a los eventos $y_i = 1/0$, como:

$$R(0) = \frac{1}{N}\left(\sum_{i=1}^{bN} y_i = 1 + \sum_{i=1}^{(1-b)N} y_i = 0\right) = \frac{1}{N}\sum_{i=1}^{bN} 1 = \frac{bN}{N} = b$$

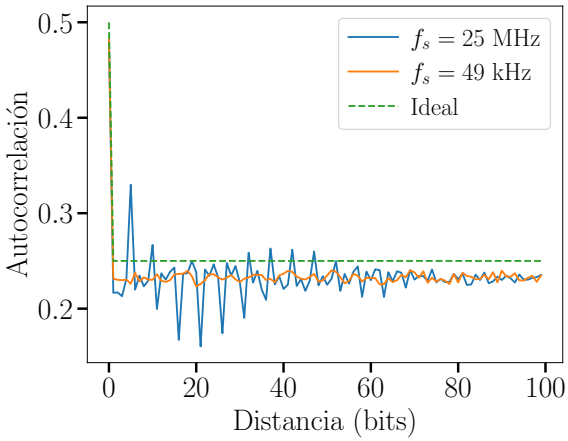


Fig. 5.3.: Curvas de autocorrelación para diferentes frecuencias de un oscilador de anillo de Galois de 7 etapas implementado en FPGA.

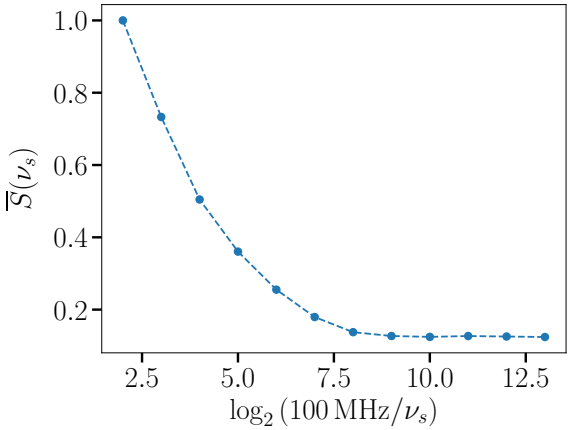


Fig. 5.4.: Área promedio contenida bajo la curva de autocorrelación de 100 anillos de Galois implementados en FPGA, frente a la frecuencia de muestreo; un valor inferior representa una longitud de autocorrelación promedio menor.

$R_i(L, \nu_s)$, siendo el índice "i" representa el i-ésimo oscilador de la matriz. A continuación, definimos el estadístico $S_i \equiv \sum_L [R_i(L) - R_i(\infty)]^2$, donde la cantidad $R(\infty)$ puede calcularse a partir de la autocorrelación bajo la conjetura de que para distancias suficientemente grandes, $L \gg 1$, la correlación entre bits será nula en el sentido de que $\mathrm{Prob}(y_{i+L}|y_i) = \mathrm{Prob}(y_{i+L})$. Esto nos permite escribir la probabilidad conjunta $\mathrm{Prob}(y_i, y_{i+L}) = \mathrm{Prob}(y_i)\mathrm{Prob}(y_{i+L})$, y calcular el valor esperado de la autocorrelación a grandes distancias como $\overline{R(\infty)} = \sum_i [\mathrm{Prob}(y_i = 1)\mathrm{Prob}(y_{i+L} = 1)] = R(0)^2$. En la figura 5.4 se representa la curva $\overline{S}(\nu_s)$ promediada para los 100 anillos implementados frente a la frecuencia de muestreo[3]. Esta figura muestra un beneficio

[3]De hecho, el eje de abscisas corresponde a la función $\log_2\left(\frac{100\,\mathrm{MHz}}{\nu_s}\right)$, ya que el reloj de muestreo se ha implementado en *hardware* como un divisor de reloj, y por tanto es natural representar esta magnitud como el factor divisor de una frecuencia de referencia dada.

sostenido en el uso de frecuencias de muestreo inferiores hasta aproximadamente $100\,\mathrm{kHz}$, *i.e.*, $\log_2 100\,\mathrm{MHz}/\nu_s = 100\,\mathrm{kHz} \approx 10$, valor a partir del cual la ganancia resulta marginal. De acuerdo con este resultado, en lo sucesivo y salvo que se especifique explícitamente en contrario, utilizaremos una frecuencia de muestreo $\nu_s = 100\,\mathrm{MHz}/2^{10} \approx 97{,}65\,\mathrm{kHz}$.

5.1.2. Propiedades de la magnitud sesgo $R(0)$ en una matriz de anillos de Galois

Un inconveniente característico de los TRNG construidos utilizando osciladores de anillo de Galois en FPGA es la existencia de un sesgo sistemático en la proporción de bits "1" a "0" producidos, que varía en función de la localización de un anillo en la matriz FPGA. Este defecto, sin embargo, resulta prometedor de cara a utilizar estos sistemas con fines de identificación y autenticación. En particular, debido a su simplicidad, estudiaremos el sesgo $R(0)$ de los anillos GARO utilizados como TRNG en función de su localización en la FPGA. Esta cantidad se podrá utilizar como magnitud característica de una PUF de medida compensada basada en las diferencias de sesgo entre los distintos anillos de Galois que constituyan una cierta matriz si cumple:

1. La distribución del sesgo es repetible en el tiempo dada una localización y una instancia (*i.e.*, un dispositivo FPGA): si se mide el sesgo de un mismo oscilador en una única FPGA para distintos instantes de tiempo, la probabilidad de que estas medidas sean cercanas, de acuerdo a una cierta métrica, debe ser alta.

2. La variabilidad del sesgo entre diferentes posiciones dentro del chip, así como en la misma posición para distintos chips, es suficiente para resolver cada anillo al menos localmente: si se mide el sesgo de un mismo oscilador en chips FPGA distintos, la probabilidad de que estas medidas sean cercanas, de acuerdo a una cierta métrica, debe ser baja.

Bajo estas condiciones es posible construir una PUF de medida compensada basada en la comparación del sesgo entre parejas de osciladores de Galois idénticos por diseño.

Para estudiar la variación del sesgo sobre distintas localizaciones de un anillo respecto de los cambios debidos al ruido estocástico al medir un mismo oscilador

Tab. 5.1.: Desviación estándar de diferentes osciladores.

	$\overline{\sigma^{\text{loc}}}$	$\overline{\sigma^{\text{rep}}}$	Q
3-GARO $f(x) = 1 + x^3$	0,0089	0,0013	7,05
5-GARO $f(x) = 1 + x^2 + x^4 + x^5$	0,021	0,0028	7,48
7-GARO $f(x) = 1 + x^2 + x^6 + x^7$	0,18	0,0030	61,67
3-RO	13,79	0,13	108,95

en distintos instantes de tiempo hemos implementado el mismo diseño GARO en un total de $N^{\text{loc}} = 100$ localizaciones distintas dentro de la FPGA, y cada medida se ha repetido $N^{\text{rep}} = 100$ veces con cada oscilador, obteniendo la batería de medidas $\{R_{ij}(0)\}_{i=1,j=1}^{N^{\text{loc}},N^{\text{rep}}}$, donde $R_{ij}(0)$ representa el sesgo medido para el i-ésimo anillo y la j-ésima repetición. Esta notación permite definir el sesgo promedio a lo largo de las localizaciones, $\overline{R_j(0)} \equiv \sum_i R_{ij}(0)/N^{\text{loc}}$, así como a lo largo del eje de repeticiones, $\overline{R_i(0)} \equiv \sum_j R_{ij}(0)/N^{\text{rep}}$, y análogamente las desviaciones:

$$\sigma_j^{\text{loc}} \equiv \sqrt{\frac{\sum_i \left(R_{ij}(0) - \overline{R_j(0)}\right)^2}{N^{\text{loc}} - 1}} \tag{5.1}$$

$$\sigma_i^{\text{rep}} \equiv \sqrt{\frac{\sum_j \left(R_{ij}(0) - \overline{R_i(0)}\right)^2}{N^{\text{rep}} - 1}} \tag{5.2}$$

Definimos las desviaciones promedio como $\overline{\sigma^{\text{loc}}} \equiv \sum_j \sigma_j^{\text{loc}}/N^{\text{rep}}$, y $\overline{\sigma^{\text{rep}}} \equiv \sum_i \sigma_i^{\text{rep}}/N^{\text{loc}}$, y finalmente introducimos un factor de calidad Q de la matriz GARO:

$$Q \equiv \overline{\sigma^{\text{loc}}}/\overline{\sigma^{\text{rep}}} \tag{5.3}$$

En la tabla 5.1 se muestran los factores de calidad $\boldsymbol{Q}$ estimados para cada uno de los polinomios descritos en la sección previa, junto con el mismo observable medido para una matriz de osciladores de anillo estándar[4] de tres etapas tal y como se introdujo en el capítulo 4 a modo de control. En todos los casos se ha obtenido una

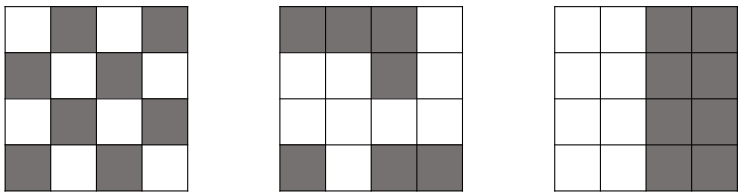
[4]En este caso utilizando la frecuencia de oscilación como respuesta física para calcular el factor de calidad.

variabilidad superior al medir los sesgos de anillos diferentes en comparación con medidas sucesivas de un mismo anillo, lo cual se puede interpretar como que el "ruido de fabricación" es superior al ruido aleatorio propio del sistema, y por tanto constituye un candidato prometedor como alternativa PUF; esto es particularmente cierto para el caso de anillos de Galois de 7 etapas, en los que la desviación promedio entre diferentes anillos es más de 60 veces superior a las fluctuaciones locales. Dados estos resultados, en lo que sigue incidiremos en el análisis de matrices de anillos de Galois de 7 etapas.

La principal ventaja de la arquitectura GARO-PUF respecto de su contrapartida basada en osciladores de anillo estándar radica en su menor correlación espacial, lo cual reduce la probabilidad de predecir el resultado de una comparación basándose en las posiciones que ocupan los anillos en el seno de la matriz FPGA. Para estudiar esta característica utilizamos el estadístico de Moran [161], I_M, el cual se define para un conjunto de datos experimentales $\{x_i\}_{i=1}^N$ como la autocorrelación ponderada por una matriz de pesos w_{ij}, con la restricción de que la suma de todos los pesos sea igual a la unidad, $\sum_{ij} w_{ij} = 1$:

$$I_M \equiv \frac{\sum_i^N \sum_j^N w_{ij} \left(x_i - \overline{x}\right) \left(x_j - \overline{x}\right)}{\sum_i^N \left(x_i - \overline{x}\right)^2 / N} \tag{5.4}$$

Esta noción generalizada de autocorrelación resulta aplicable a conjuntos de datos organizados en forma de mapas bidimensionales (u otras disposiciones de diferente dimensión), en cuyo caso la matriz de pesos se utiliza para definir el conjunto de vecinos cuya correlación cruzada se quiere estudiar (*e.g.*, una entrada $w_{ij} > 0$ implica que los datos x_i, x_j son vecinos, y en caso contrario $w_{ij} = 0$). Para el análisis llevado a cabo en esta sección hemos utilizado una matriz de primeros vecinos donde cada elemento (x, y) consta de dos vecinos, $(x + 1, y)$ y $(x, y + 1)$, sin condiciones periódicas de contorno; el índice de Moran toma valores en el intervalo $[-1, +1]$, con un promedio nulo para datos generados por una distribución aleatoria y uniforme:



$I_M = -1.0$ $I_M = 0.03$ $I_M = 1.0$

En la figura 5.5a se han representado las frecuencias características de una matriz de osciladores de anillo estándar de 7 etapas; una inspección preliminar de

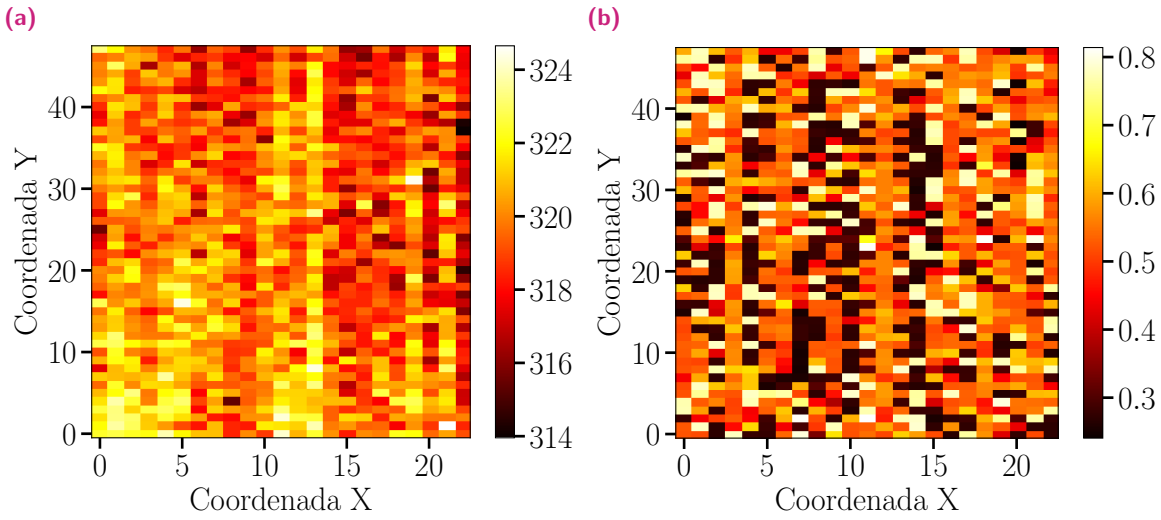


Fig. 5.5.: (a) Mapa de frecuencia característica de una matriz de osciladores de anillo estándar de 7 etapas; (b) mapa de sesgo de la misma matriz formada por anillos de Galois de 7 etapas, ubicados en las mismas posiciones que sus homólogos estándar.

este mapa expone algunos patrones de correlación evidentes, *e.g.*, los osciladores implementados en el extremo derecho de la matriz tienden a mostrar frecuencias de oscilación inferiores respecto de aquellos dispuestos en la mitad izquierda, dando lugar a un mapa con un índice de Moran asociado $I_M = 0{,}34$. Un mapa análogo ha sido obtenido para el sesgo de una matriz de anillos de Galois de 7 etapas (figura 5.5b), implementados estos en las mismas localizaciones que sus homólogos estándar, y alcanzando un valor de Moran significativamente inferior, $I_M = 0{,}028$, lo cual evidencia la menor correlación espacial de esta alternativa; esto supone una ventaja competitiva con respecto a los osciladores de anillo estándar, ya que permite extender el espacio efectivo de retos (*i.e.*, pares de osciladores) a comparar para generar una respuesta PUF, permitiendo cotas superiores de seguridad o, en su caso, niveles de confianza similares con un menor consumo de recursos.

5.2. Implementación de GARO-PUF

Se diseñan tres prototipos alternativos de funciones no-clonables físicamente basadas en osciladores de anillo de Galois de 3, 5 y 7 etapas; estos constan de una matriz de 200 anillos y producen respuestas binarias de 100 bits utilizando

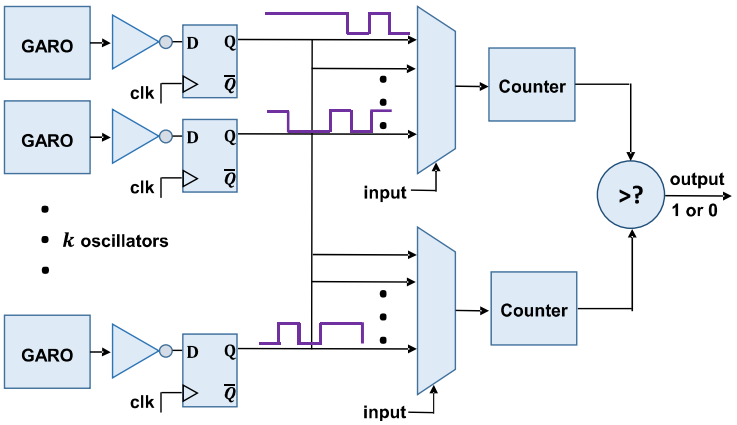


Fig. 5.6.: Esquema de una PUF basada en osciladores de anillo de Galois: GARO-PUF.

una topología de compensación $\mathcal{N}_{/2}$ (sección 3.3.2), empleando como parámetro físico el sesgo de cada anillo al ser configurado como TRNG, $\psi = R(0)$, con la estructura de PUF mostrada en la figura 5.6. La elección de esta topología se justifica reproduciendo el argumento dado en la sección 4.3.4 de que, si bien existen alternativas capaces de maximizar las propiedades de seguridad de las respuestas tal y como se estudió detalladamente en el capítulo 3, la nula influencia de esta topología en los resultados la convierte en la opción preferible para caracterizar experimentalmente el sistema.

De forma análoga a las matrices de osciladores de anillo estándar descritas anteriormente, el diseño de la matriz de anillos GARO ha sido llevado a cabo a través de un "entorno de desarrollo" escrito en lenguaje "Python", estructurado alrededor de una clase de objetos "GaloisMatrix" (apéndice E); la interfaz de usuario que presenta esta clase, así como el flujo de diseño digital, es similar a su contrapartida estándar detallada en la sección 4.2, sin embargo, la arquitectura de los anillos de Galois obliga a algunas diferencias generales respecto de los osciladores estándar, entre las cuales destacan:

- Dado el bucle de realimentación general de los anillos de Galois, no es posible dotar al oscilador de una entrada *enable* mediante una puerta inicial que corte el bucle. Pese a esto, cada anillo de N etapas consta de $N + 2$ elementos físicos, a saber, cada una de las celdas que componen el oscilador, junto con un inversor final y un *flip-flop*. Estos elementos adicionales deben ser tenidos en cuenta a la hora de inicializar un objeto "GaloisMatrix" con restricciones físicas.

- La función “medir()” del objeto “GaloisMatrix” utilizada para iniciar el proceso de medida y recopilar esta acepta la introducción de un parámetro *poly* el cual define el polinomio utilizado para configurar el oscilador, y *fdiv* que modifica la razón entre el reloj de muestreo y el reloj de referencia del sistema (implementado este último de forma inmutable en tiempo de ejecución a $100\,\mathrm{MHz}$), permitiendo variar la frecuencia de muestreo.

Además de la matriz de anillos, el diseño implementado incluye un módulo para medir el sesgo de cada oscilador en el propio chip, así como la lógica necesaria para comunicar esta información en forma de números enteros sin signo a un ordenador que construye la respuesta binaria en posprocesado.

5.2.1. Evaluación experimental

Para caracterizar el desempeño de este sistema como función no-clonable físicamente implementamos un experimento PUF sobre $N^{\text{inst}} = 40$ chips FPGA, de cada uno de los cuales extraemos $N^{\text{rep}} = 100$ respuestas binarias, utilizando un único reto que será el propio proceso de medida, $N^{\text{retos}} = 1$. Estos resultados permiten evaluar la reproducibilidad, unicidad e identificabilidad de esta propuesta PUF, tal y como se detalla a continuación.

Unicidad

La medida de la unicidad de GARO-PUF se lleva a cabo utilizando la inter-distancia de Hamming (2.89) sobre el conjunto de respuestas binarias obtenidas para cada uno de los $N^{\text{inst}} = 40$ chips FPGA diferentes utilizados en el experimento descrito anteriormente. En la figura 5.7 se han representado en color naranja los histogramas correspondientes a las distribuciones de dicha cantidad para los casos de anillos de Galois de 3, 5 y 7 etapas, donde se ha destacado el valor promedio obtenido en cada caso. Así mismo, en la tabla 5.2 se muestran las estadísticas de interés para cada caso: la inter-distancia promedio, el parámetro $\tilde{p}^{\text{inter}}$ de la distribución binomial que mejor ajusta el histograma obtenido, y la estimación del estadístico de Kolmogorov-Smirnov, $\tilde{D}_{KS}$, que mide la bondad del ajuste entre dicho histograma y la distribución binomial de parámetros $n = 100$, $p = \tilde{p}^{\text{inter}}$, y cuyos valores son consistentes con la hipótesis de que las inter-distancias están distribuidas binomialmente con una significancia $\alpha = 5\,\%$ en todos los casos.

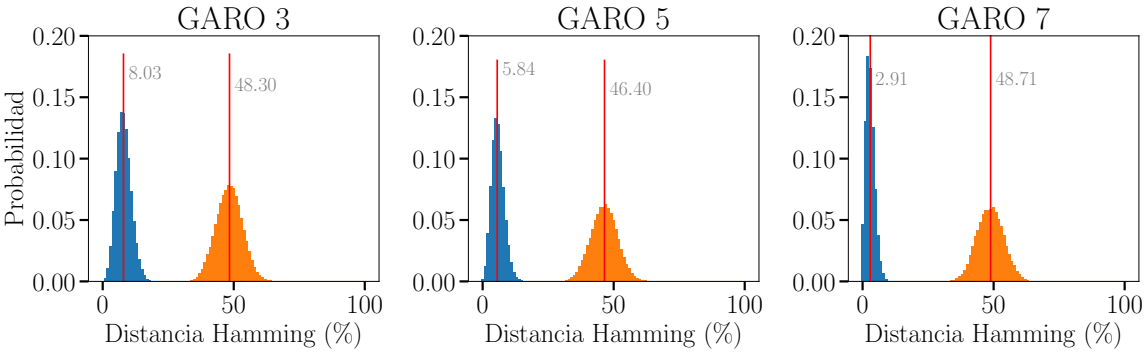


Fig. 5.7.: Distribución de la intra- (azul) e inter- (naranja) distancias de Hamming para cada una de las arquitecturas estudiadas: GARO 3, 5 y 7 etapas.

Tab. 5.2.: Resultado del análisis de unicidad llevado a cabo sobre cada arquitectura GARO estudiada.

Arquitectura	$\tilde{\mu}^{\text{inter}}$ (%)	$\tilde{p}^{\text{inter}}$	$\tilde{D}_{KS}$	$D_{KS}\|_{5\,\%}$	Test
GARO 3 $f(x) = 1 + x^3$	48,30	0,483	0,00430	0,0505	✓
GARO 5 $f(x) = 1 + x^2 + x^4 + x^5$	46,40	0,464	0,00751	0,0674	✓
GARO 7 $f(x) = 1 + x^2 + x^6 + x^7$	48,71	0,487	0,0104	0,0506	✓

Estos resultados ponen de manifiesto las excelentes propiedades de unicidad en todos los casos estudiados de PUF basadas en osciladores de Galois, obteniendo valores de inter-distancia promedio superiores al 45 %. De nuevo, destaca la propuesta de anillos de 7 etapas, cuya unicidad se aproxima al caso ideal.

Reproducibilidad

La figura de mérito para evaluar la reproducibilidad de esta propuesta PUF es la intra-distancia de Hamming (2.96). Los histogramas correspondientes a las distribuciones de intra-distancia para cada caso estudiado se han representado

Tab. 5.3.: Intra-distancia de Hamming promedio obtenidas para cada arquitectura GARO estudiada.

Arquitectura	$\tilde{\mu}^{\text{intra}}$ (%)	$\tilde{p}^{\text{intra}}$	$\tilde{D}_{KS}$	$D_{KS}\|_{5\,\%}$	Test
GARO 3 $f(x) = 1 + x^3$	8,03	0,0803	0,0133	0,0178	✓
GARO 5 $f(x) = 1 + x^2 + x^4 + x^5$	5,84	0,0584	0,0085	0,0182	✓
GARO 7 $f(x) = 1 + x^2 + x^6 + x^7$	2,91	0,0291	0,0127	0,0187	✓

en la figura 5.7 (color azul), donde se ha destacado el valor promedio obtenido. Estas cantidades se encuentran así mismo desglosadas en la tabla 5.3, junto con los polinomios utilizados en cada arquitectura, mostrando una buena reproducibilidad para todos los extremos estudiados, destacando en particular la matriz de anillos de 7 etapas, cuya intra-distancia promedio es inferior al $3\,\%$, tal y como había sido anticipado por el estudio de los factores de calidad Q asociados a cada alternativa analizados en la sección 5.1.2 (tabla 5.1).

Identificabilidad

Tal y como se discutió en la sección 2.3.3, la identificabilidad de un sistema PUF, *i.e.*, las propiedades de que este sea simultáneamente único y reproducible, se evalúa de forma integral mediante las curvas de falso rechazo (FRR) y falsa aceptación (FAR). En la figura 5.8a se muestra la superposición de ambas curvas para cada uno de los tres polinomios de Galois estudiados, donde se han destacado los puntos de intersección correspondientes a la tasa de igual error (EER) y el umbral de identificación u^{EER}; así mismo, a efectos de facilitar la comparación entre cada propuesta, se ha representado en la figura 5.8b la curva característica operativa del receptor (ROC) para cada arquitectura. Los valores obtenidos para cada una de estas métricas pueden consultarse en la tabla 5.4, donde se exponen en relación con sus correspondientes polinomios característicos. Estos resultados evidencian unas

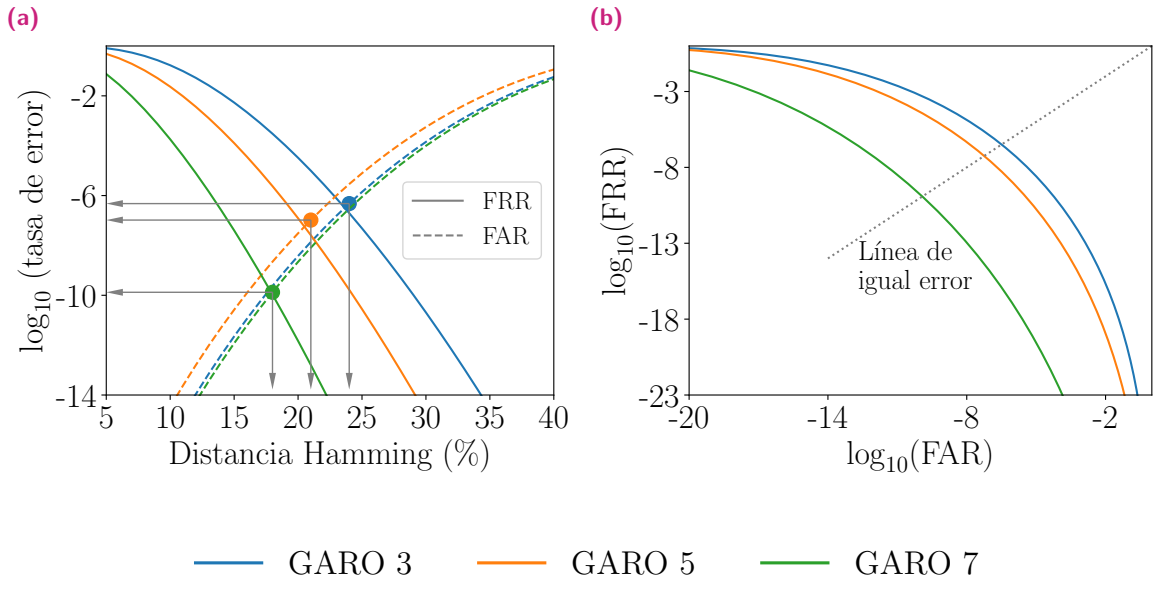


Fig. 5.8.: (a) Curvas FAR y FRR para cada arquitectura GARO, donde se ha destacado el punto de intersección y sus correspondientes coordenadas en el eje de umbral de identificación (abscisas) y tasa de error (ordenadas). (b) Curva característica operativa del receptor (ROC) para cada arquitectura.

Tab. 5.4.: Resultados del análisis de identificabilidad para cada arquitectura GARO-PUF estudiada.

Arquitectura	u^{EER}	EER
GARO 3 $f(x) = 1 + x^3$	24	$4{,}76 \times 10^{-7}$
GARO 5 $f(x) = 1 + x^2 + x^4 + x^5$	21	$1{,}04 \times 10^{-7}$
GARO 7 $f(x) = 1 + x^2 + x^6 + x^7$	18	$1{,}32 \times 10^{-10}$

buenas propiedades de identificabilidad para la PUF basada en osciladores de anillo de Galois propuesta en este capítulo y, en especial, de la arquitectura de 7 etapas, la cual devuelve unos resultados excelentes con una tasa de error EER $\sim 10^{-11}$.

5.3. Conclusión

En este capítulo se ha propuesto una arquitectura novedosa de función no-clonable físicamente basada en osciladores de anillo de Galois, adecuada para su implementación en FPGA. Su funcionalidad ha sido demostrada mediante el diseño de una serie de prototipos PUF basadas en distintas configuraciones de anillos; en particular hemos analizado tres estructuras diferentes de oscilador, demostrando que su variación en el sesgo depende de las localizaciones del anillo dentro de la FPGA, de forma similar a las frecuencias en un oscilador de anillo estándar. En este sentido, la implementación más destacable llevada a cabo ha sido la GARO-PUF de 7 etapas, la cual muestra una excelente identificabilidad comparable con los valores obtenidos para las implementaciones de osciladores de anillo estándar en FPGA (tabla 4.4), y unos resultados de reproducibilidad y unicidad homologables con el estado de la técnica.

Así mismo, se ha estudiado la distribución del sesgo en esta clase de sistemas, pudiendo concluir que estos muestran una menor correlación espacial que su contrapartida basada en osciladores de anillo estándar [139]. Esto abre la posibilidad a diseñar PUF con un mayor espacio de retos, dado que permite relajar la restricción de realizar comparaciones únicamente entre osciladores físicamente cercanos en la matriz FPGA, tal y como ocurre en el caso de RO-PUF. Esto tiene implicaciones importantes en el diseño de funciones no-clonables físicamente dado que permite desarrollar PUF con mejor desempeño en cuanto a seguridad, *e.g.*, más robustas frente a ataques de modelado. Finalmente, el trabajo mostrado aquí prueba por primera vez la viabilidad de utilizar osciladores no-lineales de forma general como funciones físicas elementales en el diseño PUF.

Conclusiones y líneas futuras

6

6.1. Conclusiones

A lo largo de estos cinco capítulos se ha descrito en detalle el papel que ocupa la seguridad de la información en el contexto de la tecnología IoT y la computación "en el borde", y se ha examinado en detalle las posibilidades que ofrecen las funciones no-clonables físicamente como raíces de la confianza para protocolos de identificación, autenticación, generación y almacenamiento seguro de claves en un entorno con restricciones al consumo de potencia y silicio. En los capítulos primero y segundo se ha introducido el campo de la seguridad de la información y la criptografía desde una perspectiva histórica, lo cual permite poner en contexto el valor añadido de esta disciplina y la importancia capital de que goza el ámbito de la seguridad de la información y la seguridad en las comunicaciones. Así mismo, se ha desglosado el conjunto de conceptos y técnicas abordados a lo largo de esta memoria de tesis, con la vocación de ser exhaustivo y autocontenido. Se ha propuesto un formalismo general en el cual separamos de manera explícita la raíz física de la confianza, *i.e.*, el elemento físico que actúa como fuente de entropía, de la interfaz digital que permite la integración de la PUF en un sistema de electrónico. También se ha propuesto un modelo original "cuasi-ideal" de PUF que permite justificar los ajustes binomiales de las curvas de intra/inter-distancia obtenidas experimentalmente, así como proporcionar una estimación numérica de la bondad de dicho ajuste a través de un test de hipótesis.

En el capítulo tercero se ha aplicado el formalismo general de PUF propuesto en el capítulo segundo para capturar la casuística de las funciones no-clonables físicamente de medida compensada. Esto ha permitido definir las "funciones físicas modulares", entendidas como aquellas constituidas por un vector de celdas físicas elementales diseñadas para proporcionar respuestas idénticas a un mismo estímulo. También se ha propuesto y elaborado un modelo estocástico para el proceso de fabricación de una función física modular y se ha introducido la noción de "topología" característica de una PUF de medida compensada, que describe la relación de pares de celdas cuya medición y subsiguiente comparación dará lugar a la respuesta

binaria. Dicho modelo de fabricación ha sido utilizado para deducir la forma de las distribuciones de respuesta características de cada topología estudiada, así como las leyes de escala de la entropía y minentropía con el tamaño de la matriz de celdas físicas. Esta aproximación semi-empírica constituye un enfoque novedoso para resolver el problema general de la entropía extraíble en funciones no-clonables físicamente en general, y PUF de medida compensada en particular. Así mismo, este estudio ha permitido valorar de manera exhaustiva el impacto de las técnicas de digitalización en las propiedades de seguridad exhibidas por PUF de medida compensada, y ha cristalizado en la propuesta de una familia inédita de topologías bautizada como "K-modular". Se han estudiado dos propuestas de topología en el ámbito de la familia "K-modular" (3-modular y 4-modular), y se ha justificado su idoneidad mediante la introducción de una figura de mérito denominada "coste", la cual es capaz de capturar el compromiso entre el nivel de seguridad proporcionado por una solución PUF y el impacto en el consumo de recursos respecto del *hardware* sobre el que se implementa. Finalmente, todos los resultados procedentes de los modelos propuestos han sido validados experimentalmente y comparados con resultados conocidos de la teoría de PUF, *e.g.*, la entropía total teórica frente al número de osciladores para una RO-PUF, o las estimaciones de entropía y minentropía obtenidas a partir de medidas experimentales.

Los capítulos cuarto y quinto están dedicados al diseño, implementación y evaluación experimental de alternativas PUF sobre FPGA. En el capítulo cuatro se han identificado claramente los parámetros de diseño accesibles en el flujo de diseño *semi-custom* propio de las FPGA Artix 7 que tienen un mayor impacto en las propiedades de seguridad del circuito PUF resultante, a saber, el tipo de *slice* (0/1), el tipo de LUT (M/L) y la orientación del CLB (I/D), y se ha relacionado cada uno de estos con un parámetro de diseño característico a determinar durante la implementación de matrices de osciladores de anillo en esta plataforma.

Se ha llevado a cabo una serie de experimentos diseñados para explorar el efecto de cada uno de estos parámetros en las propiedades de seguridad de una RO-PUF. Los resultados de estos experimentos han revelado que el tipo de *slice* tiene una influencia crucial, mientras que el impacto relacionado con el tipo de LUT es moderado y la orientación del CLB es prácticamente insignificante. Estos hallazgos han permitido concluir que una matriz de osciladores de anillo de tres etapas implementada utilizando *slice* (1) y LUT (L) representa la arquitectura óptima para la implementación de RO-PUF en las FPGA Artix 7 .

A continuación, hemos aplicado este resultado a algunas alternativas optimizadas de la PUF basada en osciladores de anillo estándar. En primer lugar, se ha estudiado la posibilidad de extraer una cantidad de entropía mayor de cada pareja de osciladores aumentando el número de bits de resolución de la diferencia de frecuencias, típicamente restringida al signo de la comparación en el caso estándar. Se han analizado las propiedades de seguridad al utilizar cada uno de los bits disponibles en una variable entera de 32 bits con signo, utilizada para almacenar la diferencia de frecuencia entre pares de osciladores. Seguidamente, se han evaluado las combinaciones de bits más prometedoras para preservar unas buenas propiedades de seguridad (unicidad, reproducibilidad e identificabilidad) y, simultáneamente, aumentar la longitud de las respuestas. Como resultado óptimo, se ha identificado la combinación de tres bits (bit de signo, bit 19 y bit 18). Esto posibilita la implementación de soluciones PUF altamente confiables con una cantidad reducida de celdas físicas, lo que conlleva a un menor consumo energético y una menor superficie de silicio.

Para concluir este apartado, se exploró otra alternativa a la RO-PUF mediante el uso de líneas de retardo programables (PDL). Esta técnica permite aumentar el número de bits efectivos extraídos de cada pareja de osciladores en una matriz de osciladores de anillo, aprovechando la configurabilidad interna de las LUT en FPGA. En esta tesis hemos propuesto una RO-PUF configurable mediante PDL y hemos llevado a cabo un análisis inédito a propósito del impacto que tiene la entrada física utilizada en cada LUT para generar el bucle de realimentación en las propiedades de la RO-PUF configurable resultante.

En el quinto capítulo se ha introducido una nueva propuesta PUF que emplea matrices de osciladores de anillo de Galois como función física modular para generar la respuesta, aprovechando un defecto inherente de los osciladores de Galois en tanto que generadores de números verdaderamente aleatorios (TRNG), por el cual cada anillo presenta un cierto sesgo sistemático en la uniformidad (*i.e.*, proporción de 1 a 0). En esta tesis se ha propuesto utilizar esta característica para diseñar una PUF de medida compensada y se han demostrado algunas propiedades deseables de esta alternativa respecto de la RO-PUF estándar. Por un lado, el sesgo de cada oscilador de Galois se correlaciona de manera significativamente más débil con la posición que ocupa dicho elemento en el chip FPGA que la frecuencia de los osciladores de anillo estándar. Esto alivia una restricción fundamental presente en la matriz de osciladores estándar, donde se deben evitar comparaciones entre parejas formadas por osciladores físicamente distantes debido a que el resultado de la comparación estará, con una alta probabilidad, predeterminado por el sesgo

sistemático de la frecuencia. Esta cualidad permite ampliar de manera efectiva el número de pares aceptables para construir la respuesta digital, permitiendo una mayor flexibilidad en la elección de una topología de comparación que se adapte de forma óptima a cada caso de aplicación. Por otro lado, esta propuesta innovadora de PUF genera de manera natural un flujo de bits verdaderamente aleatorios como parte de su operación normal, de tal forma que esta propuesta combina de hecho dos primitivas criptográficas integradas en una única estructura: una PUF y un TRNG. Dado que ambas primitivas aparecen con frecuencia como parte de un mismo protocolo criptográfico, esto permitiría optimizar los recursos *hardware* en un sentido global para una solución criptográfica que incluya una función no-clonable físicamente. No obstante, el uso de un oscilador de Galois como TRNG requerirá de lógica auxiliar destinada a corregir su comportamiento defectuoso y, en todo caso, se trata de una alternativa que deberá ser objeto de una investigación más profunda.

Para terminar, cabe mencionar que, a pesar de que el objeto de esta tesis doctoral ha sido la implementación de funciones no-clonables físicamente sobre FPGA, muchos de los resultados obtenidos son de aplicación al caso de PUF implementadas sobre integrados de propósito específico (ASIC). Por ejemplo, las conclusiones de los capítulos 2 y 3 a propósito del modelado de PUF, la distribución esperada de PUF cuasi-ideal y la noción de topología asociada a una PUF de medida compensada, son de aplicación inmediata en diseños implementados sobre tecnologías diferentes de la FPGA. Igualmente, los osciladores de anillo de Galois y la GARO-PUF propuesta en el capítulo 5 resulta adecuada para su implementación en ASIC, conservando presumiblemente muchas de las ventajas discutidas en esta tesis, *e.g.*, la ausencia de correlación espacial o de fenómenos de bloqueo frecuencial, de que también sufre la RO-PUF estándar implementada en integrados dedicados.

6.2. Líneas futuras

El ámbito de la funciones no-clonables físicamente es objeto de una investigación frenética en la carrera por alcanzar soluciones robustas y ligeras que, sin embargo, proporcionen buenos niveles de seguridad de la información a dispositivos distribuidos en el creciente ecosistema IoT. En este sentido, el trabajo realizado en esta tesis doctoral contribuye a despejar algunas de las principales incógnitas abiertas a propósito de la tecnología PUF [143], [162].

Algunas de las líneas de investigación exploradas en este trabajo y cuya continuación puede contribuir significativamente a lograr el objetivo de un estándar PUF para IoT incluyen:

1. Cálculo más general de la entropía de una PUF: refinar el modelo de fabricación dado en el capítulo 3 para contemplar no-idealidades en las propiedades de las celdas que constituyen una función física modular, por ejemplo, componentes sistemáticos en las mediciones en función de la ubicación de la celda en la FPGA. Una primera vía de aproximación para el caso de RO-PUF es el estudio de la probabilidad de inversión de bits dado en 2.3.5.

2. Diseño de una interfaz PUF integrada que evalúe las celdas físicas estudiadas en esta tesis (osciladores de anillo estándar y de Galois) de forma intrínseca al chip, lo cual permitirá realizar una estimación precisa de los recursos *hardware* empleados en un escenario de aplicación. Esto permitirá calcular el parámetro de diseño α introducido en la sección 3.4.

3. Implementación combinada de las soluciones propuestas, las cuales han sido presentadas deliberadamente de forma no-excluyente, por ejemplo, aplicación de la producción de respuestas multibit y la configurabilidad PDL estudiadas en las secciones 4.4 y 4.5 a los osciladores de Galois presentados en el capítulo 5.

4. Implementación de una solución arquetípica en un protocolo real de criptografía portable, adecuado para ser trasladado a dispositivos IoT concretos. Esto incluye el ensayo de soluciones de corrección de errores tal y como fueron descritas en la sección 2.3.5 aplicadas a las alternativas PUF propuestas en este trabajo.

5. Criptoanálisis de las soluciones propuestas, en particular un análisis de los ataques de modelado sobre las soluciones diseñadas para expandir la longitud de las respuestas (PDL y respuesta multibit) bajo el supuesto de que un adversario es capaz de acceder a fragmentos de la respuesta binaria.

Además de estas líneas de investigación vinculadas directamente con los resultados obtenidos en esta tesis doctoral, existen otras alternativas en el campo de las PUF implementadas en FPGA que están siendo objeto de estudio en la actualidad. Algunas de estas en las que trabaja actualmente el Grupo de Diseño Electrónico (GDE) de la Universidad de Zaragoza incluyen el incremento de la identificabilidad en PUF

basadas de medida compensada mediante la aplicación de transformaciones no-lineales a la magnitud medida de cada celda previamente a la generación de la respuesta binaria [163], o la generalización de los anillos de Galois utilizando funciones lógicas diferentes de XOR en el bucle de realimentación [164].

Bibliografía

[1]H. Sidhpurwala, "A brief history of cryptography," *Red Hat Customer Portal*, 2013 (vid. pág. 3).

[2]D. Kahn, *The Codebreakers: The comprehensive history of secret communication from ancient times to the internet*. Simon y Schuster, 1996 (vid. págs. 3, 5, 7, 10).

[3]J. F. Dooley, *History of cryptography and cryptanalysis: Codes, Ciphers, and their algorithms*. Springer, 2018 (vid. pág. 5).

[4]J. E. Wilcox, *Solving the Enigma*. Center for Cryptologic History, National Security Agency., 2006 (vid. pág. 10).

[5]D. Davies, "A brief history of cryptography," *Information Security Technical Report*, vol. 2, n.º 2, págs. 14-17, 1997 (vid. pág. 10).

[6]C. E. Shannon, "A mathematical theory of secrecy systems," *Bell system technical Journal*, vol. 28, págs. 623-656, 1949 (vid. págs. 10, 38).

[7]C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, n.º 3, págs. 379-423, 1948 (vid. págs. 10, 23, 24, 30).

[8]W. Diffie y M. E. Hellman, "New directions in cryptography," en *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, 2022, págs. 365-390 (vid. págs. 11, 38).

[9]Ç. K. Koç y F. Özdemir, "Development of Cryptography since Shannon," en *Handbook of Formal Analysis and Verification in Cryptography*, CRC Press, 2023, págs. 1-56 (vid. pág. 12).

[10]J. Daemen y V. Rijmen, *The design of Rijndael*. Springer, 2002, vol. 2 (vid. pág. 12).

[11]J. Franco, *Security Vulnerability Assessment Course*, 2023 (vid. pág. 13).

[12]Y. Zhou y D. Feng, "Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing," *Cryptology ePrint Archive*, 2005 (vid. pág. 13).

[13]E. Gray-Fow, *A Brief Peek Into the Fascinating World of Side Channel Attacks*, 2019 (vid. pág. 13).

[14]P. Kocher, R. Lee, G. McGraw y A. Raghunathan, "Security as a new dimension in embedded system design," en *Proceedings of the 41st annual design automation conference*, 2004, págs. 753-760 (vid. pág. 13).

[15]N. I. of Standards y T. (NIST), *Module-Lattice-based KeyEncapsulation Mechanism Standard*, Federal Information Processing Standards Publication (FIPS) NIST FIPS 203 ipd, 2023 (vid. pág. 13).

[16]N. I. of Standards y T. (NIST), *Module-Lattice-Based Digital Signature Standard*, Federal Information Processing Standards Publication (FIPS) NIST FIPS 204 ipd, 2023 (vid. pág. 13).

[17]N. I. of Standards y T. (NIST), *Stateless Hash-Based Digital Signature Standard*, Federal Information Processing Standards Publication (FIPS) NIST FIPS 205 ipd, 2023 (vid. pág. 13).

[18]H. Handschuh, G.-J. Schrijen y P. Tuyls, "Hardware intrinsic security from physically unclonable functions," *Towards Hardware-Intrinsic Security: Foundations and Practice*, págs. 39-53, 2010 (vid. pág. 14).

[19]M. Garcia-Bosque, G. Díez-Señorans, C. Sánchez-Azqueta y S. Celma, "Introduction to physically unclonable fuctions: Properties and applications," en *2020 European Conference on Circuit Theory and Design (ECCTD)*, IEEE, 2020, págs. 1-4 (vid. págs. 14, 217).

[20]Y. Gao, S. F. Al-Sarawi y D. Abbott, "Physical unclonable functions," *Nature Electronics*, vol. 3, n.º 2, págs. 81-91, 2020 (vid. pág. 14).

[21]C. Böhm y M. Hofer, *Physical unclonable functions in theory and practice*. Springer Science & Business Media, 2012 (vid. pág. 15).

[22]K. Lofstrom, W. R. Daasch y D. Taylor, "IC identification circuit using device mismatch," en *2000 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No. 00CH37056)*, IEEE, 2000, págs. 372-373 (vid. pág. 15).

[23]R. Pappu, B. Recht, J. Taylor y N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, n.º 5589, págs. 2026-2030, 2002 (vid. págs. 15, 75).

[24]B. Gassend, D. Clarke, M. Van Dijk y S. Devadas, "Silicon physical random functions," en *Proceedings of the 9th ACM Conference on Computer and Communications Security*, 2002, págs. 148-160 (vid. págs. 16, 66, 75, 127).

[25]B. Škorić, S. Maubach, T. Kevenaar y P. Tuyls, "Information-theoretic analysis of capacitive physical unclonable functions," *Journal of Applied physics*, vol. 100, n.º 2, 2006 (vid. pág. 16).

[26]U. Rührmair y M. van Dijk, "Practical security analysis of PUF-based two-player protocols," en *Cryptographic Hardware and Embedded Systems–CHES 2012: 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings 14*, Springer, 2012, págs. 251-267 (vid. pág. 16).

[27]B. Skoric, G.-J. Schrijen, W. Ophey, R. Wolters, N. Verhaegh y J. van Geloven, "Experimental hardware for coating PUFs and optical PUFs," *Security with Noisy Data: On Private Biometrics, Secure Key Storage and Anti-Counterfeiting*, págs. 255-268, 2007 (vid. pág. 16).

[28]S. Stanzione, D. Puntin y G. Iannaccone, "CMOS silicon physical unclonable functions based on intrinsic process variability," *IEEE Journal of Solid-State Circuits*, vol. 46, n.º 6, págs. 1456-1463, 2011 (vid. pág. 16).

[29]R. Maes y R. Maes, *Physically unclonable functions: Concept and constructions*. Springer, 2013 (vid. págs. 16, 59, 76, 100, 144).

[30]Y. Su, J. Holleman y B. P. Otis, "A digital 1.6 pJ/bit chip identification circuit using process variations," *IEEE Journal of Solid-State Circuits*, vol. 43, n.º 1, págs. 69-77, 2008 (vid. pág. 16).

[31]M. Majzoobi, G. Ghiaasi, F. Koushanfar y S. R. Nassif, "Ultra-low power current-based PUF," en *2011 IEEE international symposium of circuits and systems (ISCAS)*, IEEE, 2011, págs. 2071-2074 (vid. pág. 16).

[32]C.-E. Yin y G. Qu, "Temperature-aware cooperative ring oscillator PUF," en *2009 IEEE International Workshop on Hardware-Oriented Security and Trust*, IEEE, 2009, págs. 36-42 (vid. pág. 16).

[33]T. Xu y M. Potkonjak, "Digital PUF using intentional faults," en *Sixteenth International Symposium on Quality Electronic Design*, IEEE, 2015, págs. 448-451 (vid. pág. 16).

[34]J. Miao, M. Li, S. Roy y B. Yu, "LRR-DPUF: Learning resilient and reliable digital physical unclonable function," en *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, IEEE, 2016, págs. 1-8 (vid. pág. 16).

[35]R. Maes, A. Van Herrewege e I. Verbauwhede, "PUFKY: A fully functional PUF-based cryptographic key generator," en *Cryptographic Hardware and Embedded Systems–CHES 2012: 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings 14*, Springer, 2012, págs. 302-319 (vid. pág. 16).

[36]M. Hiller, L. Kürzinger, G. Sigl, S. Müelich, S. Puchinger y M. Bossert, "Low-area reed decoding in a generalized concatenated code construction for PUFs," en *2015 IEEE Computer Society Annual Symposium on VLSI*, IEEE, 2015, págs. 143-148 (vid. pág. 16).

[37]M. Hiller, "Key derivation with physical unclonable functions," Tesis doct., Technische Universität München, 2016 (vid. pág. 16).

[38]Y. Dodis, L. Reyzin y A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," en *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23*, Springer, 2004, págs. 523-540 (vid. pág. 16).

[39]E. Camacho-Ruíz, R. Castro-Lopez, E. Roca, P. Brox y F. V. Fernandez, "A novel physical unclonable function using RTN," en *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2022, págs. 160-164 (vid. pág. 16).

[40]P. S. Meka, R. Sivaraman, A. Rengarajan y S. Rajagopalan, "Metastability Influenced PUF for cryptographic key generation: a FPGA Approach," en *2020 International Conference on Computer Communication and Informatics (ICCCI)*, IEEE, 2020, págs. 1-6 (vid. pág. 16).

[41]R. Serrano, C. Duran, M. Sarmiento, T.-K. Dang, T.-T. Hoang y C.-K. Pham, "A Unified PUF and Crypto Core Exploiting the Metastability in Latches," *Future Internet*, vol. 14, n.º 10, pág. 298, 2022 (vid. pág. 16).

[42]S. C. Konigsmark, L. K. Hwang, D. Chen y M. D. Wong, "CNPUF: A carbon nanotube-based physically unclonable function for secure low-energy hardware design," en *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, IEEE, 2014, págs. 73-78 (vid. pág. 16).

[43]J. Schroeder, J. W. Borchert, P. Schuster, P. Eder, S. Heiserer, J. Biba, G. S. Duesberg, U. Rührmair y R. T. Weitz, "Organic and carbon nanotube electronics for flexible nanoscale high-frequency circuits and physical unclonable function," en *2022 IEEE International Flexible Electronics Technology Conference (IFETC)*, IEEE, 2022, págs. 1-2 (vid. pág. 16).

[44]Y. Cui, C. Wang, W. Liu, C. Gu, M. O'Neill y F. Lombardi, "Lightweight configurable ring oscillator PUF based on RRAM/CMOS hybrid circuits," *IEEE Open Journal of Nanotechnology*, vol. 1, págs. 128-134, 2020 (vid. pág. 16).

[45]E. I. Vatajelu, G. D. Natale, M. Barbareschi, L. Torres, M. Indaco y P. Prinetto, "STT-MRAM-based PUF architecture exploiting magnetic tunnel junction fabrication-induced variability," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 13, n.º 1, págs. 1-21, 2016 (vid. pág. 16).

[46]Y. Hu, L. Wu, Z. Chen, Y. Huang, X. Xu, K. Li y J. Zhang, "STT-MRAM-based reliable weak PUF," *IEEE Transactions on Computers*, vol. 71, n.º 7, págs. 1564-1574, 2021 (vid. pág. 16).

[47]J. Dreyer, R. Tönjes y N. Aschenbruck, "Towards Generating True Random Numbers using Magnetoresistive RAM," en *2023 Wireless Telecommunications Symposium (WTS)*, IEEE, 2023, págs. 1-7 (vid. pág. 16).

[48]N. Satheesh, A. Mahapatra, S. Kumar, S. Sahoo y K. K. Mahapatra, "A modified RO-PUF with improved security metrics on FPGA," en *2016 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS)*, IEEE, 2016, págs. 178-181 (vid. pág. 17).

[49]N. N. Anandakumar, M. S. Hashmi y S. K. Sanadhya, "Compact implementations of FPGA-based PUFs with enhanced performance," en *2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID)*, IEEE, 2017, págs. 161-166 (vid. pág. 17).

[50]Z. Wei, Y. Cui, Y. Chen, C. Wang, C. Gu y W. Liu, "Transformer PUF: A highly flexible configurable RO PUF based on FPGA," en *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, IEEE, 2020, págs. 1-6 (vid. págs. 17, 127).

[51]Y. Cui, Y. Chen, C. Wang, C. Gu, M. O'Neill y W. Liu, "Programmable ring oscillator PUF based on switch matrix," en *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2020, págs. 1-4 (vid. pág. 17).

[52]A. Oun y M. Niamat, "Design of a delay-based fpga puf resistant to machine learning attacks," en *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, IEEE, 2021, págs. 865-868 (vid. pág. 17).

[53]U. Chatterjee, R. S. Chakraborty y D. Mukhopadhyay, "A PUF-based secure communication protocol for IoT," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, n.º 3, págs. 1-25, 2017 (vid. pág. 17).

[54]A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, n.º 8, pág. 352, 2018 (vid. pág. 17).

[55]Z. Huang y Q. Wang, "A PUF-based unified identity verification framework for secure IoT hardware via device authentication," *World Wide Web*, vol. 23, n.º 2, págs. 1057-1088, 2020 (vid. pág. 17).

[56]P. A. Laplante, M. Kassab, N. L. Laplante y J. M. Voas, "Building caring healthcare systems in the Internet of Things," *IEEE systems journal*, vol. 12, n.º 3, págs. 3030-3037, 2017 (vid. pág. 17).

[57]Z. Shae y J. J. Tsai, "On the design of a blockchain platform for clinical trial and precision medicine," en *2017 IEEE 37th international conference on distributed computing systems (ICDCS)*, IEEE, 2017, págs. 1972-1980 (vid. pág. 17).

[58]S. He, B. Cheng, H. Wang, Y. Huang y J. Chen, "Proactive personalized services through fog-cloud computing in large-scale IoT-based healthcare application," *China Communications*, vol. 14, n.º 11, págs. 1-16, 2017 (vid. pág. 17).

[59]T. Muhammed, R. Mehmood, A. Albeshri e I. Katib, "UbeHealth: A personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities," *IEEE Access*, vol. 6, págs. 32 258-32 285, 2018 (vid. pág. 17).

[60]E. Oyekanlu, C. Nelatury, A. O. Fatade, O. Alaba y O. Abass, "Edge computing for industrial IoT and the smart grid: Channel capacity for M2M communication over the power line," en *2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON)*, IEEE, 2017, págs. 1-11 (vid. pág. 17).

[61]A. C. Panchal, V. M. Khadse y P. N. Mahalle, "Security issues in IIoT: A comprehensive survey of attacks on IIoT and its countermeasures," en *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)*, IEEE, 2018, págs. 124-130 (vid. pág. 17).

[62]D. Singh, G. Tripathi, A. M. Alberti y A. Jara, "Semantic edge computing and IoT architecture for military health services in battlefield," en *2017 14th IEEE annual consumer communications & networking conference (CCNC)*, IEEE, 2017, págs. 185-190 (vid. pág. 17).

[63]Y. Wang, Z. Ren, H. Zhang, X. Hou e Y. Xiao, ""combat cloud-fog" network architecture for internet of battlefield things and load balancing technology," en *2018 IEEE international conference on smart internet of things (SmartIoT)*, IEEE, 2018, págs. 263-268 (vid. pág. 17).

[64]A. Magyari e Y. Chen, "Review of state-of-the-art FPGA applications in IoT Networks," *Sensors*, vol. 22, n.º 19, pág. 7496, 2022 (vid. pág. 17).

[65]A. Sehgal, V. Perelman, S. Kuryla y J. Schonwalder, "Management of resource constrained devices in the internet of things," *IEEE Communications Magazine*, vol. 50, n.º 12, págs. 144-149, 2012 (vid. pág. 18).

[66]V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal y B. Sikdar, "A survey on IoT security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, págs. 82 721-82 743, 2019 (vid. pág. 18).

[67]H. Chen, Y. Chen y D. H. Summerville, "A survey on the application of FPGAs for network infrastructure security," *IEEE Communications Surveys & Tutorials*, vol. 13, n.º 4, págs. 541-561, 2010 (vid. pág. 18).

[68]M. Elnawawy, A. Farhan, A. Al Nabulsi, A.-R. Al-Ali y A. Sagahyroon, "Role of FPGA in internet of things applications," en *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, IEEE, 2019, págs. 1-6 (vid. pág. 18).

[69]C. Pham-Quoc, "FPGA-Based Hardware/Software Codesign for Video Encoder on IoT Edge Platforms," en *International Conference on Computational Science and Its Applications*, Springer, 2023, págs. 82-96 (vid. pág. 18).

[70]J. Katz e Y. Lindell, *Introduction to modern cryptography: principles and protocols*. Chapman y hall/CRC, 2007 (vid. pág. 24).

[71]S. K. Black, "CHAPTER 9 - Encryption," en *Telecommunications Law in the Internet Age*, ép. The Morgan Kaufmann Series in Networking, S. K. Black, ed., San Francisco: Morgan Kaufmann, 2002, págs. 327-387 (vid. pág. 24).

[72]F. Özdemir y Ç. K. Koç, "Development of Cryptography since Shannon," *Cryptology ePrint Archive*, 2022 (vid. pág. 24).

[73]Ė. B. Vinberg, *A course in algebra*. American Mathematical Soc., 2003 (vid. pág. 25).

[74]M. Kuczma, *Introduction to the theory of functional equations and inequalities: Cauchy's equation and Jensen's inequality*. 2009 (vid. págs. 27, 113).

[75]T. M. Cover y J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006 (vid. pág. 30).

[76]A. Vassilev y T. A. Hall, "The importance of entropy to information security," *Computer*, vol. 47, n.º 2, págs. 78-81, 2014 (vid. pág. 33).

[77]B. Espinoza y G. Smith, "Min-entropy as a resource," *Information and Computation*, vol. 226, págs. 57-75, 2013 (vid. pág. 33).

[78]J. L. Massey, "Guessing and entropy," en *Proceedings of 1994 IEEE International Symposium on Information Theory*, IEEE, 1994, pág. 204 (vid. pág. 33).

[79]D. Malone y W. G. Sullivan, "Guesswork and entropy," *IEEE Transactions on Information Theory*, vol. 50, n.º 3, págs. 525-526, 2004 (vid. pág. 33).

[80]Q. Wang y G. Qu, "A silicon PUF based entropy pump," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, n.º 3, págs. 402-414, 2018 (vid. pág. 33).

[81]O. Rioul, P. Solé, S. Guilley y J.-L. Danger, "On the entropy of physically unclonable functions," en *2016 IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2016, págs. 2928-2932 (vid. pág. 33).

[82]M. E. Whitman y H. J. Mattord, *Principles of information security*. Cengage learning, 2021 (vid. pág. 33).

[83]J. K. Elaine Barker, "Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Revised)," inf. téc. NIST Special Publication (SP) 800-90A, 2012 (vid. pág. 34).

[84]T. Apostol, *Calculus, Volume 1*. Wiley, 1991, págs. 65-69 (vid. pág. 36).

[85]E. W. Ng y M. Geller, "A table of integrals of the error functions," *Journal of Research of the National Bureau of Standards B*, vol. 73, n.º 1, págs. 1-20, 1969 (vid. pág. 36).

[86]A. J. Menezes, P. C. Van Oorschot y S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1996 (vid. págs. 39, 45-47, 77).

[87]W. Van Eck, "Electromagnetic radiation from video display units: An eavesdropping risk?" *Computers & Security*, vol. 4, n.º 4, págs. 269-286, 1985 (vid. pág. 40).

[88]D. Merli, J. Heyszl, B. Heinz, D. Schuster, F. Stumpf y G. Sigl, "Localized electromagnetic analysis of RO PUFs," en *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, IEEE, 2013, págs. 19-24 (vid. pág. 40).

[89]P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," en *Advances in Cryptology—CRYPTO'96: 16th Annual International Cryptology Conference Santa Barbara, California, USA August 18–22, 1996 Proceedings 16*, Springer, 1996, págs. 104-113 (vid. pág. 41).

[90]P. Kocher, J. Jaffe y B. Jun, "Differential power analysis," en *Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings 19*, Springer, 1999, págs. 388-397 (vid. pág. 41).

[91]D. Karakoyunlu y B. Sunar, "Differential template attacks on PUF enabled cryptographic devices," en *2010 IEEE International Workshop on Information Forensics and Security*, IEEE, 2010, págs. 1-6 (vid. pág. 41).

[92]G. T. Becker y R. Kumar, "Active and passive side-channel attacks on delay based PUF designs," *Cryptology ePrint Archive*, 2014 (vid. pág. 41).

[93]D. Schuster, "Side-channel analysis of physical unclonable functions (PUFs)," *Master's thesis, Technische Universitat Munchen*, 2010 (vid. pág. 41).

[94]D. Nedospasov, J.-P. Seifert, C. Helfmeier y C. Boit, "Invasive PUF analysis," en *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, IEEE, 2013, págs. 30-38 (vid. pág. 42).

[95]B. Gassend, M. V. Dijk, D. Clarke, E. Torlak, S. Devadas y P. Tuyls, "Controlled physical random functions and applications," *ACM Transactions on Information and System Security (TISSEC)*, vol. 10, n.º 4, págs. 1-22, 2008 (vid. pág. 42).

[96]P. Ryan y S. A. Schneider, *The modelling and analysis of security protocols: the CSP approach*. Addison-Wesley Professional, 2001 (vid. pág. 44).

[97]U. de Oxford, *FDR4*, ver. 4.2.4, 2019 (vid. pág. 44).

[98]J. L. Massey, "An introduction to contemporary cryptology," *Proceedings of the IEEE*, vol. 76, n.º 5, págs. 533-549, 1988 (vid. pág. 45).

[99]D. R. Stinson y M. B. Paterson, *Cryptography: theory and practice*. Chapman y Hall/CRC, 2019 (vid. pág. 45).

[100]S. Cryptography, "Stallings, Cryptography and Network Security: Principles and Practice," *Englewood Cliffs, NJ, USA: Printice-Hall*, vol. 11, n.º 7, págs. 655-660, 1999 (vid. pág. 47).

[101]A. Maiti, V. Gunreddy y P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," *Embedded systems design with FPGAs*, págs. 245-267, 2013 (vid. pág. 54).

[102]J.-L. Danger, S. Guilley, P. Nguyen y O. Rioul, "PUFs: Standardization and evaluation," en *2016 Mobile System Technologies Workshop (MST)*, IEEE, 2016, págs. 12-18 (vid. pág. 54).

[103]P.-N. Tan, "Receiver Operating Characteristic," en *Encyclopedia of Database Systems*, L. LIU y M. T. ÖZSU, eds. Boston, MA: Springer US, 2009, págs. 2349-2352 (vid. pág. 63).

[104]C. Martínez-Gómez e I. Baturone, "Calibration of ring oscillator PUF and TRNG," en *2020 European Conference on Circuit Theory and Design (ECCTD)*, IEEE, 2020, págs. 1-4 (vid. pág. 68).

[105]A. Maiti, J. Casarona, L. McHale y P. Schaumont, "A large scale characterization of RO-PUF," en *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, IEEE, 2010, págs. 94-99 (vid. pág. 75).

[106]A. Aysu, N. F. Ghalaty, Z. Franklin, M. P. Yali y P. Schaumont, "Digital fingerprints for low-cost platforms using MEMS sensors," en *Proceedings of the Workshop on Embedded Systems Security*, 2013, págs. 1-6 (vid. pág. 75).

[107]O. Willers, C. Huth, J. Guajardo, H. Seidel y P. Deutsch, "On the feasibility of deriving cryptographic keys from MEMS sensors," *Journal of Cryptographic Engineering*, vol. 10, n.º 1, págs. 67-83, 2020 (vid. pág. 75).

[108]B. Gassend, D. Clarke, M. van Dijk y S. Devadas, "Controlled physical random functions," en *18th Annual Computer Security Applications Conference, 2002. Proceedings.*, 2002, págs. 149-160 (vid. pág. 75).

[109]U. Rührmair, J. Sölter y F. Sehnke, "On the foundations of physical unclonable functions," *Cryptology ePrint Archive*, 2009 (vid. pág. 76).

[110]U. Rührmair y D. E. Holcomb, "PUFs at a glance," en *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2014, págs. 1-6 (vid. pág. 76).

[111]C. Herder, M.-D. Yu, F. Koushanfar y S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, n.º 8, págs. 1126-1141, 2014 (vid. pág. 76).

[112]J. Bringer, H. Chabanne y T. Icart, "On physical obfuscation of cryptographic algorithms," en *Progress in Cryptology-INDOCRYPT 2009: 10th International Conference on Cryptology in India, New Delhi, India, December 13-16, 2009. Proceedings 10*, Springer, 2009, págs. 88-103 (vid. pág. 76).

[113]R. Maes e I. M. R. Verbauwhede, "Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions," en *Towards Hardware-Intrinsic Security*, 2010 (vid. pág. 76).

[114]U. Rührmair, H. Busch y S. Katzenbeisser, "Strong PUFs: models, constructions, and security proofs," *Towards Hardware-Intrinsic Security: Foundations and Practice*, págs. 79-96, 2010 (vid. pág. 76).

[115]T. McGrath, I. E. Bagci, Z. M. Wang, U. Roedig y R. J. Young, "A puf taxonomy," *Applied physics reviews*, vol. 6, n.º 1, 2019 (vid. pág. 76).

[116]R. Padmavathy y M. N. Rajkumar, "Secured Cloud Communication Using Lightweight Hash Authentication with PUF.," *Computer Systems Science & Engineering*, vol. 43, n.º 1, 2022 (vid. pág. 80).

[117]K. Bhatia, S. K. Pandey, V. K. Singh y D. N. Gupta, "Hash and Physical Unclonable Function (PUF)-Based Mutual Authentication Mechanism," *Sensors*, vol. 23, n.º 14, pág. 6307, 2023 (vid. pág. 80).

[118]G. E. Suh y S. Devadas, "Physical unclonable functions for device authentication and secret key generation," en *Proceedings of the 44th annual design automation conference*, 2007, págs. 9-14 (vid. págs. 80, 100, 127).

[119]O. Goldreich, "Cryptography and cryptographic protocols," *Distributed Computing*, vol. 16, págs. 177-199, 2003 (vid. pág. 81).

[120]F. Vahid, *Digital design with RTL design, VHDL, and Verilog*. John Wiley & Sons, 2010 (vid. pág. 82).

[121]H. Jo, S.-T. Hong, J.-W. Chang y D. H. Choi, "Data encryption on GPU for high-performance database systems," *Procedia computer science*, vol. 19, págs. 147-154, 2013 (vid. pág. 84).

[122]C. Maxfield, *The design warrior's guide to FPGAs: devices, tools and flows*. Elsevier, 2004 (vid. pág. 87).

[123]J. L. F. Paul A. Grassi Michael E. Garcia, "Digital identity guidelines," inf. téc. NIST Special Publication (SP) 800-63-3, 2017 (vid. pág. 93).

[124]J. Pliam, *The Disparity between Work and Entropy in Cryptology*, Cryptology ePrint Archive, Paper 1998/024, `https://eprint.iacr.org/1998/024`, 1998 (vid. pág. 93).

[125]J. O. Pliam, "On the incomparability of entropy and marginal guesswork in brute-force attacks," en *International conference on cryptology in India*, Springer, 2000, págs. 67-79 (vid. pág. 93).

[126]D. Malone y K. Maher, "Investigating the distribution of password choices," en *Proceedings of the 21st international conference on World Wide Web*, 2012, págs. 301-310 (vid. pág. 93).

[127]A. Maiti, J. Casarona, L. McHale y P. Schaumont, "A large scale characterization of RO-PUF," en *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, IEEE, 2010, págs. 94-99 (vid. págs. 100, 114).

[128]Y. Nasser, J.-C. Prévotet, M. Hélard y J. Lorandel, "Dynamic power estimation based on switching activity propagation," en *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, IEEE, 2017, págs. 1-2 (vid. pág. 107).

[129]G. Diez-Senorans, M. Garcia-Bosque, C. Sánchez-Azqueta y S. Celma, "Digitization algorithms in ring oscillator physically unclonable functions as a main factor achieving hardware security," *IEEE Access*, vol. 9, págs. 147 343-147 356, 2021 (vid. págs. 115, 217).

[130]D. Merli, F. Stumpf y C. Eckert, "Improving the quality of ring oscillator PUFs on FPGAs," en *Proceedings of the 5th workshop on embedded systems security*, 2010, págs. 1-9 (vid. págs. 124, 136).

[131]H. Yu, P. H. Leong y Q. Xu, "An FPGA chip identification generator using configurable ring oscillators," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 20, n.º 12, págs. 2198-2207, 2011 (vid. pág. 124).

[132]B. Habib, K. Gaj y J.-P. Kaps, "FPGA PUF based on programmable LUT delays," en *2013 Euromicro Conference on Digital System Design*, IEEE, 2013, págs. 697-704 (vid. págs. 124, 155).

[133]W. Xiong, A. Schaller, S. Katzenbeisser y J. Szefer, "Software protection using dynamic PUFs," *IEEE Transactions on Information Forensics and Security*, vol. 15, págs. 2053-2068, 2019 (vid. pág. 127).

[134]P. H. Nguyen, D. P. Sahoo, C. Jin, K. Mahmood, U. Rührmair y M. Van Dijk, "The interpose PUF: Secure PUF design against state-of-the-art machine learning attacks," *Cryptology ePrint Archive*, 2018 (vid. pág. 127).

[135]R. W. Rhea, *Discrete oscillator design: linear, nonlinear, transient, and noise domains*. Artech House, 2010 (vid. pág. 128).

[136]F. Wilde, B. M. Gammel y M. Pehl, "Spatial correlation analysis on physical unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, n.º 6, págs. 1468-1480, 2018 (vid. pág. 131).

[137]R. Adler, "A study of locking phenomena in oscillators," *Proceedings of the IRE*, vol. 34, n.º 6, págs. 351-357, 1946 (vid. pág. 132).

[138]U. Mureddu, N. Bochard, L. Bossuet y V. Fischer, "Experimental study of locking phenomena on oscillating rings implemented in logic devices," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, n.º 7, págs. 2560-2571, 2019 (vid. pág. 132).

[139]C. Gu, C. H. Chang, W. Liu, N. Hanley, J. Miskelly y M. O'Neill, "A large scale comprehensive evaluation of single-slice ring oscillator and PicoPUF bit cells on 28nm Xilinx FPGAs," en *Proceedings of the 3rd ACM Workshop on Attacks and Solutions in Hardware Security Workshop*, 2019, págs. 101-106 (vid. págs. 136, 179).

[140]M. L. Ch, A. B. Raj y L. Abhikshit, "Design and Implementation of a Secure Physical Unclonable Function In FPGA," en *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, IEEE, 2020, págs. 1083-1089 (vid. pág. 136).

[141]R. Aparicio-Téllez, M. Garcia-Bosque, G. Díez-Señorans y S. Celma, "Oscillator Selection Strategies to Optimize a Physically Unclonable Function for IoT Systems Security," *Sensors*, vol. 23, n.º 9, pág. 4410, 2023 (vid. págs. 137, 217).

[142]N. L. Johnson, "Systems of frequency curves generated by methods of translation," *Biometrika*, vol. 36, n.º 1/2, págs. 149-176, 1949 (vid. pág. 141).

[143]N. N. Anandakumar, M. S. Hashmi y M. Tehranipoor, "FPGA-based Physical Unclonable Functions: A comprehensive overview of theory and architectures," *Integration*, vol. 81, págs. 175-194, 2021 (vid. págs. 144, 148, 184).

[144]*TPS65400 4.5- to 18-V Input Flexible Power Management Unit With PMBus/I2C Interface*, TPS65400, v1.21, Texas Instruments, 2022 (vid. pág. 148).

[145]*Zynq-7000 SoC (Z-7007S, Z-7012S, Z-7014S, Z-7010, Z-7015, and Z-7020): DC and AC Switching Characteristics*, DS187, v1.21, Xilinx, 2020 (vid. pág. 148).

[146]M. C. Martinez-Rodriguez, E. Camacho-Ruiz, P. Brox y S. Sánchez-Solano, "A configurable RO-PUF for securing embedded systems implemented on programmable devices," *Electronics*, vol. 10, n.º 16, pág. 1957, 2021 (vid. pág. 150).

[147]J. Fernández-Aragón, G. Díez-Señorans, M. Garcia-Bosque y S. Celma, "Design and characterisation of a Physically Unclonable Function on FPGA using second-order compensated measurement," en *2022 IFIP/IEEE 30th International Conference on Very Large Scale Integration (VLSI-SoC)*, IEEE, 2022, págs. 1-2 (vid. págs. 150, 218).

[148]L. Zheng, C. Li, Z. Liu y C. Ma, "Boosting Entropy Extraction of PDL-based RO PUF by High-order Difference Method," en *Proceedings of the 2020 on Great Lakes Symposium on VLSI*, 2020, págs. 59-64 (vid. pág. 156).

[149]Q. Zhang, Z. Liu, C. Ma, C. Li y L. Zhang, "Fropuf: how to extract more entropy from two ring oscillators in fpga-based pufs," en *International Conference on Security and Privacy in Communication Systems*, Springer, 2016, págs. 675-693 (vid. pág. 157).

[150]G. Diez-Senorans, M. Garcia-Bosque, C. Sánchez-Azqueta y S. Celma, "Programmable delay lines on different lut implementations for cro-puf," en *2022 17th Conference on Ph. D Research in Microelectronics and Electronics (PRIME)*, IEEE, 2022, págs. 357-360 (vid. págs. 158, 218).

[151]E. Barker, "Recommendation for Key Management: Part 1 – General," inf. téc. NIST Special Publication (SP) 800-57, Rev. 5, 2020 (vid. pág. 161).

[152]M. Garcia-Bosque, G. Díez-Señorans, C. Sánchez-Azqueta y S. Celma, "Proposal and analysis of a novel class of PUFs based on Galois ring oscillators," *IEEE Access*, vol. 8, págs. 157 830-157 839, 2020 (vid. págs. 165, 217).

[153]M. Schramm, R. Dojen y M. Heigl, "Experimental assessment of FIRO-and GARO-based noise sources for digital TRNG designs on FPGAs," en *2017 International Conference on Applied Electronics (AE)*, IEEE, 2017, págs. 1-6 (vid. pág. 165).

[154]J. Lin, Y. Wang, Z. Zhao, C. Hui y Z. Song, "A new method of true random number generation based on galois ring oscillator with event sampling architecture in FPGA," en *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, IEEE, 2020, págs. 1-6 (vid. pág. 165).

[155]S. Eiroa e I. Baturone, "Hardware authentication based on PUFs and SHA-3 2 nd round candidates," en *2010 International Conference on Microelectronics*, IEEE, 2010, págs. 319-322 (vid. pág. 165).

[156]T. Idriss y M. Bayoumi, "Lightweight highly secure PUF protocol for mutual authentication and secret message exchange," en *2017 IEEE International Conference on RFID Technology & Application (RFID-TA)*, IEEE, 2017, págs. 214-219 (vid. pág. 165).

[157]I. Baturone, R. Román y Á. Corbacho, "A Unified Multibit PUF and TRNG based on Ring Oscillators for Secure IoT Devices," *IEEE Internet of Things Journal*, vol. 10, n.º 7, págs. 6182-6192, 2022 (vid. pág. 165).

[158]J. D. Golic, "New methods for digital generation and postprocessing of random data," *IEEE transactions on computers*, vol. 55, n.º 10, págs. 1217-1229, 2006 (vid. págs. 165, 167).

[159]S. Su, B. Yang, V. Rožić, M. Yang, M. Zhu, S. Wei y L. Liu, "A Closer Look at the Chaotic Ring Oscillators based TRNG Design," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, págs. 381-417, 2023 (vid. pág. 167).

[160]E. Watson, "Primitive polynomials (mod 2)," *Math. Comp*, vol. 16, n.º 368, pág. 1962, 1962 (vid. pág. 167).

[161]P. A. Moran, "Notes on continuous stochastic phenomena," *Biometrika*, vol. 37, n.º 1/2, págs. 17-23, 1950 (vid. pág. 172).

[162]A. Babaei y G. Schiele, "Physical unclonable functions in the internet of things: State of the art and open challenges," *Sensors*, vol. 19, n.º 14, pág. 3208, 2019 (vid. pág. 184).

[163]R. Aparicio-Tellez, M. Garcia-Bosque, G. Diez-Señorans y S. Celma, "Boosting the identifiability of a compensated measurement PUF via non-linear transformations," en *ECCTD*, 2023 (vid. págs. 186, 218).

[164]M. Garcia-Bosque, A. Naya, G. Díez-Señorans, C. Sánchez-Azqueta y S. Celma, "Suitability of Generalized GAROs on FPGAs as PUFs or TRNGs Considering Spatial Correlations," *IEEE Open Journal of the Industrial Electronics Society*, vol. 4, págs. 112-122, 2023 (vid. págs. 186, 217).

[165]L. H. Crockett, R. A. Elliot, M. A. Enderwitz y R. W. Stewart, *The Zynq book: embedded processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 all programmable SoC*. Strathclyde Academic Media, 2014 (vid. pág. 206).

[166]M. Garcia-Bosque, G. Díez-Señorans, A. Pérez-Resa, C. Sánchez-Azqueta, C. Aldea y S. Celma, "A 1 gbps chaos-based stream cipher implemented in 0.18 μm CMOS technology," *Electronics*, vol. 8, n.º 6, pág. 623, 2019 (vid. pág. 217).

[167]G. Díez-Señorans, M. Garcia-Bosque, C. Sánchez-Azqueta y S. Celma, "A new approach to analysis the security of compensated measuring PUFs," en *2020 European Conference on Circuit Theory and Design (ECCTD)*, IEEE, 2020, págs. 1-5 (vid. pág. 218).

[168]M. Garcia-Bosque, G. Díez-Señorans, C. Sánchez-Azqueta y S. Celma, "FPGA implementation of a new PUF based on Galois ring oscillators," en *2021 IEEE 12th Latin America Symposium on Circuits and System (LASCAS)*, IEEE, 2021, págs. 1-4 (vid. pág. 218).

[169]G. Diez-Senorans, M. Garcia-Bosque, C. Sanchez-Azqueta y S. Celma, "Entropy Analysis of RO-based Physically Unclonable Functions," en *SMACD/PRIME 2021; International Conference on SMACD and 16th Conference on PRIME*, VDE, 2021, págs. 1-4 (vid. pág. 218).

[170]M. Garcia-Bosque, A. Naya, G. Diez-Senorans, C. Sanchez-Azqueta y S. Celma, "On the Behavior of a Wide Set of Oscillators: PUFs or TRNGs?" En *SMACD/PRIME 2021; International Conference on SMACD and 16th Conference on PRIME*, VDE, 2021, págs. 1-4 (vid. pág. 218).

[171]M. Garcia-Bosque, R. Aparicio, G. Díez-Señorans, C. Sánchez-Azqueta y S. Celma, "An analysis of the behaviour of a PUF based on ring oscillators depending on their locations," en *2022 17th Conference on Ph. D Research in Microelectronics and Electronics (PRIME)*, IEEE, 2022, págs. 361-364 (vid. pág. 218).

[172]A. Naya-Forcano, M. Garcia-Bosque, G. Díez-Señorans y S. Celma, "Multiprogram tools for FPGA boards with single identifier on Windows," en *2023 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, IEEE, 2023, págs. 1-4 (vid. pág. 219).

[173]R. Aparicio, J. Fernández-Aragón, A. Naya-Forcano, G. Díez-Señorans, M. Garcia-Bosque y S. Celma, "Proposal of a new PUF based on sensors for the identification of IoT smart mobile devices," en *PUF-enabled Security Challenge*, European Cyber Security Awareness Week (CSAW), 2023 (vid. pág. 219).

[174]G. Díez-Señorans, M. Garcia-Bosque, F. Aznar-Tabuenca, C. Sánchez-Azqueta y S. Celma, "Análisis de la seguridad criptográfica en capa física proporcionada por PUFs de medida compensada," en *VIII Congreso Nacional de I+D en Defensa y Seguridad*, Ministerio de Defensa del Gobierno de España, 2020 (vid. pág. 219).

[175]M. Garcia-Bosque, G. Díez-Señorans, F. Aznar-Tabuenca, C. Sánchez-Azqueta y S. Celma, “Propuesta de una nueva clase de funciones no clonables físicamente para comunicaciones seguras,” en *VIII Congreso Nacional de I+D en Defensa y Seguridad*, Ministerio de Defensa del Gobierno de España, 2020 (vid. pág. 219).

[176]G. Díez-Señorans, M. Garcia-Bosque, F. Aznar-Tabuenca, C. Sánchez-Azqueta y S. Celma, “Mejora de la eficiencia de funciones no-clonables físicamente integrando líneas de retardo programables,” en *IX Congreso Nacional de I+D en Defensa y Seguridad*, Ministerio de Defensa del Gobierno de España, 2022 (vid. pág. 219).

[177]M. Garcia-Bosque, G. Díez-Señorans, F. Aznar-Tabuenca, C. Sánchez-Azqueta y S. Celma, “Estrategias de selección de osciladores en una PUF de oscilador de anillo para optimizar su comportamiento,” en *IX Congreso Nacional de I+D en Defensa y Seguridad*, Ministerio de Defensa del Gobierno de España, 2022 (vid. pág. 219).

Apéndices

Entropía máxima

A

Dada una variable aleatoria X la cual puede tomar N valores diferentes $\{x_i\}_{i=1}^{N}$ con probabilidades asociadas $p_i = P(\mathsf{X} = x_i)$, $\vec{p} \equiv (p_1, \cdots, p_N)$ y sea la función entropía:

$$H = H(\mathsf{X}) = -\sum_{i=1}^{N} p_i \log_2 p_i \tag{A.1}$$

junto con la restricción $\sum p_i = 1$. Se pueden calcular los valores $\vec{p} = \vec{p}^{\max}$ que maximizan la entropía H cuando están restringidos a tomar valores en la superficie de nivel $r = 1$, donde:

$$r = r(\vec{p}) = \sum_{i=1}^{N} p_i \tag{A.2}$$

haciendo uso del método de los multiplicadores de Lagrange[1]:

$$\nabla H|_{\vec{p}^{\max}} = \lambda \nabla r|_{\vec{p}^{\max}} \tag{A.3}$$

donde λ es una constante. Desarrollando (A.3) con la notación del operador derivada parcial $\partial_i \equiv \frac{\partial}{\partial p_i}$:

$$\begin{aligned}
\partial_j \left(-\sum_{i=1}^{N} p_i \log_2 p_i \right) &= \lambda \partial_j \left(\sum_{i=1}^{N} p_i \right) r \\
-\sum_{i=1}^{N} \left[\partial_j p_i \log_2 p_i + p_i \, \partial_j \log_2 p_i \right] &= \lambda \sum_{i=1}^{N} \partial_j p_i \\
-\sum_{i=1}^{N} \left[\delta_{j,i} \log_2 p_i + \frac{p_i}{|p_i|} \delta_{j,i} \right] &= \lambda \sum_{i=1}^{N} \delta_{j,i} \\
\log_2 p_j + 1 = \lambda \longrightarrow p_j^{\max} &= 2^{\lambda - 1}
\end{aligned} \tag{A.4}$$

[1] Sea un punto $\vec{q}$ en la superficie de nivel dada por la restricción $r =$ cte., el gradiente de la función ∇H en este punto se podrá escribir como la suma de una componente normal y otra tangente a dicha superficie de nivel. Bajo la hipótesis de que tal punto $\vec{q}$ corresponde a un extremal de H en la restricción (*i.e.*, $H(\vec{q})$ permanece constante al variar infinitesimal el argumento en cualquier dirección permitida por r), la componente tangente se anulará (en caso contrario un desplazamiento infinitesimal en esta dirección producirá un cambio a primer orden en H, contradiciendo la hipótesis de punto extremal). Por otra parte el gradiente de una función siempre es normal a sus curvas de nivel, en particular el gradiente ∇r en $\vec{q}$, de modo que en el punto extremal $\vec{q}$ los gradientes de H y r serán paralelos.

Introducimos esta expresión en la restricción $r = 1$ para despejar el multiplicador de Lagrange:

$$\sum_{i=1}^{N} p_i = \sum_{i=1}^{N} 2^{\lambda-1} = 1$$
$$2^{\lambda-1} N = 1 \longrightarrow \lambda - 1 = \log_2 \frac{1}{N} \tag{A.5}$$

Y sustituytendo $\lambda - 1$ en (A.4) tenemos finalmente:

$$\boxed{p_j^{\max} \equiv p = \frac{1}{N}} \tag{A.6}$$

Flujo de diseño en FPGA

B

En este apéndice mostramos los detalles del flujo de diseño digital sobre un chip Zynq7000 SoC (*System-on-Chip*) de Xilinx, el cual consta de un microprocesador ARM Cortex-A9 y una FPGA Artix 7 , utilizando el software de diseño asistido por ordenador Vivado-2019.1.

Los archivos necesarios para llevar a cabo este proceso se pueden clasificar en:

- Fuentes del diseño: uno o varios archivos escritos en algún lenguaje de descripción de hardware (HDL). Los estándares más extendidos en la industria son Verilog y VHDL. Estos lenguajes tienen una sintaxis muy similar a un lenguaje de programación de bajo nivel y, como estos, permiten una descripción conductual (*behavioral*) del circuito. Sin embargo, a diferencia del código fuente, este es elaborado por un software de diseño digital en lugar de compilado.

- Archivo de restricciones: en este documento (típicamente un archivo de texto) se especifican las restricciones físicas que debe cumplir el diseño. Entre estas, ocupa un lugar destacado la descripción de los relojes del sistema y sus frecuencias de operación. El *software* de diseño tratará de disponer los elementos físicos y sus interconexiones cumpliendo con las reglas dadas en este documento; en caso de violar algunas de las reglas descritas en este archivo, el proceso de implementación digital fallará.

En cuanto al proceso de diseño sobre FPGA, este consta de los siguientes pasos:

1. Elaboración: el diseño descrito en alto nivel utilizando un lenguaje de descripción de *hardware* se convierte en un diseño equivalente (en el sentido de que la relación lógica de entradas/salidas es la misma) denominado *netlist*, el cual está descrito únicamente mediante modelos HDL de puertas lógicas y elementos secuenciales (registros y *flip-flops*) ideales.

2. Síntesis: se sustituyen los elementos lógicos ideales por sus correspondientes modelos de elementos lógicos reales. El resultado es de nuevo una lista de elementos modelados en HDL (*netlist*), pero los módulos que forman el diseño corresponden a módulos reales que la herramienta EDA puede identificar para su posterior implementación. Las bibliotecas de elementos están pre-cargadas en el *software* propietario del fabricante de las FPGA (en este caso, Xilinx Vivado). Además, en el diseño sobre FPGA, el elemento lógico combinacional estándar es la LUT, la cual permite implementar cualquier celda lógica (2.4.1).

3. Implementación: en esta fase se localizan e interconectan los elementos reales descritos por la lista sintetizada. Durante la implementación, el *software* accede a los archivos de restricciones para verificar, mediante modelos temporales y eléctricos de los elementos reales, que estas se cumplen. Este proceso de optimización puede ser computacionalmente exigente, y puede fallar si las restricciones físicas han sido mal diseñadas.

4. Generación del archivo *bitstream*: esta fase es específica del flujo de diseño en FPGA, y comprende la exportación de un archivo *bitstream* ".bit", el cual contiene toda la información necesaria para que un chip FPGA concreto implemente un diseño digital, para lo cual este archivo puede ser volcado sobre la FPGA. Generalmente este proceso se lleva a cabo mediante un protocolo JTAG de cuatro hilos [165], sin embargo, las placas de desarrollo utilizadas en este trabajo incluyen un módulo FTDI FT2232HL para traducir el protocolo USB a JTAG, de modo que la herramienta Vivado es capaz de cargar el archivo ".bit" generado sobre el chip de forma autónoma.

Protocolo *handshake*

C

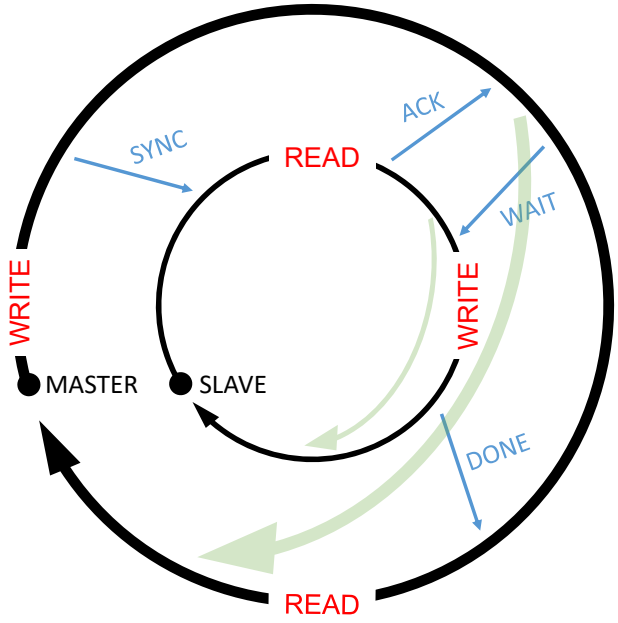


Fig. C.1.: Bucle de comunicación *master-slave* utilizando un protocolo *handshake*.

El protocolo *handshake* es un estándar de comunicación digital ampliamente extendido por su sencillez, y por la facilidad con la que puede ser sucesivamente sofisticado en función de los requerimientos de cada aplicación. Consta de dos canales: "datos" y "control", los cuales comunican dos nodos de una red: *master* y *slave*. El nodo *master* se define como aquel que inicia la comunicación, *i.e.*, el emisor, mientras *slave* define al nodo receptor. Notar que estos atributos no son invariantes, y un protocolo de comunicación puede implicar que un mismo nodo actúe como *master* o *slave* en instantes diferentes. En la figura C.1 se muestra el esquema de un ciclo de comunicación *master* a *slave* general, donde se han representado los canales de datos en el eje angular, y el canal de control en forma de líneas (aproximadamente) radiales. Así mismo, el flujo de tiempo transcurre tal y como indican las flechas angulares (en sentido horario). El protocolo comienza cuando *master* escribe la información que quiere transmitir en el canal de datos; a continuación envía la señal SYNC por el canal de control. Una vez esta es recibida, *slave* lee el canal de datos y seguidamente envía una señal ACK por el canal de control. En este instante la comunicación se puede dar por finalizada (notar las flechas semitransparentes en la

figura) e iniciar un nuevo ciclo de comunicación si *master* y *slave* están sincronizados o disponen de una memoria "primero en entrar, primero en salir" (*First-In-First-Out*, FIFO)[1] a modo de *buffer*, de tal manera que puedan asignar inequívocamente cada solicitud de *master* con su correspondiente respuesta de *slave*. En caso contrario, *i.e.*, si los nodos no están sincronizados o no disponen de una memoria, el protocolo debe continuar para garantizar que *master* ha recibido la respuesta de *slave* antes de iniciar un nuevo intercambio. Para ello, cuando *master* lee la señal ACK enviada por *slave*, cambia el valor del canal de control a WAIT. Por último, *slave* reacciona a este cambio escribiendo a su vez DONE en el canal de control (y, quizá, algún tipo de información en el canal de datos). En este punto, el sistema se encuentra en su estado inicial, preparado para que *master* reinicie el ciclo.

[1] *First-In-First-Out*, es un tipo de memoria en la cual, por construcción, la escritura y lectura se realiza de forma secuencial en el tiempo, de tal modo que el orden de salida de datos es el mismo que el orden de llegada.

Interfaz ordenador – FPGA

D

Para realizar los experimentos sobre la plataforma Zynq-7000 se ha diseñado una interfaz que permite la comunicación del segmento de lógica programable (*Programmable Logic*, PL) —la FPGA— con el ordenador (PC) a través del sistema de procesamiento (*Processing System*, PS) —el procesador ARM Cortex— integrado en el chip Zynq. En la figura D.1 se ha representado esquemáticamente la interfaz de comunicación diseñada para enviar información entre el ordenador y la FPGA. En esta red, el ordenador actúa como nodo maestro y se encarga de iniciar un ciclo de procesamiento, así como de proporcionar la información necesaria (en caso de haberla) para dicho procesamiento. Este nodo ha sido implementado utilizando la biblioteca "pyserial" de Python, la cual permite establecer una comunicación síncrona con el sistema de procesamiento a través de un protocolo (*Universal Asynchronous Receiver-Transmitter*, UART). El nodo central PS ha sido programado en lenguaje C utilizando el entorno de desarrollo de *software* de Xilinx (*Software Development Kit*, SDK) para programar el procesador integrado en los SoC de la serie 7, y cumple la función de traductor entre el ordenador, a través de un protocolo UART, y PL, a través del protocolo de interfaz extensible avanzada de ARM (*Advanced Extensible Interface*, AXI).

Los parámetros del sistema de comunicación que se ha diseñado son la anchura del *buffer* de entrada (*Buffer-in Width*, biw) y de salida (*Buffer-out Width*, bow), así como la anchura del bus de transmisión PS – PL (*Data Width*, dw). La comunicación ordenador – PL se lleva a cabo en dos etapas a través de PS.

Comunicación ordenador – PS

Esta se realiza mediante un protocolo estándar UART, capaz de transmitir palabras de ocho bits a través de un bus de un único bit (comunicación serie), incluyendo una memoria *buffer* a la entrada de cada nodo participante en la comunicación que garantiza la recepción de los mensajes en el orden correcto en que son enviados, sin necesidad de que los extremos estén sincronizados. El ciclo de comunicación comien-

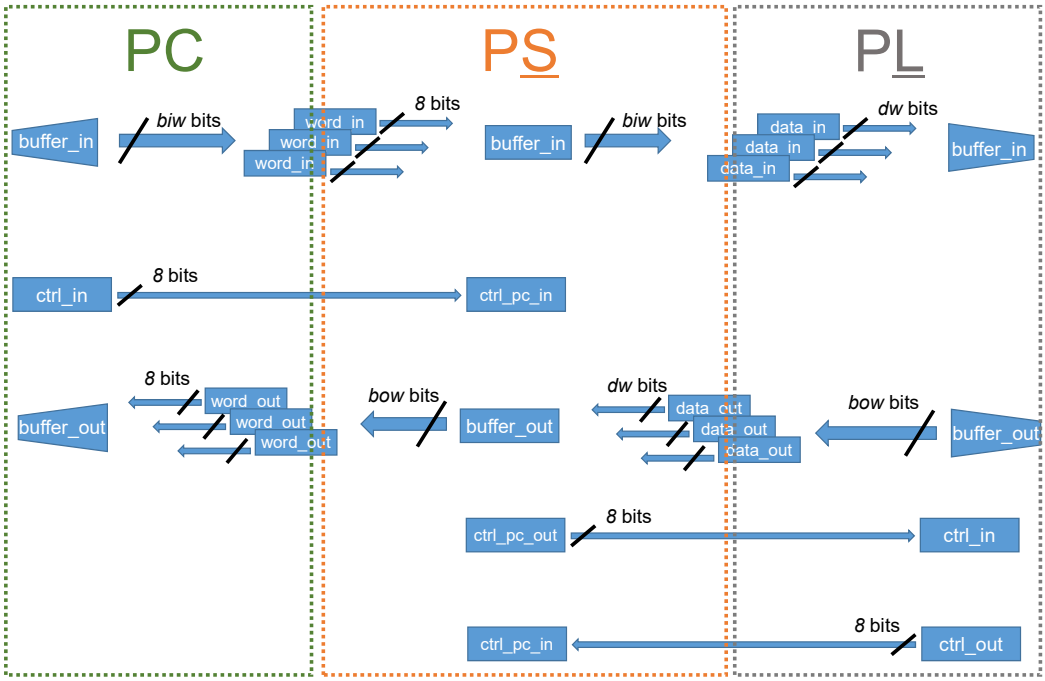


Fig. D.1.: Representación esquemática de la interfaz de comunicación PC – FPGA.

za cuando el ordenador envía un comando, el cual es interpretado por una máquina de estados finitos en el extremo PS (figura D.3). Las señales de comunicación en el extremo PC de la red son las siguientes:

- *buffer_in*, trama de datos que contiene los datos de entrada para la operación del diseño en la FPGA (PL). Se trata de una palabra binaria de *biw* bits, que se envía al sistema de procesamiento del chip Zynq (*i.e.*, el procesador ARM) a través del puerto UART. La interfaz con el usuario se realiza a través del módulo "interfaz_pcps" implementado en Python (apéndice E). Este permite a un operador introducir el vector de bits de entrada *buffer_in* desde una consola o *script*, así como encargarse de fragmentar este vector en bytes (palabras de ocho bits) adecuados para ser enviados a través de un protocolo UART, el cual se ha implementado utilizando la API de Python "pyserial" de forma transparente para el usuario (*i.e.*, en el *backend* del módulo "interfaz_pcps").

- *ctrl_in*, señal constituida por un vector de ocho bits, el cual transmite un comando específico a la máquina de estados programada sobre PS (*i.e.*, un bucle cuyo flujo está controlado por el comando recibido desde el ordenador, ver figura D.3). Estos comandos están diseñados para inducir a PS a realizar

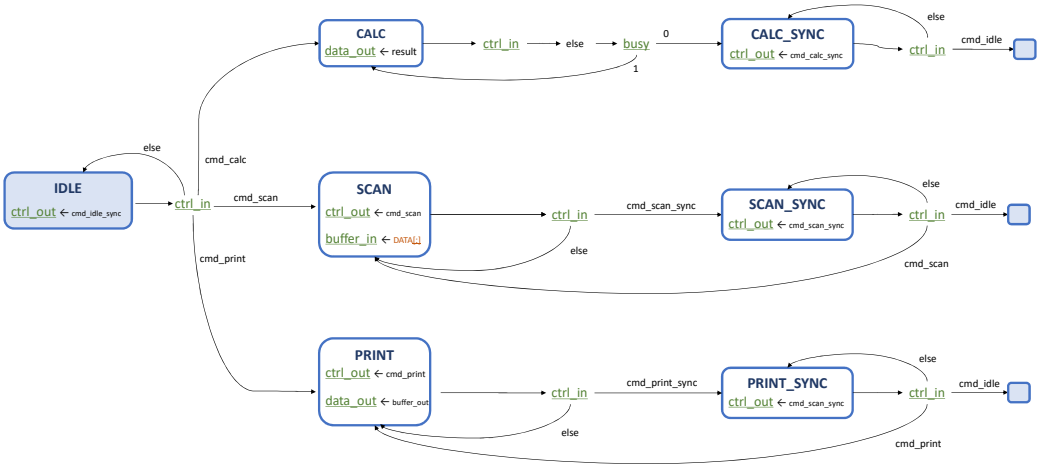


Fig. D.2.: Máquina de estados para implementar interfaz en el lado de la lógica programable (FPGA Artix 7).

ciertas acciones, por ejemplo, reenviar los datos *buffer_in* a PL, devolver la respuesta leída de PL de vuelta al ordenador, etc. Los comandos diseñados para controlar operación de esta máquina son:

- *calc*, este comando lleva a PS al estado CALC; ahora PS (como *master*) inicia una comunicación con PL (*slave*) a través de un protocolo *handshake* (apéndice C). En esta comunicación, PS envía la información recibida del ordenador durante la fase SCAN (en caso de que el diseño requiera de esta información), espera a que PL la procese y la recibe de vuelta. Una vez este ciclo ha terminado, PS salta automáticamente al estado PRINT. Finalmente, PS toma el rol de *master*, enviando *buffer_out_width* bits a través del UART hacia el ordenador.

- *scan*, este comando lleva PS al estado SCAN; una vez el ordenador ha enviado esta orden, coloca la trama de datos de tamaño *buffer_in* bits en el *buffer* de salida UART (no hay restricción en el tamaño de *buffer_in*, de modo que esto puede suponer enviar más de un byte a través del UART). Simultáneamente, PS leerá estos bytes y los reintegrará en la variable *data_pc_in*. Dado que las memorias *buffer* del protocolo UART se implementan como FIFO, el envío de la orden *scan* y los sucesivos bytes de la trama de datos siempre serán leídos en el orden correcto por PS, de modo que no se requiere sincronización entre el ordenador y la máquina PS en esta etapa.

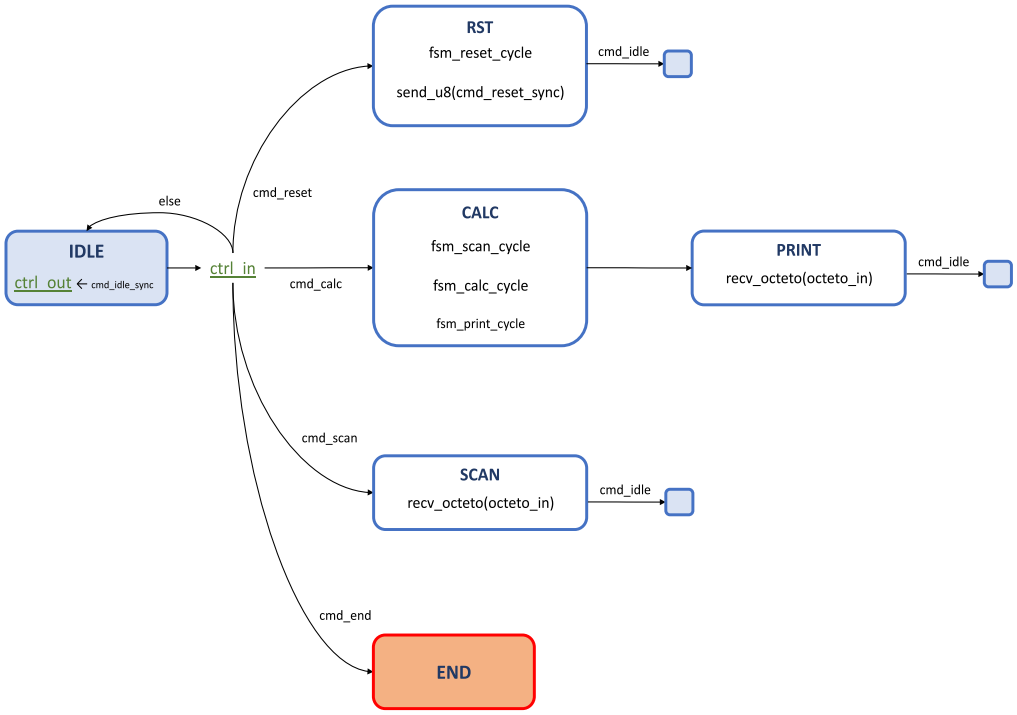


Fig. D.3.: Máquina de estados para implementar interfaz en el lado del sistema de procesamiento (microprocesador ARM Cortex-A9).

- *buffer_out*, variable que recibe el resultado de las operaciones de PS al finalizar un ciclo de operación; es leída por el puerto UART en fragmentos de ocho bits, y reintegrada a una palabra binaria de tamaño *bow* bits por el módulo "interfaz_pcps". Finalmente se presenta al usuario en forma de un valor entero (entre 0 y 2^{bow-1}).

En cuanto al desarrollo de un ciclo de comunicación, este se lleva a cabo ejecutando el siguiente algoritmo:

1. El módulo "interfaz_pcps" construye el vector *buffer_in* y lo fragmenta en grupos de 8 bits.

2. "interfaz_pcps" envía el comando *scan* a través del puerto UART; esto coloca la máquina de estados en el lado de PS en el estado SCAN, preparada para recibir un byte de datos.

3. “interfaz_pcps” envía un byte de datos. Una vez este haya sido recibido por PS, la máquina de estados en transita automáticamente al estado IDLE, donde permanece en espera.

4. Si no quedan más bytes de datos por enviar (*i.e.*, si *buffer_in* ha sido completamente enviado), entonces “interfaz_pcps” envía el comando *calc* a través del UART, que mueve la máquina PS al estado CALC, donde inicia la comunicación con PL. En caso contrario, vuelve al segundo paso.

Comunicación PS – PL

La comunicación entre el procesador ARM y la FPGA embebida en el chip Zynq se realiza a través de un protocolo *handshake* (apéndice C) donde PS actúa como *master*, y se inicia a instancias del ordenador, cuando PS recibe el comando “calc”. Para llevar a cabo esta comunicación se ha implementado una máquina de estados en la FPGA, cuyo flujo está controlado por una serie de comandos predefinidos, que son enviados por PS. Esta comunicación consta de las siguientes etapas:

1. PS envía la señal (*cmd_scan*), y espera a recibir la misma respuesta *cmd_scan* desde PL; de esta forma PS confirma que PL se encuentra en el estado SCAN, dispuesta a leer la entrada de datos. A continuación se sigue una serie de ciclos *handshake* en los que PS envía *buffer_in_width* bits en paquetes de *data_width* bits a PL.

2. Una vez PS ha transmitido a PL el *buffer* de entrada, envía la señal *cmd_calc* a PL. Cuando la FPGA lee esta orden, se inicia la fase de cálculo, en la que el diseño digital realiza cualesquiera operaciones para las que ha sido diseñado, posiblemente utilizando la información introducida en el *buffer* de entrada (*buffer_in*) proveniente de PS (o, en última instancia, del ordenador). Cuando PL ha terminado las operaciones, envía *cmd_calc_sync* a PS. En este momento, PS responde con *cmd_idle*, lo cual envía PL al estado IDLE. Esta etapa termina cuando PL envía *cmd_idle_sync* a PS.

3. Finalmente, PS envía a PL la orden *cmd_print*, lo cual lleva a PL al estado PRINT; ahora se inicia un protocolo *handshake* en el que PL actúa como *master*, enviando un total de *buffer_out_width* bits en paquetes de *data_width* bits. Una

vez la transferencia ha terminado, PS formatea los bits leídos en un vector de enteros adecuado para ser transmitido al ordenador a través del puerto UART.

Tasa de producción de bits

El retardo en la medida de frecuencia de un oscilador (t^{med}) estará dado por los retardos característicos de cada fase del proceso, a saber: (*i*) procesamiento en el ordenador (t^{PC}), (*ii*) comunicación ordenador – PS ($t^{\text{PC-PS}}$), (*iii*) procesamiento en PS (t^{PS}), (*iv*) comunicación PS – PL ($t^{\text{PS-PL}}$), y (*v*) procesamiento en PL (t^{PL}):

$$t^{\text{med}} = t^{\text{PC}} + t^{\text{PC-PS}} + t^{\text{PS}} + t^{\text{PS-PL}} + t^{\text{PL}} \approx t^{\text{PC-PS}} + t^{\text{PL}} \tag{D.1}$$

De estos, los términos dominantes son la transmisión ordenador – PS a través del protocolo UART y el tiempo de cálculo de la frecuencia en la FPGA. En este trabajao, se ha utilizado una velocidad de transmisión estándar de $9600\,\text{bauds}$, *i.e.*, se envían 9600 bits por segundo entre el ordenador y el procesador PS; por lo tanto, el tiempo total para enviar *biw* bits y recibir *bow* bits será:

$$t^{\text{PC-PS}} = (biw + bow)/9600\,\text{s} \tag{D.2}$$

En los experimentos estándar llevados a cabo, el *buffer* de entrada consta de 8 bits para seleccionar qué oscilador medir de una matriz de 200 anillos, y 5 bits adicionales para seleccionar la resolución de la medida (*i.e.*, el número de ciclos del reloj de referencia M^{ref} definido en la sección 4.1); en total, $\text{biw} = 13$. Por otra parte, el *buffer* de salida es un entero de 32 bits, $\text{BOW} = 32$, de modo que $t^{\text{PC-PS}} = (13+32)/9600 \simeq 4{,}7\,\text{ms}$. En cuanto al tiempo de medida (t^{PL}), este es igual al periodo del reloj de referencia ($1/\nu^{\text{ref}}$) multiplicado por el número de ciclos de referencia,

$$t^{\text{PL}} = M^{\text{ref}}/\nu^{\text{ref}} \tag{D.3}$$

En general, utilizaremos un reloj de referencia proporcionado por el microprocesador ARM Cortex-A9 de $100\,\text{MHz}$, y un número de ciclos $M^{\text{ref}} = 2^{17} = 131072$, de modo que se tiene $t^{\text{PL}} \simeq 1{,}3\,\text{ms}$. Con todo esto, el tiempo total para medir la frecuencia característica de un oscilador será $t^{\text{med}} \simeq 6\,\text{ms}$; dado que se requieren dos mediciones para producir un bit mediante la técnica de medida compensada, la tasa de producción de bits (ν^{bit}) de esta solución será:

$$\nu^{\text{bit}} = \frac{1}{2\,t^{\text{med}}} \simeq 83{,}3\,\text{bits/s} \tag{D.4}$$

Programas y *scripts*

E

Aplicaciones y recursos *software* diseñados para esta tesis.

Documentación

`https://gudise.github.io/thesis-doc/`

Repositorio de *GitHub*

`https://github.com/gudise/thesis`

Lista de publicaciones propias

F

Publicaciones en revistas

[166] M. Garcia-Bosque, G. Díez-Señorans, A. Pérez-Resa, C. Sánchez-Azqueta, C. Aldea y S. Celma, "A 1 gbps chaos-based stream cipher implemented in 0.18 μm CMOS technology," *Electronics*, vol. 8, n.o 6, pág. 623, 2019.

[152] M. Garcia-Bosque, G. Díez-Señorans, C. Sánchez-Azqueta y S. Celma, "Proposal and analysis of a novel class of PUFs based on Galois ring oscillators," *IEEE Access*, vol. 8, págs. 157 830-157 839, 2020.

[129] G. Diez-Senorans, M. Garcia-Bosque, C. Sánchez-Azqueta y S. Celma, "Digitization algorithms in ring oscillator physically unclonable functions as a main factor achieving hardware security," *IEEE Access*, vol. 9, págs. 147 343-147 356, 2021.

[141] R. Aparicio-Téllez, M. Garcia-Bosque, G. Díez-Señorans y S. Celma, "Oscillator Selection Strategies to Optimize a Physically Unclonable Function for IoT Systems Security," *Sensors*, vol. 23, n.o 9, pág. 4410, 2023.

[164] M. Garcia-Bosque, A. Naya, G. Díez-Señorans, C. Sánchez-Azqueta y S. Celma, "Suitability of Generalized GAROs on FPGAs as PUFs or TRNGs Considering Spatial Correlations," *IEEE Open Journal of the Industrial Electronics Society*, vol. 4, págs. 112-122, 2023.

Publicaciones en congresos internacionales

[19] M. Garcia-Bosque, G. Díez-Señorans, C. Sánchez-Azqueta y S. Celma, "Introduction to physically unclonable fuctions: Properties and applications," en *2020 European Conference on Circuit Theory and Design (ECCTD)*, IEEE, 2020, págs. 1-4.

[167] G. Díez-Señorans, M. Garcia-Bosque, C. Sánchez-Azqueta y S. Celma, "A new approach to analysis the security of compensated measuring PUFs," en *2020 European Conference on Circuit Theory and Design (ECCTD)*, IEEE, 2020, págs. 1-5.

[168] M. Garcia-Bosque, G. Díez-Señorans, C. Sánchez-Azqueta y S. Celma, "FPGA implementation of a new PUF based on Galois ring oscillators," en *2021 IEEE 12th Latin America Symposium on Circuits and System (LASCAS)*, IEEE, 2021, págs. 1-4.

[169] G. Diez-Senorans, M. Garcia-Bosque, C. Sanchez-Azqueta y S. Celma, "Entropy Analysis of RO-based Physically Unclonable Functions," en *SMACD/PRIME 2021; International Conference on SMACD and 16th Conference on PRIME*, VDE, 2021, págs. 1-4.

[170] M. Garcia-Bosque, A. Naya, G. Diez-Senorans, C. Sanchez-Azqueta y S. Celma, "On the Behavior of a Wide Set of Oscillators: PUFs or TRNGs?" En *SMACD/PRIME 2021; International Conference on SMACD and 16th Conference on PRIME*, VDE, 2021, págs. 1-4.

[150] G. Diez-Senorans, M. Garcia-Bosque, C. Sánchez-Azqueta y S. Celma, "Programmable delay lines on different lut implementations for cro-puf," en *2022 17th Conference on Ph. D Research in Microelectronics and Electronics (PRIME)*, IEEE, 2022, págs. 357-360.

[171] M. Garcia-Bosque, R. Aparicio, G. Díez-Señorans, C. Sánchez-Azqueta y S. Celma, "An analysis of the behaviour of a PUF based on ring oscillators depending on their locations," en *2022 17th Conference on Ph. D Research in Microelectronics and Electronics (PRIME)*, IEEE, 2022, págs. 361-364.

[147] J. Fernández-Aragón, G. Díez-Señorans, M. Garcia-Bosque y S. Celma, "Design and characterisation of a Physically Unclonable Function on FPGA using second-order compensated measurement," en *2022 IFIP/IEEE 30th International Conference on Very Large Scale Integration (VLSI-SoC)*, IEEE, 2022, págs. 1-2.

[163] R. Aparicio-Tellez, M. Garcia-Bosque, G. Diez-Señorans y S. Celma, "Boosting the identifiability of a compensated measurement PUF via non-linear transformations," en *ECCTD*, 2023.

[172] A. Naya-Forcano, M. Garcia-Bosque, G. Díez-Señorans y S. Celma, "Multiprogram tools for FPGA boards with single identifier onWindows," en *2023 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, IEEE, 2023, págs. 1-4.

[173] R. Aparicio, J. Fernández-Aragón, A. Naya-Forcano, G. Díez-Señorans, M. Garcia- Bosque y S. Celma, "Proposal of a new PUF based on sensors for the identification of IoT smart mobile devices," en *PUF-enabled Security Challenge, European Cyber Security Awareness Week (CSAW)*, 2023.

Publicaciones en congresos nacionales

[174] G. Díez-Señorans, M. Garcia-Bosque, F. Aznar-Tabuenca, C. Sánchez-Azqueta y S. Celma, "Análisis de la seguridad criptográfica en capa física proporcionada por PUFs de medida compensada," en *VIII Congreso Nacional de I+D en Defensa y Seguridad*, Ministerio de Defensa del Gobierno de España, 2020.

[175] M. Garcia-Bosque, G. Díez-Señorans, F. Aznar-Tabuenca, C. Sánchez-Azqueta y S. Celma, "Propuesta de una nueva clase de funciones no clonables físicamente para comunicaciones seguras," en *VIII Congreso Nacional de I+D en Defensa y Seguridad*, Ministerio de Defensa del Gobierno de España, 2020.

[176] G. Díez-Señorans, M. Garcia-Bosque, F. Aznar-Tabuenca, C. Sánchez-Azqueta y S. Celma, "Mejora de la eficiencia de funciones no-clonables físicamente integrando líneas de retardo programables," en *IX Congreso Nacional de I+D en Defensa y Seguridad*, Ministerio de Defensa del Gobierno de España, 2022.

[177] M. Garcia-Bosque, G. Díez-Señorans, F. Aznar-Tabuenca, C. Sánchez-Azqueta y S. Celma, "Estrategias de selección de osciladores en una PUF de oscilador de anillo para optimizar su comportamiento," en *IX Congreso Nacional de I+D en Defensa y Seguridad*, Ministerio de Defensa del Gobierno de España, 2022.