CCN-CERT
centro criptológico nacional

CCN-TEC 012

El estándar de criptografía ligera, ASCON

CCN
centro criptológico nacional

Julio 2024

Edita:

CCN-PYTEC

Fecha de Edición: julio de 2024

El Grupo de investigación en Criptología y Seguridad de la Información (CiGSI) del Instituto de Tecnologías Físicas y de la Información "Leonardo Torres Quevedo" (ITEFI) del Consejo Superior de Investigaciones Científicas (CSIC), ha participado en el desarrollo del presente documento y sus anexos.

ÍNDICE

FIGURAS

TABLAS

1. INTRODUCCIÓN

1. A día de hoy, es cada vez más común y está más extendido el uso y despliegue de pequeños dispositivos electrónicos, ya sean las etiquetas de identificación por radiofrecuencia o RFID (*Radio Frequency IDentification*), pequeños sensores, controladores industriales, tarjetas inteligentes y dispositivos usables (*wearables*) como relojes o pulseras. Este cambio desde los ordenadores de sobremesa y portátiles hacia los pequeños dispositivos lleva aparejado grandes ventajas por cuanto su miniaturización permite llevarlos consigo, así como la gran diversidad de aplicaciones de todo tipo.

2. Sin embargo, también su uso masivo trae consigo otras preocupaciones, sobre todo las relacionadas con su seguridad y la privacidad de los datos que manipulan o almacenan. De hecho, estos dispositivos electrónicos se diseñan teniendo en cuenta, fundamentalmente, su aplicabilidad y usabilidad; mientras que apenas si se tiene en cuenta la protección de los datos que utilizan o la seguridad que puedan ofrecer a sus usuarios. La razón básica de esta carencia en el diseño de tales dispositivos se justifica por el hecho de que tienen recursos limitados, esto es, ofrecen poca capacidad de cómputo, escaso consumo y un limitado espacio de almacenamiento.

3. Por otra parte, es innegable la importancia que ha logrado la llamada Internet de las cosas o IoT (*Internet of Things*), cuyo crecimiento es mayor cada día. Es bien sabido que la IoT lo constituyen los dispositivos y sistemas que combinan la capacidad de adquirir, almacenar y acceder a cualquier tipo de información, y combinan estas características con la capacidad de conectarse e intercambiar información con otros dispositivos similares a través de redes de comunicación, como Internet, Wi-Fi, Bluetooth, etc. [12]. Por lo tanto, la IoT es la conexión en red de varios dispositivos físicos cotidianos que utilizan tecnologías de la información y las comunicaciones y se aplica en muchos campos diferentes.

4. Al margen de sus aplicaciones, también es importante considerar el aspecto de seguridad de la información gestionada por estos dispositivos. Si bien los dispositivos IoT relacionados con el usuario hacen que esta información sea accesible de manera conveniente, en general, su seguridad no se ha tenido en cuenta cuando tales dispositivos se diseñaron.

5. Es cierto que este hecho está cambiando y en la actualidad se considera la seguridad por diseño, es decir, el establecimiento de parámetros relacionados con la seguridad del dispositivo desde su primer paso de desarrollo. En todo caso, los dispositivos IoT relacionados con el usuario todavía implican el riesgo inherente de filtrar datos profesionales y personales [25]. Terceras partes, con fines maliciosos, podrían utilizar esta información con diferentes objetivos: fraude de identidad, falsificación de datos, robo económico, etc. En consecuencia, resulta de enorme importancia emplear sistemas de seguridad capaces de defender los sistemas y dispositivos electrónicos de cualquier tipo de ataque.

2. CRIPTOGRAFÍA LIGERA

6. Como es bien sabido, la criptografía ha sido tradicionalmente la ciencia responsable de garantizar la autenticidad, confidencialidad e integridad de los datos, ya sean almacenados o transmitidos. Ahora bien, en el caso de los dispositivos con escasos recursos, surge el problema de que tal criptografía no puede ser implementada porque tales dispositivos no soportan los algoritmos criptográficos al uso, debido a la gran cantidad de memoria, recursos computacionales y energía de procesamiento que requieren.

7. Para solucionar el problema de los limitados recursos que caracteriza a estos dispositivos electrónicos, y en particular a los que constituyen la IoT, y agregarles propiedades de seguridad, surgió la criptografía ligera.

8. Así pues, la criptografía ligera responde a las exigencias de seguridad en escenarios con recursos limitados de hardware y software, como las redes de sensores, las etiquetas RFID, los electrodomésticos inteligentes y, en general, los dispositivos de la IoT.

9. La criptografía ligera abarca una amplia gama de algoritmos criptográficos con diferentes propiedades y diversos casos de uso. De hecho, las principales propiedades que caracterizan a tales algoritmos son su reducida necesidad de cómputo, poco consumo de energía y su escasa capacidad de memoria.

10. Sin embargo, debe tenerse en cuenta que esta ligereza (de la criptografía) no debe ser sinónimo de una menor seguridad. El desafío de la criptografía ligera, por tanto, radica en mantener un elevado nivel de seguridad, equiparable al que ofrece la criptografía actual, pero requiriendo menos recursos de todo tipo. Así pues, el objetivo fundamental consiste en lograr el máximo equilibrio entre la mayor seguridad posible y el menor riesgo a ser vulnerable; manteniendo los costes (de memoria, energía, etc.) y el rendimiento (latencia, capacidad de procesado, etc.) tan bajos como sea posible. Tratando de evitar, además, que este tipo de dispositivo se convierta en una puerta trasera que permita poner en riesgo toda la red que une a los dispositivos conectados.

11. Dado que el desarrollo de nuevos algoritmos criptográficos ligeros seguros y eficientes es un reto importante para la comunidad criptográfica, se han propuesto dos convocatorias para seleccionar nuevos algoritmos: CAESAR (*Competition for Authenticated Encryption: Security, Applicability, and Robustness*), patrocinada por un grupo internacional de criptógrafos [3] y la convocatoria de Criptografía Ligera o LwC (*Lightweight Cryptography*) del NIST (*National Institute of Standards and Technology*) norteamericano [16].

12. El principal objetivo de estas convocatorias era el de establecer un nuevo sistema de cifrado ligero que fuera aceptado por toda la comunidad criptográfica y que sustituyera a los sistemas de cifrado empleados hasta ahora en los dispositivos IoT y otros con escasas capacidades de cómputo.

13. Debe tenerse en cuenta que, en la actualidad, el número de sistemas de cifrado ligeros empleados en este entorno es elevado. Entre los más importantes destacan SIMON, SPECK, PICCOLO, TWINE, PRESENT y KATAN. Cada uno tiene sus propias características, que no detallaremos en este informe, con sus ventajas e inconvenientes y se utilizan en las aplicaciones de la IoT más destacadas. Entre ellas destacamos las siguientes:

- Hogares inteligentes, en especial con el uso de los contadores inteligentes (*smart meters*) que permiten proteger la confidencialidad e integridad de la información intercambiada entre el contador de gas, electricidad, agua, etc. de un hogar y la compañía suministradora.

- La agricultura inteligente, que utiliza sensores para medir, por ejemplo, los datos relativos a la humedad o temperatura del suelo y tomar decisiones sobre los cultivos. En este caso, son importantes la integridad de los datos por cuanto es la base para la toma de decisiones sobre tales cultivos, y el bajo consumo de energía, debido a que estos dispositivos permanecen mucho tiempo aislados.

- Los dispositivos médicos implantables, como los *bypass*, bombas de insulina, etc., para los que es imprescindible garantizar que solo puede acceder a ellos quien esté autorizado. En este caso, se trata de mantener la privacidad de los datos del paciente de modo que no se haga pública su dolencia y, también, evitar que tales dispositivos puedan ser atacados por un adversario que podría alterar el consumo de la batería, disminuyendo drásticamente su vida útil, o modificar las dosis del fármaco establecida.

- Las denominadas tecnologías vestibles o usables (*wearables*) como las pulseras, relojes, pulsómetros, etc. que pueden medir la humedad de la piel, el ritmo cardíaco y otros valores asociados a la salud de quien los viste. Estos dispositivos deben mantener la confidencialidad de los datos que miden y transmiten.

- Los microcontroladores industriales, como por ejemplo, los empleados en la industria automovilística, ya sea en motores, controles de transmisión, sistemas de frenos y de dirección, y en especial en los coches eléctricos e híbridos. En este caso, es importante que los datos suministrados por los diferentes sensores a la central del vehículo sean los reales y no hayan sido modificados por un atacante, de modo que no se produzca un mal funcionamiento del mismo, lo que puede poner en peligro la vida de los ocupantes.

3. CONVOCATORIA LwC DEL NIST

14. Después de una serie de reuniones que se iniciaron en 2013, el NIST comenzó un proceso para solicitar, evaluar y estandarizar esquemas que proporcionaran cifrado autenticado con datos asociados o AEAD (*Authenticated Encryption with Associated Data*) y funcionalidades de hash opcionales. Todo ello para entornos con capacidades restringidas, donde el rendimiento de los estándares criptográficos del NIST de la época no era aceptable. De hecho, en 2018, el NIST publicó una convocatoria para seleccionar nuevos algoritmos ligeros como futuros estándares. En esta convocatoria se describían los requisitos, el proceso de selección y los criterios de evaluación que se tendrían en cuenta para tal selección [16].

15. Debe tenerse en cuenta que los AEAD protegen la confidencialidad de un mensaje, pero también permiten incluir información adicional, como el encabezado de un mensaje o la dirección IP de un dispositivo, sin cifrarla. Este tipo de algoritmo garantiza que todos los datos protegidos son auténticos y no han cambiado durante el tránsito. Algunas aplicaciones de los AEAD se centran en las comunicaciones entre vehículos o en la ayuda a la prevención de la falsificación de mensajes intercambiados con RFID. El AEAD más eficiente aprobado por el NIST es el AES (*Advanced Encryption Standard*) publicado en [13] y utilizado con el modo Galois/Contador [14].

16. La convocatoria LwC del NIST no pretendía reemplazar los estándares establecidos por el propio instituto, dado que los sigue recomendando en los dispositivos que no tienen limitaciones de recursos, y también por cuestiones de interoperabilidad. Tampoco los nuevos algoritmos ligeros estarían destinados a ser utilizados para el cifrado postcuántico (ver [4] y [5]).

17. Con relación al desarrollo de la convocatoria LwC, conviene recordar que en marzo de 2019, el NIST recibió 57 propuestas para ser consideradas en el proceso de estandarización. La primera ronda de este proceso finalizó en agosto de 2019 y en [17] se presentó la evaluación de las propuestas presentadas. De los candidatos presentados, solo 32 de ellos pasaron a la segunda ronda.

18. La segunda ronda acabó en marzo de 2021, cuando los 10 finalistas se anunciaron en [18]. Estos finalistas fueron ASCON, Elephant, GIFT-COFB, Grain-128AEAD, ISAP, PHOTON-Beetle, Romulus, SPARKLE, TinyJAMBU y Xoodyak.

19. La ronda final de este proceso de estandarización finalizó con la elección de la familia ASCON como estándar, en febrero de 2023. En [19] se describe la evaluación de los 10 finalistas que se habían considerado y se explican las razones de la elección de la familia ASCON como ganadora. Según el propio NIST, los algoritmos elegidos están diseñados para proteger la información creada y transmitida por la IoT, incluidos sus sensores y actuadores, dispositivos médicos implantables, detectores de estrés y controles remotos de entrada sin llave para vehículos.

4. ASCON

20. ASCON [7] fue desarrollado en 2014 por un equipo de criptógrafos de la Universidad Tecnológica de Graz, de Infineon Technologies, de Lamarr Security Research y de la Universidad de Radboud. En 2019 fue seleccionado como la principal opción para el cifrado autenticado ligero en la fase final de la convocatoria CAESAR [6], lo que pone de manifiesto que ASCON es resistente a los análisis realizados durante este proceso.

21. La familia ASCON originalmente constaba de siete miembros, pero no todos ellos han pasado a formar parte del estándar de criptografía ligera publicado por el NIST. De hecho, como se comentará más adelante, los principales usos aprobados de ASCON son para el cifrado autenticado con datos asociados y como función hash.

22. Como se sabe, las funciones hash proporcionan un resumen de un mensaje a modo de huella digital corta que permite determinar si el mensaje ha sido modificado. El estándar actual de función hash publicado por el NIST es SHA-3 [15]. En criptografía ligera las funciones hash se pueden utilizar para comprobar si una actualización de software es apropiada o si se ha descargado correctamente.

23. Una limitación a destacar de las variantes de ASCON es la ausencia de una opción que permita el empleo de claves de 256 bits, lo que puede ser un problema cuando se necesita seguridad de 128 bits contra ataques cuánticos. Recuérdese que el algoritmo cuántico de Grover (véanse [8] y [9]) permite reducir el tamaño de la clave a la mitad.

24. En todo caso, es de destacar que una de las variantes de ASCON ofrece cierta resistencia a la computación cuántica, como se comentará más adelante.

4.1. DESCRIPCIÓN DE ASCON COMO SISTEMA DE CIFRADO

25. En este apartado se muestra la descripción del funcionamiento de ASCON como sistema de cifrado. No se pretende hacer una presentación excesivamente minuciosa, pero sí dar a conocer sus principales características y propiedades a través de la descripción de sus algoritmos.

26. A continuación se presentan con más detalle las cuatro fases que constituyen el cifrado ASCON y que ya se comentaron en la Sección 4.1.

27. En las Figuras 1 y 2 se muestran los procesos de cifrado y descifrado de ASCON, respectivamente. Como se puede apreciar en las mismas, solo las terceras fases son diferentes. Estas fases se comentan con más detalle en las secciones siguientes.

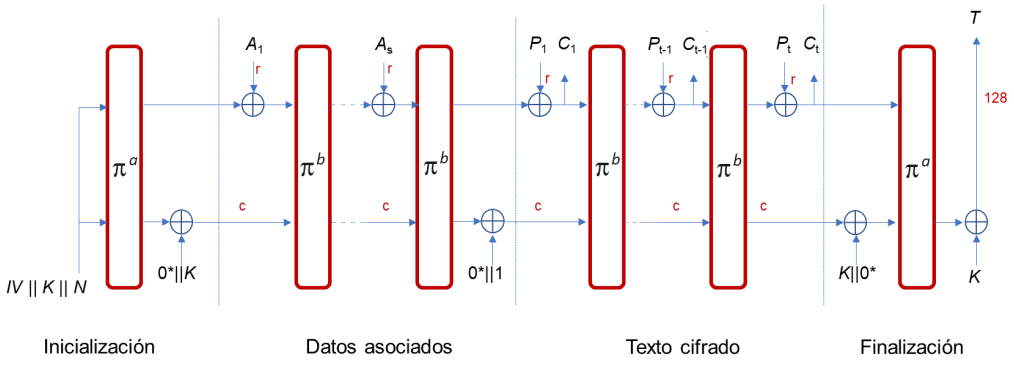

Figura 1: Esquema de las cuatro fases para el proceso de cifrado

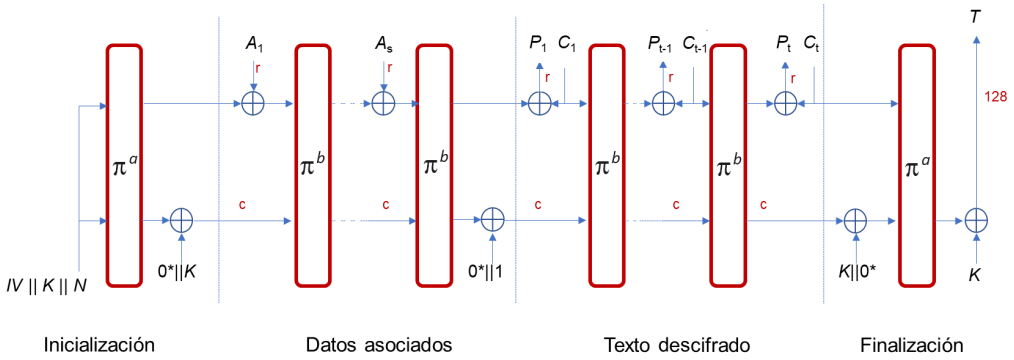

Figura 2: Esquema de las cuatro fases para el proceso de descifrado

4.1.1. INICIALIZACIÓN

28. Antes de proceder al procesamiento de los bloques de información, ASCON inicializa su estado interno mediante un valor inicial que se encuentra directamente vinculado al vector de inicialización o IV (ver Figuras 1 y 2). Este vector toma diferentes valores en función de la variante específica del algoritmo considerado, tal como se ilustra a continuación.

29.

$$IV_{k,r,a,b} \leftarrow k||r||a||b||0^{160-k} = \begin{cases} \text{80400c0600000000}, & \text{para ASCON-128} \\ \text{80800c0800000000}, & \text{para ASCON-128a} \\ \text{a0400c06}, & \text{para ASCON-80pq} \end{cases}$$

30. La fase de inicialización de ASCON se configura mediante la concatenación del vector inicial fijo con la clave K y el nonce N, seguida de la aplicación a veces de la permutación π, como se expone en el Algoritmo 1.

31. De forma más detallada, el Algoritmo 1 consta de los siguientes pasos:

    1. En el primer paso se concatena el vector de inicialización IV con la clave K y con el nonce N, cuyo resultado se almacenada en el estado interno de ASCON, S.

Algoritmo 1 Inicialización

1: $S \leftarrow IV||K||N$
2: $S \leftarrow \pi^a(S) \oplus (0^{320-k}||K)$

2. En el segundo y último paso se lleva a cabo una operación XOR entre el resultado de aplicar a veces la permutación π al registro de estado, S, y la concatenación de un conjunto de ceros, que está determinado por la extensión de la clave, con la propia clave K.

4.1.2. PROCESADO DE LOS DATOS ASOCIADOS

32. El procesado de los datos asociados, A, comienza su subdivisión en bloques de r bits. El valor de r depende de la variante específica del algoritmo de cifrado autenticado que se emplee, como se señala en la Tabla 1. Los bloques se van insertando de forma sucesiva en el registro de estado, S, mediante la aplicación de la operación XOR. A medida que se introducen los bloques, se ejecuta la permutación π durante b rondas. La descripción detallada de este proceso se presenta en el Algoritmo 2 (ver Figuras 1 y 2).

Algoritmo 2 Procesado de los datos asociados

1: **if** $|A| > 0$ **then**
2: $(A_1 \dots A_s) \leftarrow A$ se divide en bloques de r bits $(A||1||0^*)$
3: **for** $i = 1 \dots s$ **do**
4: $S \leftarrow \pi^b\left((S_r \oplus A_i)||S_c\right)$
5: **end for**
6: **end if**
7: $S \leftarrow S \oplus \left(0^{319}||1\right)$

33. Los pasos que sigue el Algoritmo 2 son los siguientes:

1. Los datos asociados, representados como A, se dividen en bloques de r bits, A_i, $1 \leq i \leq s$. En el caso de que el último bloque no esté completo, se lleva a cabo el relleno necesario (*pad*). Este relleno se hace añadiendo un 1 como primer bit después del último bit del bloque y se concatena con tantos ceros como sea preciso hasta alcanzar el tamaño de r bits en el último bloque. Es lo que se ha denotado como $A||1||0^*$.

2. En el proceso iterativo, en el registro de estado, S, cada bloque es inyectado mediante una operación XOR entre la parte interna del estado, S_r, y el bloque de datos asociados A_i. El resultado de esta operación XOR se concatena con la parte externa del estado, S_c.

3. El estado se actualiza ejecutando b veces la permutación π.

4. Todos los pasos anteriores se repiten para cada uno de los bloques de datos asociados A_i.

5. Por último, en el registro de estado se almacena la operación XOR del estado S con una cadena de 319 ceros y un 1 como el bit menos significativo.

6. El resultado final es el estado obtenido, S.

4.1.3. PROCESADO DEL TEXTO CLARO (CIFRADO)

34. El procesado del texto claro, P, es de hecho, la operación de cifrado. El procedimiento que se sigue es análogo al de los datos asociados. La diferencia entre ambos radica en que, en este caso, la salida obtenida después del procesamiento corresponde al texto cifrado. El algoritmo utilizado para este propósito se especifica en el Algoritmo 3 (ver Figura 1).

Algoritmo 3 Procesado del texto claro (algoritmo de cifrado)

1: $(P_1 \dots P_t) \leftarrow P$ se divide en bloques de r bits $(P||1||0^*)$
2: **for** $i = 1 \dots t-1$ **do**
3: $\quad S_r \leftarrow S_r \oplus P_i$
4: $\quad C_i \leftarrow S_r$
5: $\quad S \leftarrow \pi^b(S)$
6: **end for**
7: $S_r \leftarrow S_r \oplus P_t$
8: $C_t \leftarrow \lfloor S_r \rfloor_{|P_t|}$

35. Esta fase corresponde con el proceso de cifrado y sigue los siguientes pasos:

1. El texto claro, P, se divide en bloques de r bits, P_t. Al igual que en el Algoritmo 3, si el último bloque el texto cifrado está incompleto, se añade un 1 como bit siguiente al último bit del bloque y luego se completa este bloque con tantos ceros como sea necesario hasta que el último bloque tenga una extensión de r bits.

2. El registro de estado, S, se actualiza con los bloques resultantes de la operación XOR entre la parte interna del registro, S_r, y los bloques de texto claro, P_i. Dicha operación da como resultado el bloque de texto cifrado C_i.

3. Cada vez que se obtiene un bloque de texto cifrado correspondiendo con el estado interno, se procede a actualizar la totalidad del estado mediante la aplicación de la permutación π un total de b veces.

4. Los pasos anteriores se repiten para cada bloque de texto claro, excepto para el último bloque de r bits.

5. Si es necesario, se trunca la longitud del último bloque de texto claro, P_t, de modo que el último bloque de texto cifrado, C_t, haga que la longitud del texto cifrado, C, sea exactamente la misma que la del texto claro, P.

6. Tras obtener el último bloque de texto cifrado no se aplica la permutación, como con los bloques anteriores.

7. La salida del algoritmo es el texto cifrado, $(C_1 \dots C_t)$.

4.1.4. PROCESADO DEL TEXTO CIFRADO (DESCIFRADO)

36. El procesado del texto cifrado tienen una gran similitud con el procesamiento del texto claro. Debe tenerse en cuenta que ASCON es un sistema de cifrado simétrico, por lo que esta característica era de esperar. La única diferencia entre este proceso de descifrado y el de cifrado reside en el orden en el que se ejecutan las operaciones, tal y como se muestra en el Algoritmo 4 (ver Figura 2).

Algoritmo 4 Procesado del texto cifrado (algoritmo de descifrado)

1: $(C_1 \dots C_t) \leftarrow C$ se divide en bloques de r bits $(C||1||0^*)$
2: **for** $i = 1 \dots t-1$ **do**
3: $\quad P_i \leftarrow S_r \oplus C_i$
4: $\quad S \leftarrow C_i||S_c$
5: $\quad S \leftarrow \pi^b(S)$
6: **end for**
7: $P_t \leftarrow \lfloor S_r \rfloor_{|C_t|} \oplus C_t$
8: $S_r \leftarrow S_r \oplus (P_t||1||0^*)$

37. El algoritmo de descifrado, es decir, el procesado del texto cifrado sigue el siguiente procedimiento:

    1. El texto cifrado, C, se divide en bloques de r bits. Como en los casos anteriores, si el último bloque resulta ser incompleto, se incorpora un relleno que consiste en un 1 colocado inmediatamente después del último bit del bloque, seguido por un conjunto de ceros hasta alcanzar la longitud de un bloque de r bits.

    2. La operación de descifrado se lleva a cabo mediante la ejecución de una operación XOR entre la parte interna del estado, S_r y cada bloque cifrado C_i. El resultado de esta operación proporciona cada uno de los bloques del texto claro, P_i.

    3. El valor del registro de estado durante el proceso de cifrado se recupera introduciendo el texto cifrado, C_i, en la parte interna del registro de estado, S_r, y luego ejecutando b veces la permutación π.

    4. Estos pasos se repiten con todos los bloques del texto cifrado, C_i, a excepción del último.

    5. Mediante la operación XOR entre el último bloque de texto cifrado, C_t, y la parte interna del registro de estado S_r, se obtiene el último bloque de texto plano, P_t.

    6. Finalmente, en la sección interna del estado, S_r, se incorpora el resultado de la operación XOR entre esta parte del estado y el último bloque de texto plano, P_t (con su correspondiente relleno, caso de ser necesario).

    7. Esta última actualización del estado es esencial para finalizar adecuadamente el proceso de finalización, donde se verifica o no el MAC/tag a generar.

4.1.5. FINALIZACIÓN

38. En la fase de finalización se distinguen dos posibles escenarios (ver Figuras 1 y 2).

39. En el primero de ellos, el objetivo de la ejecución del algoritmo es cifrar un mensaje. En este caso, el procedimiento de finalización genera y transmite una etiqueta o *tag*, T. La generación de esta etiqueta depende de los procesos previos llevados a cabo.

40. Por el contrario, si el propósito de la ejecución del algoritmo es descifrar y autenticar un mensaje, el proceso de finalización compara la etiqueta generada después del proceso de descifrado con la etiqueta recibida, que se generó previamente durante el proceso de cifrado.

41. El algoritmo encargado de calcular la etiqueta se encuentra detallado en el Algoritmo 5.

Algoritmo 5 Finalización

1: $S \leftarrow \pi^a \left(S \oplus (0^r || K || 0^{c-k})\right)$
2: $T \leftarrow \lceil S \rceil^{128} \oplus \lceil K \rceil^{128}$

42. De forma más detallada, la fase de finalización consta de los siguientes dos pasos:

    1. El estado, S, se actualiza mediante la ejecución de la permutación π un total de a veces al resultado de la operación XOR entre el registro de estado y una concatenación de r ceros, la clave K, y el número de ceros restantes necesario para completar los 320 bits del estado.
    2. La etiqueta T se genera mediante la operación XOR entre los r bits menos significativos del estado, S, y los r bits menos significativos de la clave K.

4.2. ASCON COMO CIFRADO AUTENTICADO CON DATOS ASOCIADOS

43. ASCON (ver [6] y [7]) es un AEAD basado en permutaciones y un esquema hash. La componente principal de la familia ASCON es una permutación de 320 bits que se lleva a cabo con diferentes constantes y diferente número de rondas para cada una de las variantes de la familia.

44. ASCON, como cifrado autenticado, es un criptosistema simétrico por bloques (en el Apartado 4.1 se presenta una descripción pormenorizada de este algoritmo como sistema de cifrado simétricos por bloques) y consta de dos algoritmos, uno de cifrado, $\mathcal{E}_{k,r,a,b}$, y otro de descifrado, $\mathcal{D}_{k,r,a,b}$. Los parámetros empleados son los siguientes:

    - k: longitud de la clave, de modo que $k \leq 160$ bits.
    - r: tamaño de cada uno de los bloques.
    - a: número de rondas de la permutación en las fases de inicialización y finalización (π^a).
    - b: número de rondas de la permutación en la fase intermedia (π^b).

45. ASCON-AEAD utiliza la construcción monoDuplex [2] en la que se añaden claves adicionales durante los procesos de inicialización y finalización y consta de las siguientes cuatro fases:

    - Inicialización: da lugar al inicio del estado con dos valores, el tamaño de la clave, k, y el *nonce*, N.
    - Procesado de datos asociados: se actualiza el estado de 320 bits con los bloques de datos asociados.
    - Procesado del texto claro (cifrado) o del texto cifrado (descifrado): inyecta los bloques del texto claro, P_i, o del texto cifrado, C_i, en el estado y extrae los bloques del texto cifrado, C_i, o del texto claro, P_i, según se trate de uno yu otro algoritmo.
    - Finalización: hace uso, de nuevo, de la clave y extrae la etiqueta (*tag*) para la autenticación.

46. La entrada del algoritmo de cifrado, $\mathcal{E}_{k,r,a,b}$, toma k bits de la clave K, 128 bits del nonce N, un dato asociado, A, de longitud arbitraria y un texto claro, P, de longitud arbitraria. Su salida consta de un texto cifrado autenticado, C, de la misma longitud que el texto claro, P, además de una etiqueta de autenticación de 128 bits, T, que autentica tanto los datos asociados como el mensaje cifrado, es decir,

$$\mathcal{E}_{k,r,a,b}(K, N, A, P) = (C, T).$$

47. En el caso del algoritmo de descifrado, $\mathcal{D}_{k,r,a,b}$, se considera como entrada la clave K, el nonce N, los datos asociados A, el texto cifrado C y el etiqueta T. La salida del algoritmo devuelve el texto claro, P, si el proceso de verificación de la etiqueta T es correcto y devuelve un error, $\perp$, si el proceso de verificación falla, de modo que:

$$\mathcal{D}_{k,r,a,b}(K, N, A, C, T) \in \{P, \perp\}.$$

48. La Tabla 1 muestra los diferentes conjuntos de parámetros que definen las variantes de ASCON aprobadas por el NIST como sistema de cifrado. Los tamaños de la clave, k, del nonce, N, de la etiqueta o tag, T, y de los bloques, r, se expresa en bits; mientras que los parámetros a y b denotan el número de rondas en las fases inicial y final y de las fases intermedias, respectivamente.

Variante	Tamaño (en bits) de				Rondas	
	Clave (K)	Nonce (N)	Tag (T)	Bloque (r)	a	b
ASCON-128	128	128	128	64	12	6
ASCON-128a	128	128	128	128	12	8
ASCON-80pq	160	128	128	64	12	6

Tabla 1: **Variantes de ASCON-AEAD con sus parámetros asociados**

49. Las dos primeras variantes de ASCON-AEAD, ASCON128 y ASCON128a, proporcionan seguridad de 128 bits tanto para la confidencialidad del texto claro como para su integridad y la integridad del texto claro y los datos asociados. Por su parte, la versión ASCON80pq ofrece una seguridad de 160 bits, considerándose resistente a la computación cuántica.

50. El número de bloques de texto claro y datos asociados protegido por el algoritmo de cifrado está limitado hasta un total de 2^{64} bloques por clave.

4.3. ASCON COMO FUNCIÓN HASH

51. Por su parte, ASCON-HASH utiliza la construcción esponja [1], un tipo de funciones similares a las empleadas en SHA-3 y SHAKE.

52. Existen dos variantes hash de ASCON, una es como función hash con un resumen fijo de 256 bits, ASCON-HASH, y la otra es ASCON-XOF, que proporciona un resumen con una salida de extensión variable, ℓ. El número de rondas para las variantes de hash se representa mediante cuatro valores: el número de rondas durante la inicialización, i, la absorción del mensaje, a, la compresión del primer bloque, c, y la compresión de los bloques restantes, b, respectivamente.

53. La Tabla 2 muestra los diferentes conjuntos de parámetros que definen las variantes de ASCON aprobadas por el NIST como función hash. Se muestran el tamaño, en bits, del resumen, del bloque, de la capacidad, y del número de rondas, respectivamente.

Variante	Tamaño (en bits) de			Rondas			
	Resumen	Bloque	Capacidad	i	a	c	b
ASCON-HASH	256	64	256	12	12	12	12
ASCON-HASHa	256	64	256	12	8	12	8
ASCON-XOF	ℓ	64	256	12	12	12	12
ASCON-XOFa	ℓ	64	256	12	8	12	8

Tabla 2: Variantes de ASCON-HASH con sus parámetros asociados

54. ASCON-HASH y ASCON-HASHa proporcionan seguridad de 128 bits contra ataques de colisión y ataques a la segunda preimagen; mientras que ASCON-XOF y ASCON-XOFa, con un tamaño de salida de ℓ bits, proporcionan una seguridad mínima de $(128, \ell/2)$ bits contra ataques de colisión y una seguridad mínima de $(128, \ell)$ bits contra ataques a la segunda imagen.

4.4. SEGURIDAD DE ASCON

55. Los ataques más importantes que se han publicado contra la seguridad de ASCON pueden dividirse en dos grandes grupos: los ataques algebraicos y los de canal lateral.

56. Los criptoanálisis diferencial y lineal son los ataques algebraicos más utilizados para intentar vulnerar los cifradores en bloque y en flujo, y las funciones hash [10]. El criptoanálisis diferencial estudia cómo determinados cambios en la entrada de un algoritmo afectan a su salida. Su objetivo es encontrar estrategias que permitan identificar las diferencias que se producen en la salida a través de una colección de transformaciones en la entrada, para determinar en qué partes del algoritmo de cifrado este muestra un comportamiento que no es aleatorio.

57. Por otra parte, los ataques por canal lateral intentan obtener información a partir de las implementaciones hechas en dispositivos criptográficos, como son los ataques por análisis de tiempo, de consumo de potencia, de radiaciones electromagnéticas, etc.

58. En [7] se comentan los principales ataques que se han llevado a cabo contra ASCON, tanto en su faceta de cifrador en bloque como en la de función hash. En este caso, las recomendaciones más importantes señalan que las implementaciones deben asegurarse no repetir un mismo nonce para dos cifrados bajo una misma clave y también que el número de bloques del texto claro y datos asociados no sea cercano al límite protegido por el algoritmo de cifrado que, como ya se ha mencionado, está limitado a 2^{64} bloques por clave.

59. A modo de resumen, se puede afirmar que los mejores ataques de recuperación de claves (*key-recovery*) se llevan a cabo con las variantes AEAD de ASCON con una inicialización de 7 rondas, de las 12 establecidas [24]; mientras que el mejor ataque a las variantes hash de ASCON es un ataque de preimagen que solo considera 4 rondas [20].

60. Los ataques de este tipo más sofisticados unen los ataques por análisis de potencia (*power analysis*) con técnicas de aprendizaje automático (*machine learning*) (ver [11] y [21]).

61. Así pues, se puede afirmar que ninguno de los ataques mencionados vulnera los requerimientos de seguridad establecidos por el NIST, por lo tanto, la familia ASCON presenta un alto grado de seguridad.

4.5. IMPLEMENTACIONES DE ASCON

62. Como ya se ha comentado, ASCON está especialmente diseñado para proporcionar una implementación eficiente, tanto en hardware como en software. Además, ASCON es independiente de otros algoritmos de cifrado, no requiere aritmética en cuerpos finitos ni otro tipo de operaciones matemáticas complejas. A continuación se presenta la Tabla 3, donde se recogen las principales implementaciones, en ciclos por byte (c/B) para mensajes largos, desarrolladas en lenguaje C y que están recomendadas en la web oficial de ASCON.

Implementación	ASCON-128 y ASCON-80pq	ASCON-128a
AMD EPYC 7742	6.5	4.2
AMD Ryzen 9 5950X	8.1	5.2
Apple M1 (ARMv8)	9.3	6.3
Cortex-A72 (ARMv8)	10.5	7.0
Intel Xeon E5-2609 v4	10.6	7.2
Intel Core i5-6300U	11.4	7.8
Intel Core i5-4200U	15.8	10.6
Cortex-A9 (ARMv7)	33.3	24.0
Cortex-A7 (NEON)	46.5	30.7
Cortex-A7 (ARMv7)	57.2	41.2
ARM1176JZF-S (ARMv6)	56.8	42.9

Tabla 3: Implementaciones de ASCON en lenguaje C en ciclos por byte (c/B)

63. Dada la importancia de los tiempos de las implementaciones, se ha realizado una comparativa de distintas implementaciones de ASCON y la comparativa con otros algoritmos de cifrado. En el Anexo A se puede encontrar una comparativa entre distintas versiones de ASCON y en el Anexo B se pueden encontrar las comparativa de las implementaciones de ASCON y de otros algoritmos de cifrado.

5. REFERENCIAS

[1] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In *Proc. International Conference on Selected Areas in Cryptography (SAC 2011), Lecture Notes Comput Sci.*, volume 7118, pages 320–337, 2012. https://doi.org/10.1007/978-3-642-28496-0_19. 17

[2] Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. Permutation-based encryption, authentication and authenticated encryption. *Keccack Team*, pages 1–12, 2012. https://keccak.team/files/KeccakDIAC2012.pdf. 16

[3] CAESAR. *CAESAR call for submissions*. Competition for Authenticated Encryption: Security, Applicability, and Robustness, 2014. https://competitions.cr.yp.to/caesar-call.html. 7

[4] CCN. *Recomendaciones para una transición postcuántica segura, CCN-TEC 009*. Centro Criptológico Nacional, Ministerio de Defensa, España, 2022. https://www.ccn.cni.es/index.php/es/docman/documentos-publicos/boletines-pytec/495-ccn-tec-009-recomendaciones-transicion-postcuantica-segura/file. 9

[5] CCN. *Guía de Mecanismos Criptográficos autorizados por el CCN, CCN-STIC 221*. Centro Criptológico Nacional, Ministerio de Defensa, España, 2023. https://www.ccn-cert.cni.es/series-ccn-stic/guias-de-acceso-publico-ccn-stic/6954-ccn-stic-221-guia-de-mecanismos-criptograficos-autorizados-por-el-ccn-1/file.html. 9

[6] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schlaffer. Ascon v1.2. submission to the CAESAR competition. Technical report, NIST, 2016. https://ascon.iaik.tugraz.at/files/asconv12.pdf. 10, 15

[7] Christoph Dobraunig, Florian Mendel, Maria Eichlseder, and Martin Schlaffer. Status update on ascon v1.2, update to the NIST lightweight cryptography standardization process. Technical report, NIST, 2022. https://csrc.nist.gov/csrc/media/Projects/lightweight-cryptography/documents/fnalist-round/status-updates/ascon-update.pdf. 10, 15, 18

[8] L.K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79(2):325–328, 1997. https://doi.org/10.1103/PhysRevLett.79.325. 10

[9] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. 28^{th} annual ACM symposium on Theory of Computing (STOC'96)*, pages 212–219, 1996. https://doi.org/10.1145/800070.802214. 10

[10] Howard M. Heys. A tutorial on linear and differential cryptoanalysis. Technical report, Memorial University of Newfoundland, 2017. http://www.cs.bc.edu/~straubin/crypto2017/heys.pdf. 18

[11] Priyanka Joshi and Bodhisatwa Mazumdar. A subset fault analysis of ASCON. *Cryptology ePrint Archive, Report 2019/1370*, 2019. https://eprint.iacr.org/2019/1370. 18

[12] Shancang Li, Li Da Xu, and Shanshan Zhao. The internet of things: A survey. *Information Systems Frontiers*, 17(2):243–259, 2015. https://doi.org/10.1007/s10796-014-9492-7. 6

[13] NIST. *Advanced Encryption Standard*. National Institute of Standard and Technology, Federal Information Processing Standard Publication, FIPS 197, 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. 9

[14] NIST. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*. National Institute of Standards and Technology, Special Publication SP 800–38D, 2007. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf. 9

[15] NIST. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. National Institute of Standard and Technology, NIST FIPS 202, 2015. http://csrc.nist.gov/publications/drafts/fips-202/fips_202_draft.pdf. 10

[16] NIST. *Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process*. National Institute of Standards and Technology, 2018. https://csrc.nist.gov/csrc/media/Projects/lightweight-cryptography/documents/final-lwc-submission-requirements-august2018.pdf. 7, 9

[17] NIST. *Status Report on the First Round of the NIST Lightweight Cryptography Standardization Process, NIST IR 8268*. National Institute of Standards and Technology, 2019. https://csrc.nist.gov/pubs/ir/8268/final. 9

[18] NIST. *Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process, NIST IR 8369*. National Institute of Standards and Technology, 2021. https://csrc.nist.gov/pubs/ir/8369/final. 9

[19] NIST. *Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process, NIST IR 8454*. National Institute of Standards and Technology, 2023. https://nvlpubs.nist.gov/nistpubs/ir/2023/NIST.IR.8454.pdf. 9, 30

[20] Lingyue Qin, Jialiang Hua, Xiaoyang Dong, Hailun Yan, and Xiaoyun Wang. Meet-in-the-middle preimage attacks on sponge-based hashing. In *Proc. Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT 2023, Lecture Notes Comput. Sci.*, volume 14007, pages 158–188, 2023. https://doi.org/10.1007/978-3-031-30634-1_6. 18

[21] Keyvan Ramezanpour, Paul Ampadu, and William Diehl. SCARL: Side-channel analysis with reinforcement learning on the ascon authenticated cipher. *arXiv preprint*, 2020. https://doi.org/10.48550/arXiv.2006.03995. 18

[22] Sebastian Renner, Enrico Pozzobon, and Jürgen Mottok. The final round: Benchmarking NIST LWC ciphers on microcontrollers. In *Proc. Attacks and Defenses for the Internet-of-Things (ADIoT 2022), Lecture Notes Comput. Sci.*, volume 13745, pages 1–20, 2022. https://doi.org/10.1007/978-3-031-21311-3_1. 23

[23] Sebastian Renner, Enrico Pozzobon, and Jürgen Mottok. *NIST LWC Software Performance Benchmarks on Microcontrollers*. Centro Criptológico Nacional, Ministerio de Defensa, España, 2023. https://lwc.las3.de/. 23

[24] Raghvendra Rohit, Kai Hu, Sumanta Sarkar, and Siwei Sun. Misuse-free key-recovery and distinguishing attacks on 7-round Ascon. In *IACR Trans. Symmetric Cryptol.*, volume 1, pages 130–155, 2021. https://doi.org/10.46586/tosc.v2021.i1.130-155. 18

[25] Ghaidaa Shaabany and Reiner Anderl. Security by design as an approach to design a secure industry 4.0-capable machine enabling online-trading of technology data. In *Proc. 2018 International Conference on System Science and Engineering (ICSSE)*, pages 1–5, 2018. https://doi.org/10.1109/ICSSE.2018.8520195. 6

A. ANEXO A. COMPARATIVA ENTRE LAS VERSIONES DE ASCON

64. A modo de ejemplo y con el fin de comparar algunas implementaciones de las diferentes versiones de ASCON en distintos dispositivos, se incluyen a continuación algunas gráficas. Los datos originales para la elaboración de las mismas se han obtenido de [22] y [23].

65. Las Figuras 3 y 4 presentan el tiempo medio de ejecución, en microsegundos [μs], para diferentes implementaciones de ASCON128 en un Arduino UNO y en un STM32F1, respectivamente.

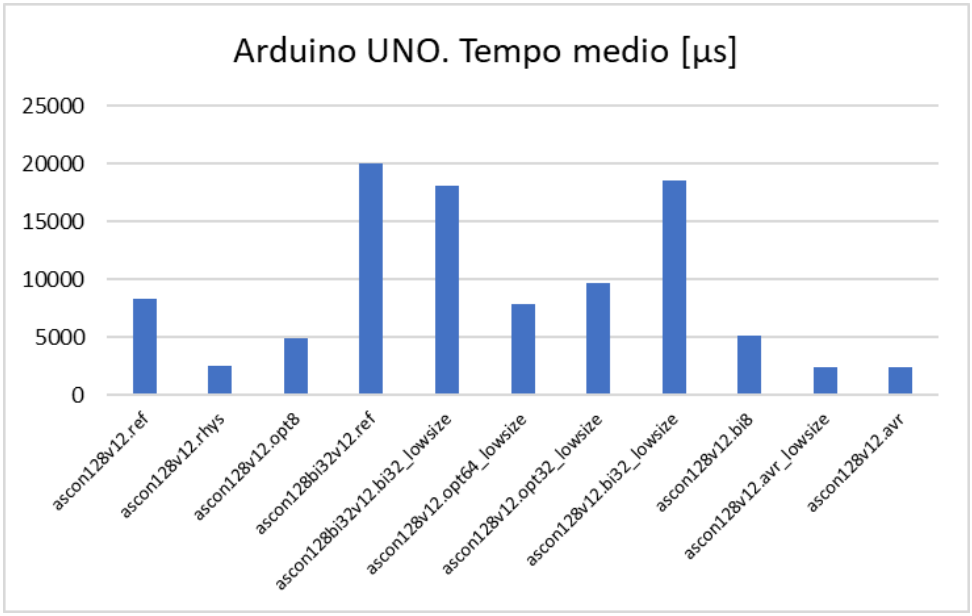

Figura 3: Tiempo medio de ejecución en un Arduino UNO para diferentes implementaciones de ASCON128

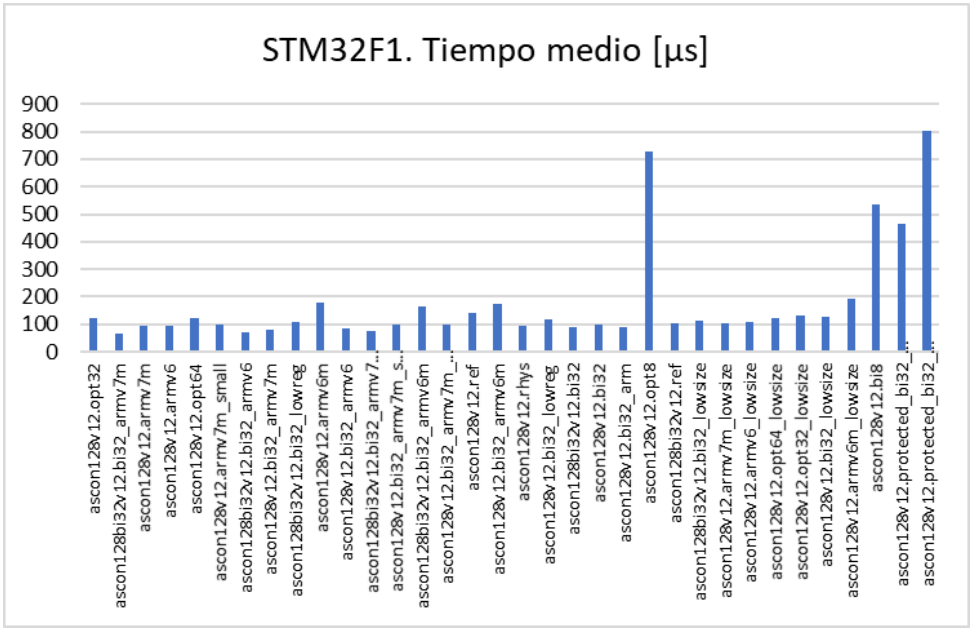

Figura 4: Tiempo medio de ejecución en un STM32F1 para diferentes implementaciones de ASCON128

66. Por su parte, en las Figuras 5 y 6 se muestra el consumo de memoria ROM, en bytes [B], para diferentes implementaciones de ASCON128 en un Arduino UNO y en un STM32F1, respectivamente.

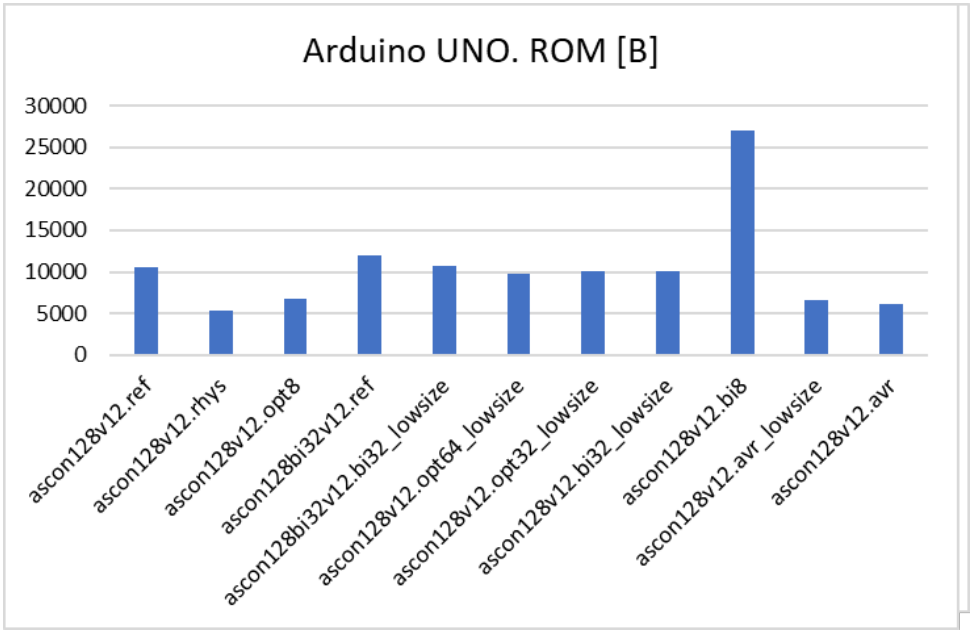

Figura 5: Consumo de ROM en un Arduino UNO para diferentes implementaciones de ASCON128

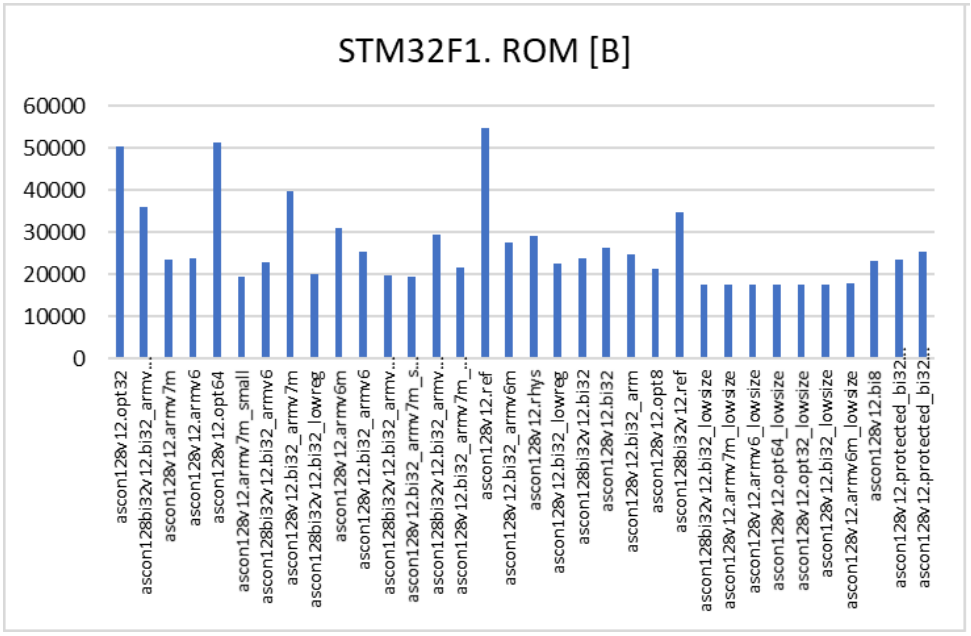

Figura 6: Consumo de ROM en un STM32F1 para diferentes implementaciones de ASCON128

67. En las Figuras 7 y 8 se puede ver el tiempo medio de ejecución, en microsegundos [μs], para diferentes implementaciones de ASCON128a en un Arduino UNO y en un STM32F1, respectivamente.

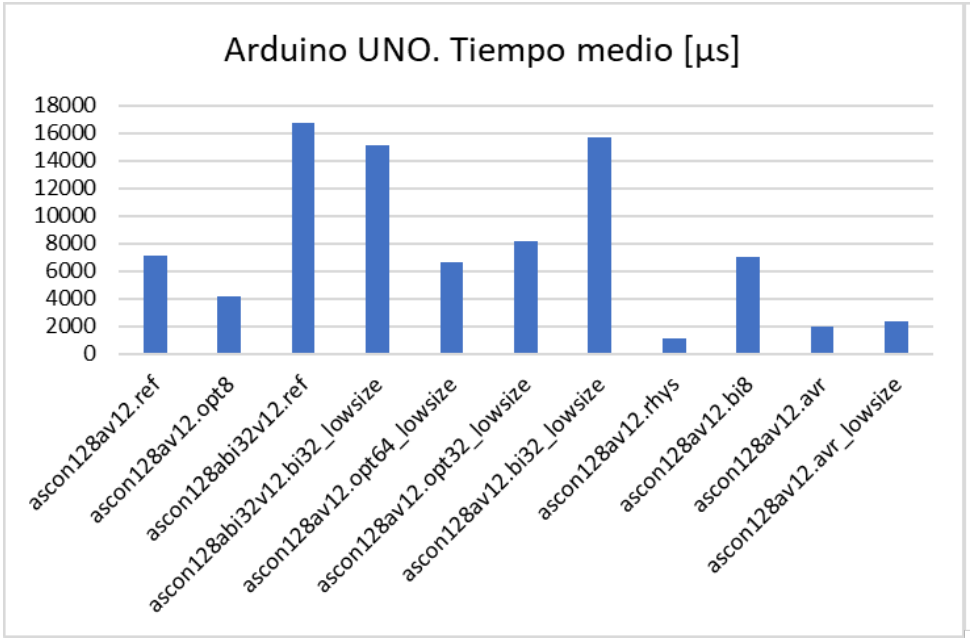

Figura 7: Tiempo medio de ejecución en un Arduino UNO para diferentes implementaciones de ASCON128a

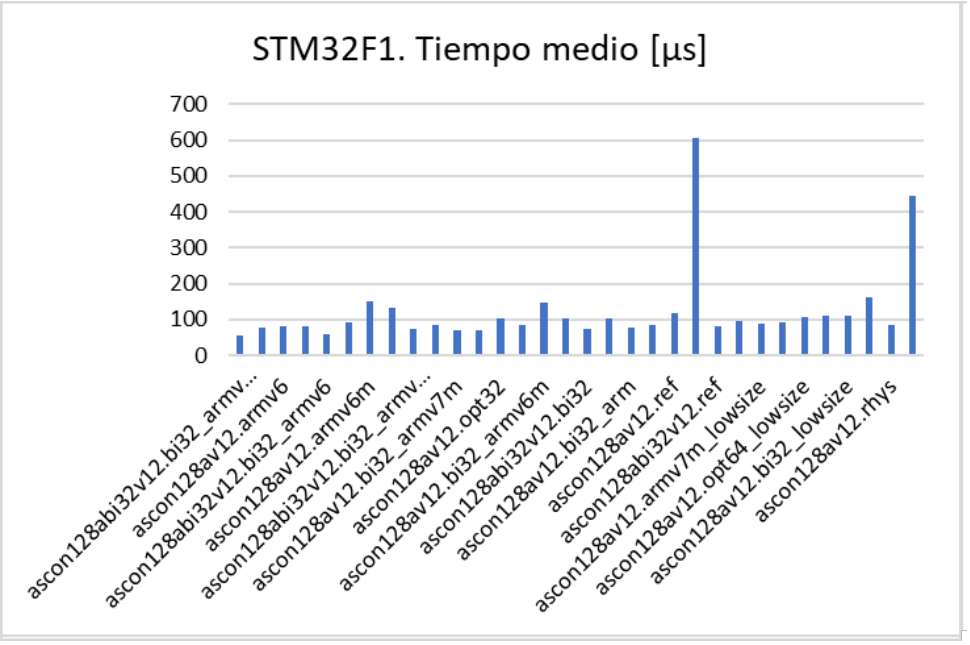

Figura 8: Tiempo medio de ejecución en un STM32F1 para diferentes implementaciones de ASCON128a

68. El consumo de memoria ROM, en bytes [B], para diferentes implementaciones de ASCON128a en un Arduino UNO y en un STM32F1, respectivamente, se presentan en las Figuras 9 y 10.

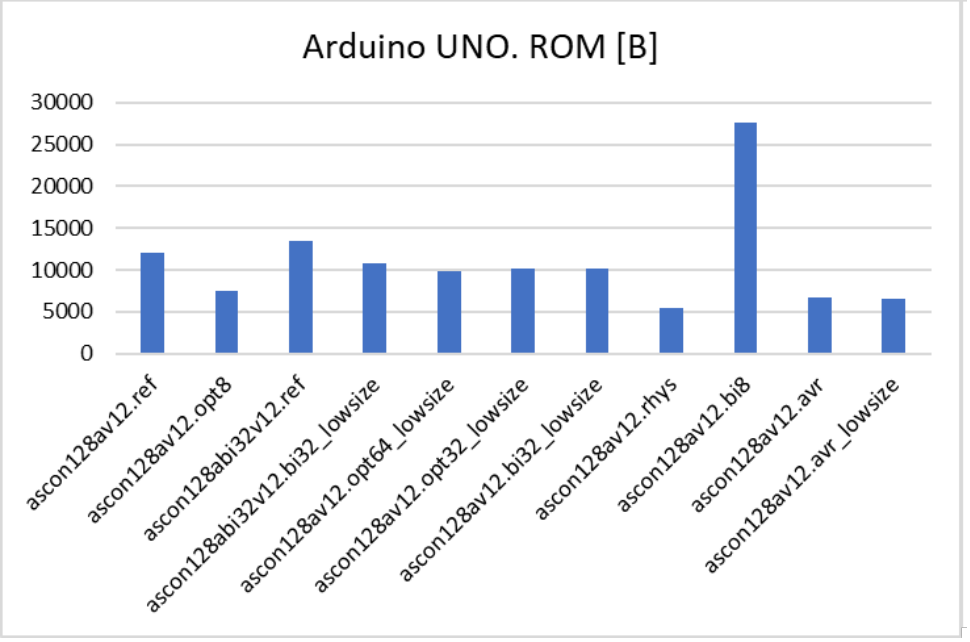

Figura 9: Consumo de ROM en un Arduino UNO para diferentes implementaciones de ASCON128a

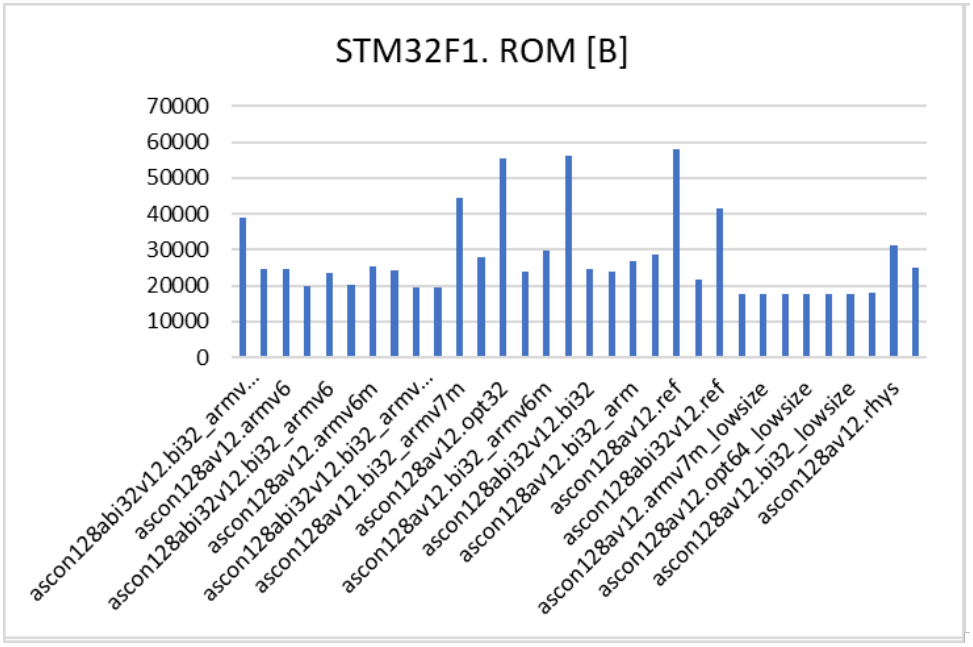

Figura 10: Consumo de ROM en un STM32F1 para diferentes implementaciones de ASCON128a

69. Finalmente, el tiempo medio de ejecución, en microsegundos [μs], para diferentes implementaciones de ASCON80pq en un Arduino UNO y en un STM32F1, respectivamente, se muestra en las Figuras 11 y 12.

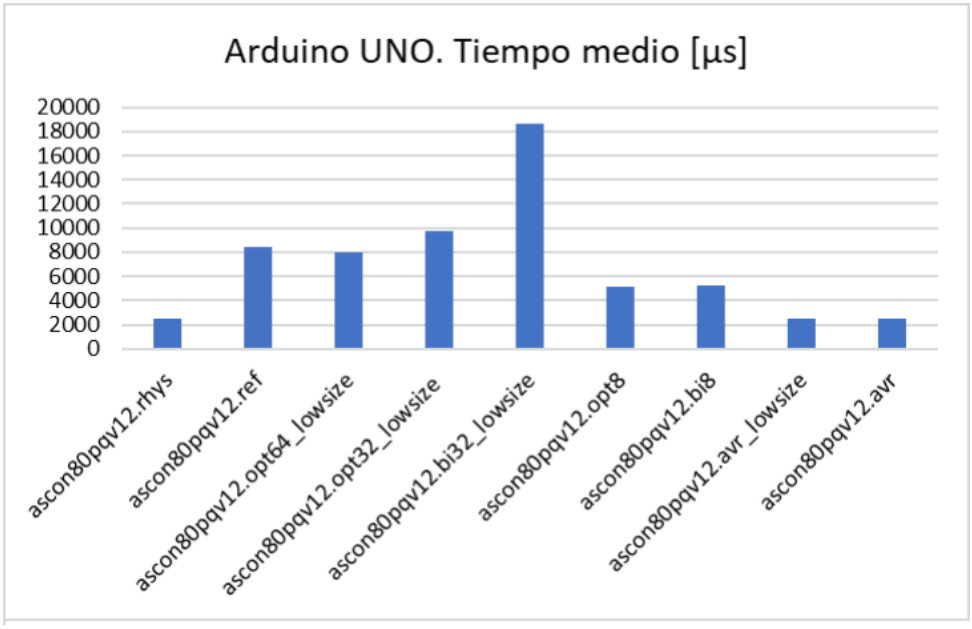

Figura 11: Tiempo medio de ejecución en un Arduino UNO para diferentes implementaciones de ASCON80pq

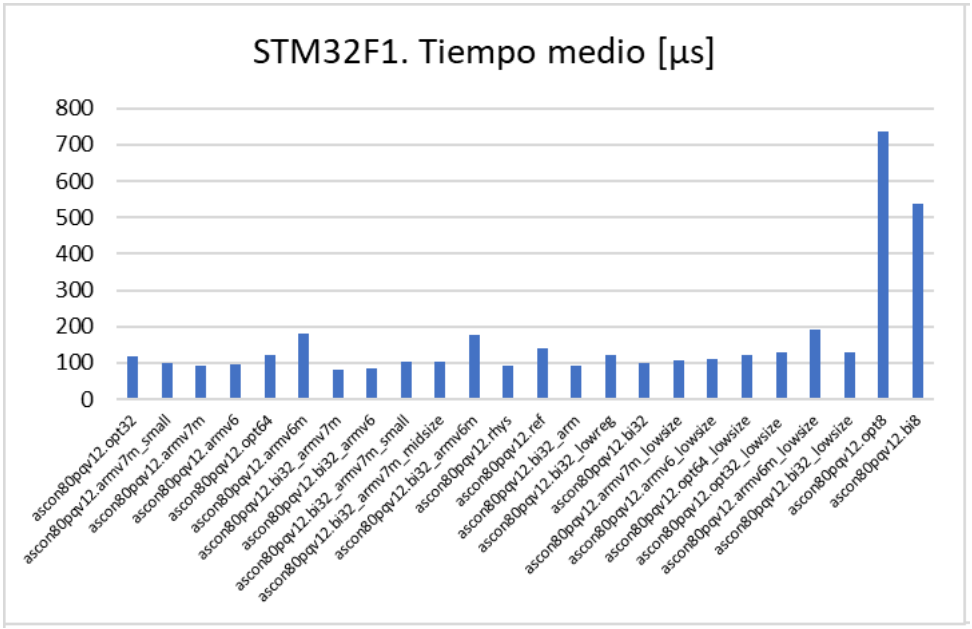

Figura 12: Tiempo medio de ejecución en un STM32F1 para diferentes implementaciones de ASCON80pq

70. Del mismo modo, el consumo de memoria ROM, en bytes [B], para diferentes implementaciones de ASCON80pq en un Arduino UNO y en un STM32F1, respectivamente, se muestra en las Figuras 13 y 14.

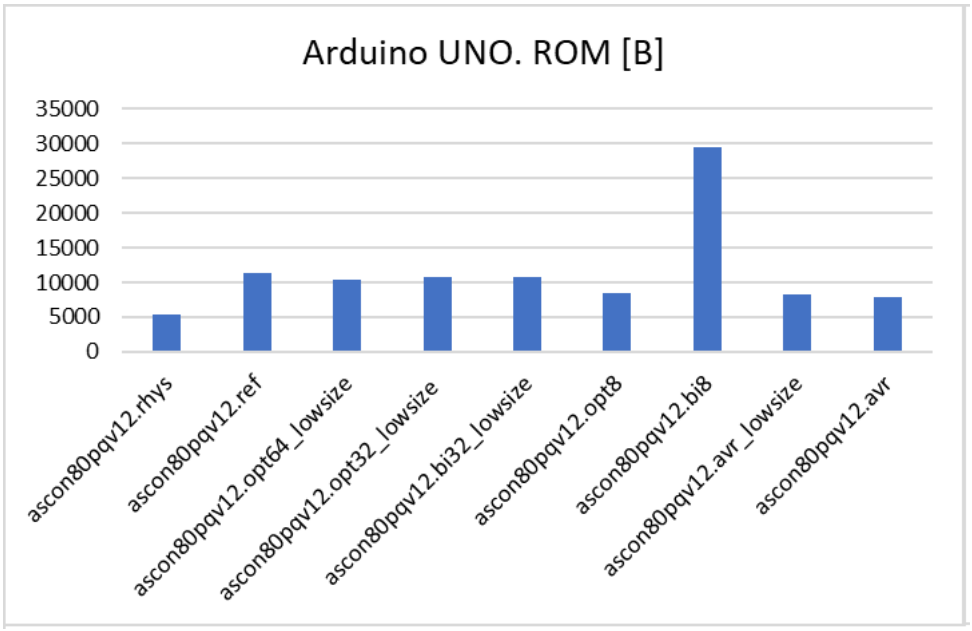

Figura 13: Consumo de ROM en un Arduino UNO para diferentes implementaciones de ASCON80pq

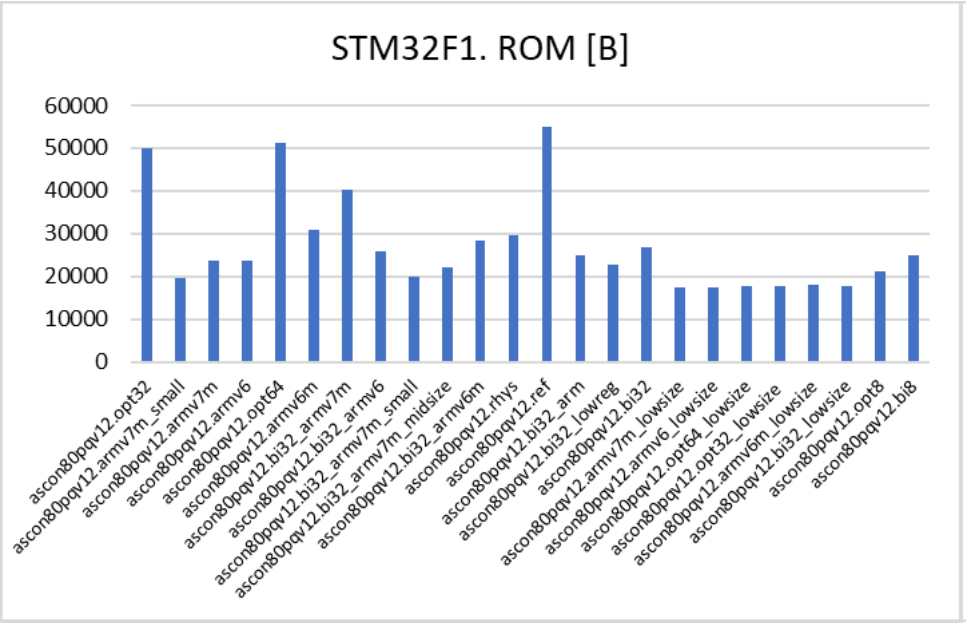

Figura 14: Consumo de ROM en un STM32F1 para diferentes implementaciones de ASCON80pq

B. ANEXO B. COMPARATIVA DE ASCON CON OTROS CIFRADORES

71. En general, para realizar una comparativa entre diferentes algoritmos criptográficos, se suelen considerar determinadas métricas específicas, de modo que reflejen el comportamiento de cada uno de ellos para cada una de las mismas. Estas métricas corresponden a dos parámetros fundamentales, el coste y el rendimiento.

72. La Figura 15 muestra la relación existente entre la seguridad de un sistema de criptografía ligera y las métricas de su coste y rendimiento. Es claro que se trata de lograr la mayor seguridad posible con el menor coste y menor rendimiento, lo que repercute en la propia arquitectura del dispositivo y en parámetros del criptosistema relacionados con el tamaño de la clave y el número de rondas o iteraciones.

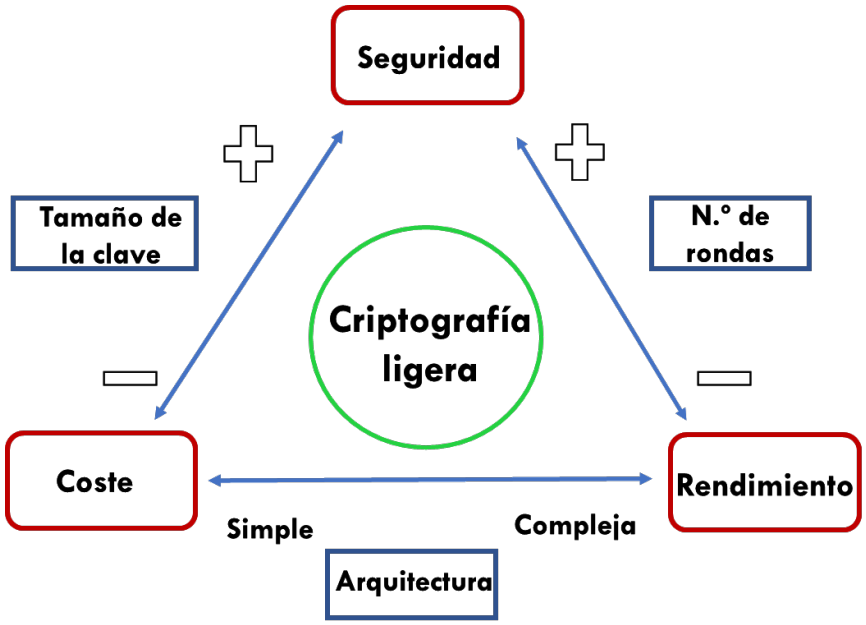

Figura 15: Relación entre seguridad, coste y rendimiento en LwC

73. Estas métricas, para el caso de la criptografía ligera, son las siguientes:

    1. RAM y ROM: es una medida que señala el coste en memoria de las implementaciones en software.
    2. Superficie de puertas (*gate area*): es una medida que indica el coste de las implementaciones en hardware.
    3. Latencia: es una medida de rendimiento que informa del tiempo que requiere el algoritmo para su ejecución desde su inicio hasta que genera la salida.
    4. Rendimiento (*throughput*): mide la cantidad de datos que el algoritmo puede generar en una cantidad de tiempo determinada.
    5. Consumo de energía: es una medida de rendimiento que informa sobre el consumo de energía de las diferentes implementaciones en entornos embebidos.

74. A la hora de llevar a cabo una comparación entre las diferentes métricas, se ha considerado el conjunto de algunos de los sistemas aprobados por el NIST antes de la fase final de la convocatoria LwC y otros sistemas de cifrado ligeros usados antes de la convocatoria del NIST. Para más detalles, recomendamos al lector [19].

75. En las figuras que siguen se han separado en dos grupos los criptosistemas considerados por el NIST en su convocatoria LwC (NIST) de los otros sistemas ligeros, que hemos dentado como Clásicos.

76. Se han incluido los valores correspondientes al estándar AES para apreciar mejor los valores de los sistemas ligeros. Es evidente que no se trata de hacer una comparación de AES con los criptosistemas ligeros en cuanto a tales métricas, pero sí para poder apreciar las diferencias con un patrón estándar bien conocido.

77. La Figura 16 muestra una comparativa con relación a la métrica del coste en RAM+ROM. Como se puede apreciar, ASCON es el algoritmo del NIST que presenta los mejores valores para la métrica relacionada con el consumo de memoria.

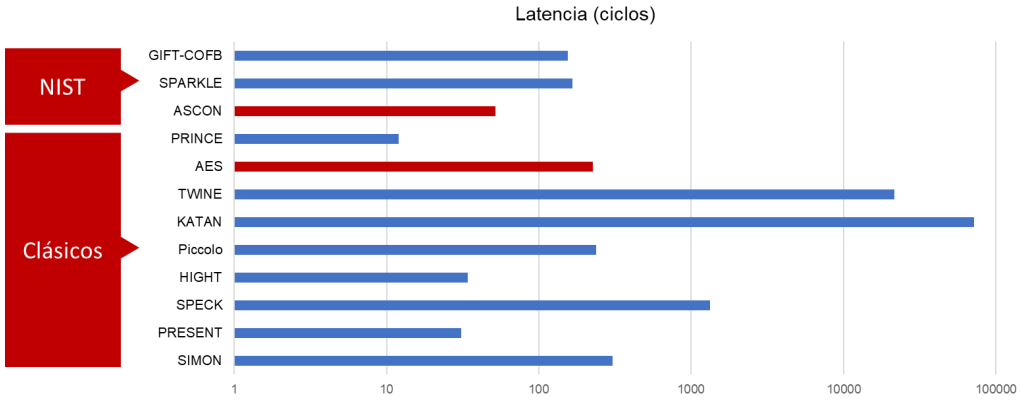

Figura 16: Comparativa en el coste de memoria

78. ASCON presenta, como puede verse en la Figura 17 valores aceptables y en algunos casos más bajos que los algoritmos LwC clásicos y los candidatos del NIST. De los algoritmos clásicos, destaca Piccolo, que es un algoritmo especialmente diseñado para las implementaciones en hardware.

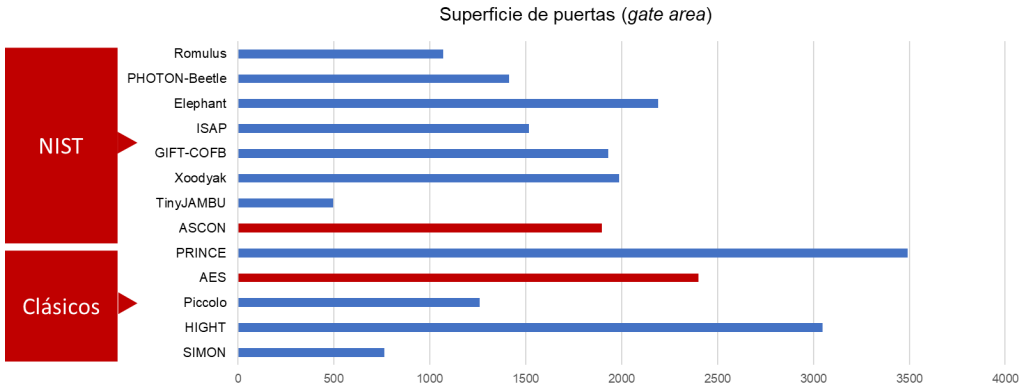

Figura 17: Comparativa en el coste de superficie de puertas

79. En la Figura 18 puede observarse que ASCON es el algoritmo del NIST que presenta los mejores valores en la métrica latencia, esto es, el algoritmo más rápido a la hora de cifrar y descifrar datos. Por su parte, Prince, HIGHT y PRESENT son los únicos algoritmos clásicos que superan la rapidez de ASCON, aunque estos están diseñados especialmente para rendir mejor en hardware.

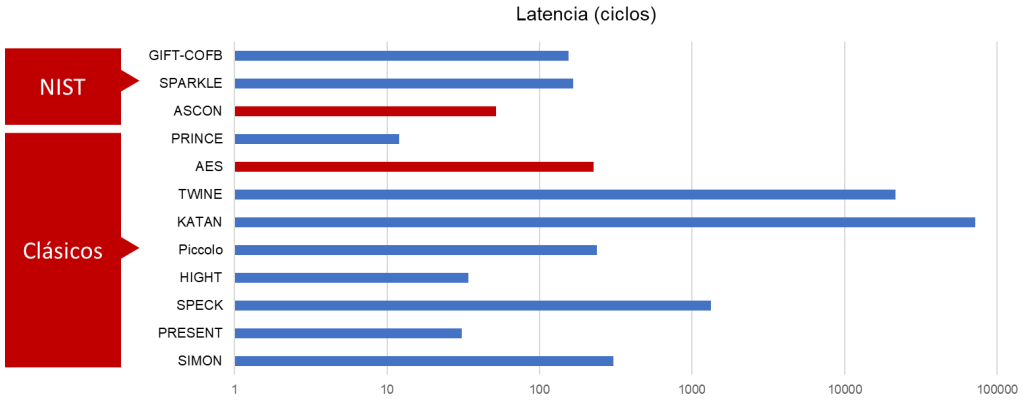

Figura 18: Comparativa en la métrica de latencia

80. Como se observa en la Figura 19, ASCON presenta una significativa mejora en rendimiento respecto el resto de algoritmos presentados, a excepción de Xoodyak, pero que presenta menor seguridad que ASCON.

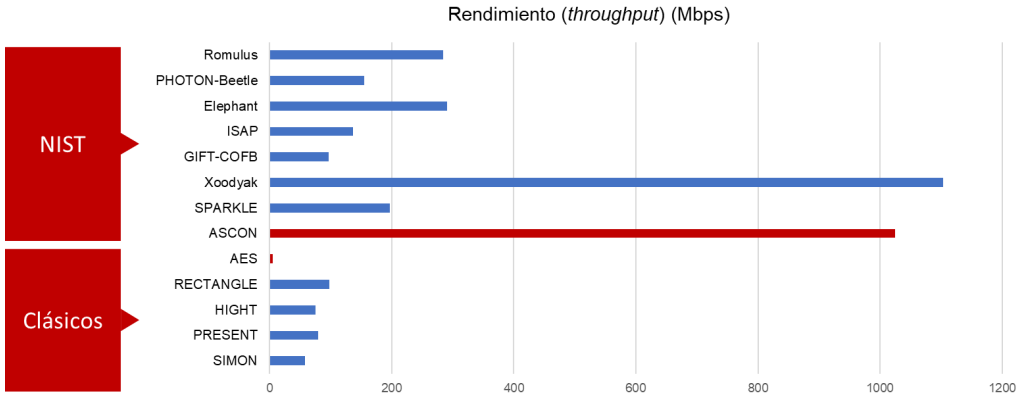

Figura 19: Comparativa en el rendimiento

81. En cuanto a la métrica de consumo de energía, existe una gran diferencia entre los algoritmos clásicos y los presentados al NIST, como era de esperar, dado que los últimos tienen como uno de sus principales objetivos un mínimo consumo de energía. De todos ellos, ASCON es el algoritmo que menos energía consume. Los algoritmos clásicos distan mucho de igualar los números conseguidos por ASCON. Nótese que, como era previsible, AES tiene un consumo mucho mayor que el de ASCON (unas 67.000 veces más nJ).

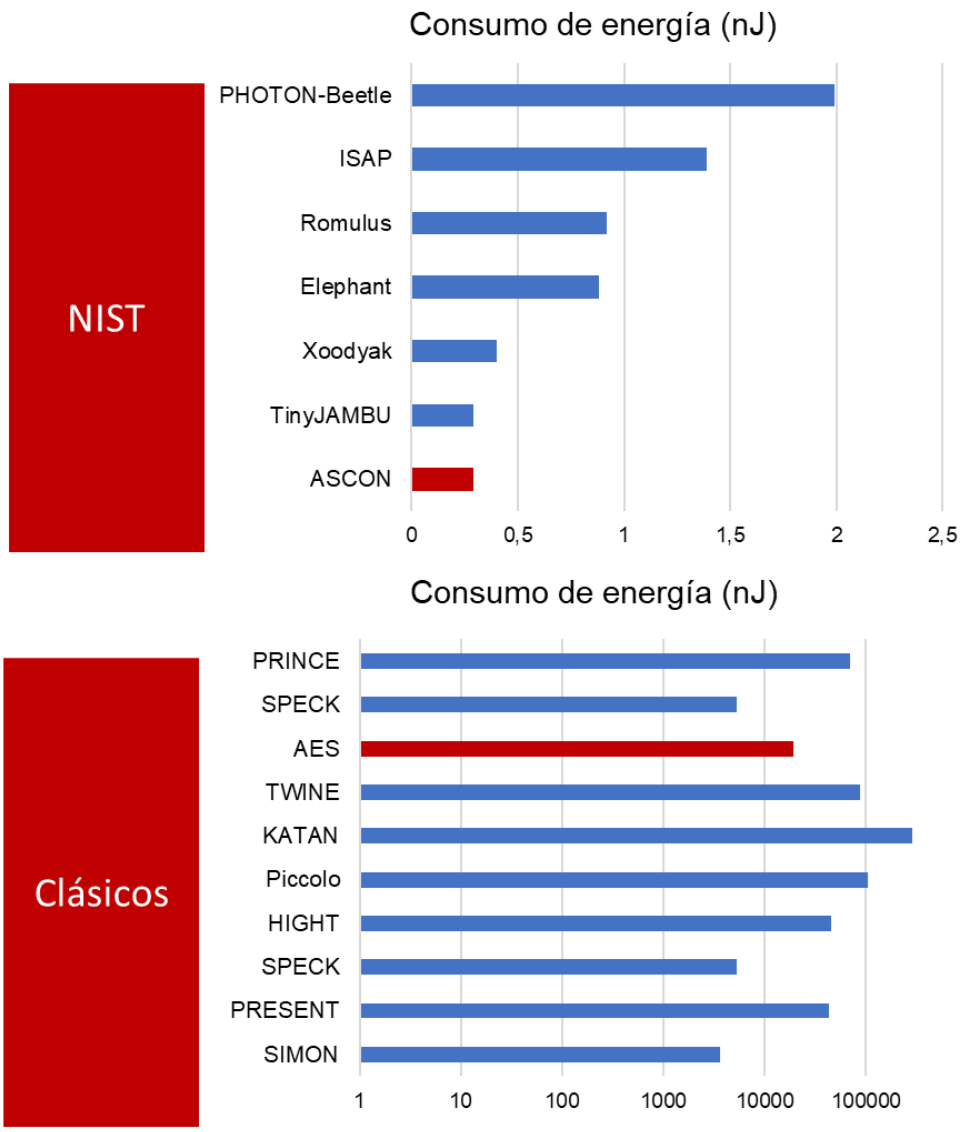

Figura 20: Comparativa en el consumo de energía

CCN-PyTec
centro criptológico nacional

El Departamento de Productos y Tecnologías de Seguridad TIC del Centro Criptológico Nacional (CCN-PyTec) promueve el desarrollo, la evaluación, la certificación y el uso de productos para garantizar la seguridad de los sistemas de tecnologías de la información y la comunicación.

CPSTIC
CATÁLOGO DE PRODUCTOS Y SERVICIOS STIC

CATÁLOGO DE PRODUCTOS Y SERVICIOS DE SEGURIDAD TIC

PDF

ONLINE

ccn-pytec@cni.es

@CCNPYTEC

CCN-PYTEC

www.ccn.cni.es **https://oc.ccn.cni.es** **www.ccn-cert.cni.es**