Published: 30 August 2025
Online AM: 12 June 2025
Accepted: 07 June 2025
Revised: 03 June 2025
Received: 12 April 2023

SURVEY

A Survey on the Implementations, Attacks, and Countermeasures of the NIST Lightweight Cryptography Standard: ASCON

JASMIN KAUR, University of South Florida, Tampa, Tampa, FL, United States

ALVARO CINTAS CANTO, Marymount University, Arlington, VA, United States

MEHRAN MOZAFFARI KERMANI, University of South Florida, Tampa, Tampa, FL, United States

REZA AZARDERAKHSH, Florida Atlantic University, Boca Raton, FL, United States

Open Access Support provided by:

Marymount University

Florida Atlantic University

University of South Florida, Tampa

A Survey on the Implementations, Attacks, and Countermeasures of the NIST Lightweight Cryptography Standard: ASCON

JASMIN KAUR, CSE, University of South Florida, Tampa, United States
ALVARO CINTAS CANTO, EE, Marymount University, Arlington, United States
MEHRAN MOZAFFARI KERMANI, Bellini College of AI, Cybersecurity, and Computing, University of South Florida, Tampa, United States
REZA AZARDERAKHSH, Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, United States

This survey is the first work on the current standard for lightweight cryptography, standardized in 2023. Lightweight cryptography plays a vital role in securing resource-constrained embedded systems such as deeply-embedded systems (implantable and wearable medical devices, smart fabrics, smart homes, and the like), radio frequency identification (RFID) tags, sensor networks, and privacy-constrained usage models. National Institute of Standards and Technology (NIST) initiated a standardization process for lightweight cryptography and after a relatively-long multi-year effort, eventually, in February 2023, the competition ended with ASCON as the winner. ASCON can be viewed as the dual of the widely-deployed AES-GCM block-cipher construction, which, while still state-of-the-art for general-purpose platforms, is resource-intensive for constrained devices, thus it is useful in deeply-embedded architectures to provide security through confidentiality and integrity/authentication. ASCON's lightweight design utilizes a 320-bit permutation which is bit-sliced into five 64-bit register words, providing 128-bit level security. This work summarizes the different implementations of ASCON on field-programmable gate array (FPGA) and ASIC hardware platforms on the basis of area, power, throughput, energy, and efficiency overheads. The presented work also reviews various differential and side-channel analysis attacks (SCAs) performed across variants of ASCON cipher suite in terms of algebraic, cube/cube-like, forgery, fault injection, and power analysis attacks as well as the countermeasures for these attacks. We also provide our insights and visions throughout this survey to provide new future directions in different domains. This survey is the first one in its kind and a step forward toward scrutinizing the advantages and future directions of the NIST lightweight cryptography standard introduced in 2023.

CCS Concepts: • **Hardware** → *Application specific integrated circuits*; *Hardware reliability screening;*

Additional Key Words and Phrases: ASCON, ASIC, differential cryptanalysis, field-programmable gate array (FPGA), lightweight cryptography (LWC), machine-learning (ML) attacks, NIST, side-channel analysis attacks (SCA)

Authors' Contact Information: Jasmin Kaur, CSE, University of South Florida, Tampa, Florida, United States; e-mail: jasmink1@usf.edu; Alvaro Cintas Canto, EE, Marymount University, Arlington, Virginia, United States; e-mail: acintas@marymount.edu; Mehran Mozaffari Kermani, Bellini College of AI, Cybersecurity, and Computing, University of South Florida, Tampa, Florida, United States; e-mail: Mehran2@usf.edu; Reza Azarderakhsh, Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, Florida, United States; e-mail: razarderakhsh@fau.edu.

ACM 0360-0300/2025/09-ART6
https://doi.org/10.1145/3744640

ACM Reference Format:
Jasmin Kaur, Alvaro Cintas Canto, Mehran Mozaffari Kermani, and Reza Azarderakhsh. 2025. A Survey on the Implementations, Attacks, and Countermeasures of the NIST Lightweight Cryptography Standard: ASCON. *ACM Comput. Surv.* 58, 1, Article 6 (September 2025), 16 pages. https://doi.org/10.1145/3744640

1 Introduction

Lightweight cryptography (**LWC**) has become a necessity today as the world is extensively adopting the **Internet of Things** (**IoT**), and the Internet of Nano-Things [1]-[2]. LWC is extensively used in resource constraint devices such as **radio frequency identification** (**RFID**) tags, **wireless sensor networks** (**WSN**), and embedded systems (implantable and wearable medical devices, smart fabrics, smart homes, and the like) to ensure their applications are secure [3]-[4]. However, mathematical security of an algorithm does not imply implementation security, and thus many lightweight cryptographic algorithms, including ASCON, are vulnerable to SCAs [5]-[11]. These studies were conducted before any standardization efforts existed for lightweight cryptography. The **National Institute of Standards and Technology** (**NIST**) initiated a standardization process for lightweight cryptography and after a relatively-long multi-year effort, eventually, in February 2023, the competition ended with ASCON [12] as the winner among the other round three candidates - Elephant [13], GIFT-COFB [14], Grain128-AEAD [15], ISAP [16], Photon-Beetle [17], Romulus [18], Sparkle [19], TinyJambu [20], and Xoodyak [21]. This lightweight cryptographic standard will be used in deeply-embedded architectures to provide security. Previously, ASCON was also chosen as a finalist of the CAESAR competition for authenticated encryption.

ASCON is a lightweight cipher suite that provides **authenticated encryption with associated data** (**AEAD**) as well as hashing functionalities. It uses a duplex-based mode of operation [12]. The 320-bit permutation of ASCON iteratively applies a substitution-permutation network to encrypt/decrypt data in a bit-slice fashion. This bit-slice implementation of ASCON permutation makes it scalable to 8-, 16-, 32-, and 64-bit platforms while remaining lightweight. ASCON has two different variants for different message lengths - ASCON-128 and ASCON-128a. ASCON-128 uses a message length of 64 bits while ASCON-128a uses a message length of 128 bits. ASCON also has a post-quantum secure variant called ASCON-128pq, which is the same as ASCON-128 but uses a 160-bit length key [12]. We note that **post-quantum cryptography** (**PQC**) refers to attacks enabled at the presence of powerful quantum computers. The algorithms for public-key cryptography were standardized in 2022; yet, symmetric-key cryptography is secure against quantum computers since larger key sizes can be used for security against such threats. The current NIST PQC winners are CRYSTALS-KYBER [22], CRYSTALS-DILITHIUM [23], FALCON [24], and SPHINCS+ [25].

Over the years, various cryptanalysis, both differential and side-channel, have been performed on different ASCON variants. Madushan et al. [26] explore the various fault analysis of the NIST LWC standardization process finalists. Furthermore, Dobraunig et al. [27] perform in- depth cryptanalysis of ASCON for key-recovery attacks, forgery attacks, and algebraic attacks by using zero-sum distinguisher. Moreover, they leverage the low algebraic degree of ASCON to construct a zero-sum distinguisher, i.e., a set of input and output values for which sum to zero over $\mathbb{F}_2^n$, for the 12-round ASCON that is able to highlight the ASCON permutation from a random permutation with a complexity of 2^{130} by targeting the internal state after round 5. The recent cryptanalysis of ASCON has strived toward improving the work of [27] as well as to propose new methodology for determining new distinguishers for differential, cube, algebraic, and forgery based key-recovery attacks. This study extends the work of [26], and summarizes the new differential cryptanalysis and SCA works performed on ASCON in hardware/software implementations.

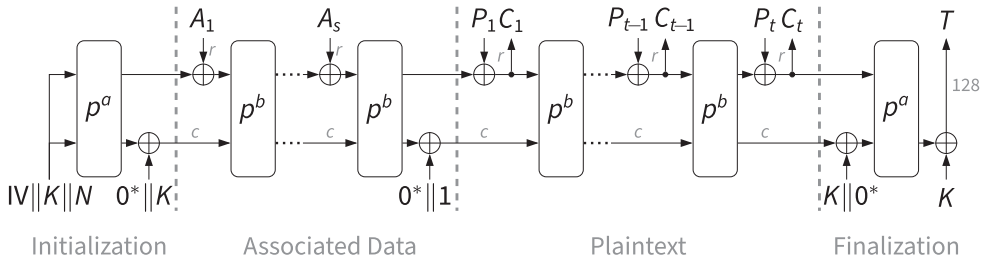

Fig. 1. The associated encryption of ASCON [12].

The SCA works reviewed in this survey also include **statistical fault analysis** (**SFA**), **machine learning** (**ML**), and differential power analysis-based key-recovery attacks performed on the hardware implementations of ASCON. In SFA, the attacker performs a statistical analysis of the injected fault on the output of the ASCON permutation to fully recover the secret key; in ML strategies implement deep and reinforced/unsupervised learning techniques to extract the secret key; while, the **differential power analysis** (**DPA**) exploits the vulnerabilities in the ASCON's initialization operation through statistical analysis of power traces to gain secret information in implementation-specific scenarios. The SCA countermeasures, which are also reviewed in this work, include threshold implementation strategies, stronger S-box design, error-detection mechanisms, as well as protected architectures against side-channel leakage. This article also summarizes the various hardware implementations of ASCON that have improved the design for better area utilization, power consumption, throughput, energy, and efficiency on FPGA and ASIC hardware platforms. We also provide our insights and visions throughout this survey to provide new future directions in different domains. This survey compiles all the recent ASCON implementations and summarizes all the related attacks, and a step forward toward scrutinizing the advantages/future directions of this NIST lightweight cryptography standard introduced in 2023, thus making it a valuable study for expert researchers and practitioners working in this field.

The organization of the article is as follows: Section 2 describes the design and architecture of ASCON. Section 3 explores the hardware implementations of ASCON presented in previous and current literature. Section 4 lists the existent differential cryptanalysis and SCAs performed on ASCON. Finally, we conclude the review in Section 5.

2 Preliminaries

The entire design specification of ASCON is given in [12]. ASCON's encryption process (Figure 1) is designed as a sponge-based MonkeyDuplex construction which consists of 4 stages, namely, initialization operation, processing of the associated data, processing of the plaintext, and the finalization operation. These four stages get updated using two 320-bit permutations, i.e., p^a and p^b, where a and b are the number of rounds. Both of the permutations are bit-sliced into five 64-bit register words which make up the 5-bit internal state. In a full 12-round ASCON, the permutations iteratively apply a **substitution-permutation network** (**SPN**)-based round transformation which consists of adding round constants, applying the substitution layer, and employing the linear layer for diffusion to the internal state.

The substitution layer consists of a non-linear 5-bit S-box (Figure 2) whose hexadecimal form is shown in Table 1. This S-box is applied 64 times in parallel to update each bit-slice of the internal state. The 5-bit S-box is designed using Boolean logic which makes it highly compact and lightweight for implementations on both ASIC and FPGA hardware platforms. The linear layer of ASCON updates each 64-bit word of the internal state by first rotating register words with

Table 1. LUT Representation of the Non-linear 5-bit S-box *SB* of ASCON-128 in Hexadecimal form for Input Vector μ

μ	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$SB(\mu)$	**0**4	**0**b	1f	14	1a	15	**0**9	**0**2	1b	**0**5	**0**8	12	1d	**0**3	**0**6	1c
μ	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
$SB(\mu)$	1e	13	**0**7	**0**e	**0**0	**0**d	11	18	10	**0**c	**0**1	19	16	**0**a	**0**f	17

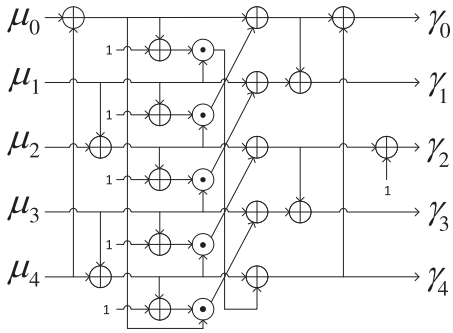

Fig. 2. The 5-bit S-box of ASCON [12, 45].

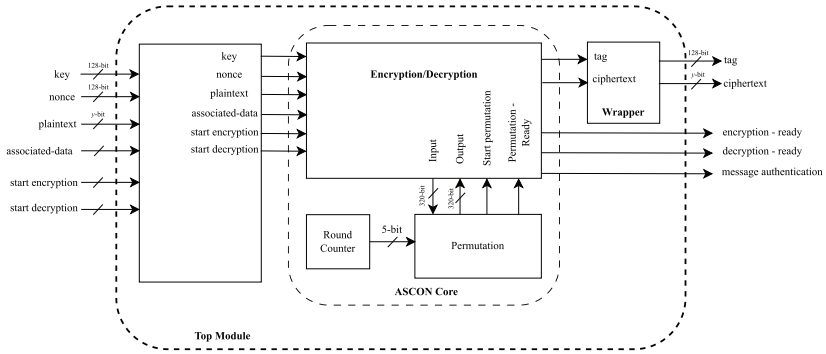

Fig. 3. ASCON architecture [12, 28].

different shift values, and then performing a modulo-2 addition on the shifted word values. A top level view of ASCON's architecture in hardware implementations is presented in Figure 3.

In the ASCON architecture (Figures 1—3), most hardware optimizations are focused on the 320-bit permutation function of ASCON, specifically the 5-bit S-box. Section 3 discusses various emerging studies focused on optimizing ASCON through look-up tables, combinational logic, rolled/unrolled, recursive, and bit-sliced approaches in hardware implementations. In addition to architectural modifications, protected/unprotected ASCON architectures are also discussed in the next section - where the ASCON implementations are protected through secure designs of the 5-bit S-Box. The ASCON's permutation function, providing AEAD functionality, is also often the main focus of various cryptanalysis attacks. In addition to exploiting mathematical vulnerabilities of ASCON, reducing the rounds to mount successful implementation-attacks such as differential power analysis attacks are other strategies that attackers adopt. These mathematical and and implementation specific attacks are discussed in Section 4.

Table 2. Overhead Results of Different Hardware Implementations of ASCON on FPGA Hardware Platforms

ASCON Architecture	FPGA Hardware Platform	Area (LUTs)	Power (mW) textbf@Freq.	Throughput (Mbps)	Efficiency (Mbps/LUT)
ASCON (original) [45]	Spartan 7	371 (slices)	99@100MHz	6.646	17.914
ASCON (original) [45]	Kintex 7	376 (slices)	88@100MHz	6.709	17.843
ASCON (unprotected) [50]	Spartan 6	2,048	11.5@100MHz	255.4	0.1247
ASCON [32]	Artix7	1,808	26.8@232MHz	39.0	-
ASCON-128 [34]	Artix7	1,330	31	457	0.343
RECO-HCON (128) [36]	Artix7	1,548	-	5, 926	-
RECO-HCON (128a) [36]	Artix7	1,548	-	9, 077	-
RECO-HCON (hash) [36]	Artix7	1,548	-	3, 160	-
RECO-HCON (hash-a) [36]	Artix7	1,548	-	4, 534	-
ASCON (protected) [50]	Spartan 6	6,364	37.5	134.6	0.0212
ASCON-128 (Logic One-bit) [45]	Spartan 7	373 (slices)	99@100MHz	6,705	17.783
ASCON-128 (Logic Interl.-bit) [45]	Spartan 7	380 (slices)	99@100MHz	6,687	17.444
ASCON-128 (Logic CRC-3) [45]	Spartan 7	407 (slices)	99@100MHz	6,603	15.601
ASCON-128 (LUT One-bit) [45]	Spartan 7	372 (slices)	100@100MHz	6,448	17.333
ASCON-128 (LUT Interl.-bit) [45]	Spartan 7	377 (slices)	100@100MHz	6,443	17.090
ASCON-128 (LUT CRC-3) [45]	Spartan 7	425 (slices)	100@100MHz	6,431	15.131
ASCON-128 (Logic One-bit) [45]	Kintex 7	381 (slices)	89@100MHz	6,705	17.598
ASCON-128 (Logic Interl.-bit) [45]	Kintex 7	384 (slices)	89@100MHz	6,687	17.414
ASCON-128 (Logic CRC-3) [45]	Kintex 7	385 (slices)	89@100MHz	6,603	17.150
ASCON-128 (LUT One-bit) [45]	Kintex 7	363 (slices)	89@100MHz	6,776	18.667
ASCON-128 (LUT Interl.-bit) [45]	Kintex 7	372 (slices)	89@100MHz	6,409	17.228
ASCON-128 (LUT CRC-3) [45]	Kintex 7	384 (slices)	89@100MHz	6,383	16.224
ASCON-128 (unrolled) [37]	Virtex4	26,943	-	817.41	0.031
ASCON-128 (recursive) [37]	Virtex4	4,021	-	506.29	0.125
ASCON-128 (unrolled) [37]	Virtex7	22,636	-	1,342.31	0.059
ASCON-128 (recursive) [37]	Virtex7	2,708	-	721.53	0.266
ASCON-128 (unrolled) [37]	Spartan6	22,636	-	688.83	0.031
ASCON-128 (recursive) [37]	Spartan6	2,781	-	346.50	0.124
ASCON-128a (unrolled) [37]	Virtex4	30,006	-	1,496.25	0.049
ASCON-128a (recursive) [37]	Virtex4	4,215	-	970.25	0.231
ASCON-128a (unrolled) [37]	Virtex7	25,187	-	2,419.88	0.096
ASCON-128a (recursive) [37]	Virtex7	2,916	-	1,357.08	0.465
ASCON-128a (unrolled) [37]	Spartan6	25,187	-	1,247.22	0.049
ASCON-128a (recursive) [37]	Spartan6	2,918	-	638.44	0.218
Fault-injected ASCON [40]	SASEBO-GII	217	7.8	198	-
Key-bypass HT on ASCON [60]	SoC Cyclone V	827	22.4	-	-
Round-red. HT on ASCON [60]	SoC Cyclone V	771	22.3	-	-
ASCON (128+128a Iterative) [31]	Kintex-7	1,497	236@331MHz	400 (128); 914 (128a)	0.267 (128); 0.611 (128a)
ASCON (128+128a Iterative) [31]	Spartan-7	1,804	219@309MHz	355 (128); 853 (128a)	0.197 (128); 0.473 (128a)
ASCON (128+128a Iterative) [31]	Artix-7	1,756	222@317MHz	376 (128); 853 (128a)	0.214 (128); 0.486 (128a)
ASCON (128+128a Iterative) [31]	Virtex-7	1,632	239@335MHz	400 (128); 914 (128a)	0.245 (128); 0.560 (128a)

3 Hardware Implementation of ASCON

This section briefly goes over the various hardware implementations of the ASCON family along with any optimizations that have been proposed in recent years. All the overhead results are presented in terms of area, power, delay, throughput for FPGA implementations, and energy utilization for ASIC implementations are tabulated in Table 2 and Table 3, respectively. The performance

Table 3. Overhead Results of Different Hardware Implementations of ASCON on ASIC Hardware Platform

ASCON Architecture	ASIC Technology	Area (kGE)	Power (mW)	Throughput (Mbps)	Energy
ASCON-fast-6-rounds [29]	90 nm	25.80	0.184	13,218	23 $\mu J/byte$
ASCON-64-bit [29]	90 nm	5.86	0.032	72	1,397 $\mu J/byte$
ASCON-x-low-area [29]	90 nm	3.75	0.015	14	5,706 $\mu J/byte$
ASCON-fast-TI-6-rounds [29]	90 nm	125.19	0.830	9,028	104 $\mu J/byte$
ASCON-x-low-TI [29]	90 nm	9.19	0.045	15	17,234 $\mu J/byte$
RECO-HCON-128 [36]	28/32 nm	25.1	1.990	5, 926	0.335 pJ/bit
RECO-HCON-128a [36]	28/32 nm	25.1	1.990	9, 077	0.219 pJ/bit
RECO-HCON-hash [36]	28/32 nm	25.1	1.990	3, 160	0.637 pJ/bit
RECO-HCON-hash-a [36]	28/32 nm	25.1	1.990	4, 534	0.439 pJ/bit
ASCON CMOS-64-bit [38]	28 nm	10.1529	0.7459	-	478.9 pJ
ASCON CMOS-256-bit [38]	28 nm	10.1529	0.7459	-	736.2 pJ
ASCON CMOS-512-bit [38]	28 nm	10.1529	0.7459	-	1079.4 pJ
ASCON CMOS/STT-MRAM-64-bit [38]	28 nm	10.7155	0.7148	-	427.9 pJ
ASCON CMOS/STT-MRAM-256-bit [38]	28 nm	10.7155	0.7148	-	556.5 pJ
ASCON CMOS/STT-MRAM-512-bit [38]	28 nm	10.7155	0.7148	-	728.1 pJ
ASCON-loop-folded [33]	32 nm	36.7	1.01	1,058	-
ASCON-loop-unrolled-2 [33]	32 nm	45.8	1.14	1,228	-
ASCON-loop-unrolled-3 [33]	32 nm	53.8	1.32	1,288	-
ASCON-loop-unrolled-4 [33]	32 nm	76.2	1.87	1,206	-
ASCON-loop-fully-unrolled [33]	32 nm	277.1	6.8	2,178	-

metrics are chosen to encapsulate resource tradeoffs between area, power, and delay for higher throughput and efficiency for implementations in resource-constrained devices.

3.1 Architectural Optimizations for Hardware Implementations

Various hardware designs of ASCON are implemented in [29] for applications such as RFID tags, WSNs, and embedded systems. Such hardware implementations of ASCON that [29] proposes are: ASCON-fast, ASCON-64-bit, and ASCON-x-low-area.

ASCON-fast [29] is a high throughput design with minimal processing delay, which uses unrolled round transformations. At least one round transformation is performed each clock cycle without any pipelining. This allows multiple rounds to complete in a single clock cycle and each ASCON-fast variant uses a different number of the unrolled round transformations. The unrolled round transformation is connected with the data bus and key registers using a few additional multiplexers and XOR gates.

ASCON-64-bit [29] uses an **arithmetic logic unit** (**ALU**) where the control path executes similarly to a sequential code. The design uses two temporary registers in addition to the five state registers which along the inputs from the control path constitute the inputs to the ALU. The ALU takes the 64-bit data input and arranges them in either the high or low part of the selected operand using a barrel-shift unit, a data storage unit, and a three logic operations. The result of the operation is selected at the output of the ALU which is then applied to the destination register. The S-box and the linear layer are iteratively calculated using the ALU operations during the execution phase, thus making one round operation 59 clock cycles. The design of the S-box is altered to use twenty-five 3-operand instructions and two temporary registers to decrease the area.

In the ASCON-x-low-area variant proposed in [29], the datapath is designed to use a radical low-area "one-bit operation per cycle" approach. The five state registers are clock-gated shift registers with independent shift-enable inputs. All the state registers are active during the S-box calculation and the data is shifted bit-slice-wise in each S-box instance in 64 clock cycles. The linear diffusion layer updates each state register individually in five interleaved sub-iterations. A temporary shift

register is used to store the results of the current linear layer in one iteration and which are then written back in the next iteration. This low-area design uses 512 clock cycles per round transformation. All the overhead results for the aforementioned implementations are tabulated in Table 3.

In [30], a flexible, reconfigurable, and energy-efficient crypto-processor to run ASCON is introduced by Wei et al. The proposed ASCON crypto-processor runs in six different modes: Encryption, decryption, and hash function with different data sizes. The crypto-processor consists of an ASCON core, shift registers (FIFO), and an I/O interface. First, the data and text inputs are loaded into 128-bit shift registers as FIFO, while the key, the nonce, and the target instance mode are processed on the *Start* signal. The ASCON core stays occupied until the entire ciphertext reaches the same size as the input message in AEAD mode, reaches 256 bits when in hashing mode, or the tag verification result is given out once done. The four shift registers provide flexibility in adapting the ASCON processor to various IoT systems with variable block sizes as they are used to divide and pad the inputs to match the block size of the used default variant ASCON-128a. The input sizes for other ASCON variants ASCON-128 and ASCON-hash are processed either by splitting the 128-bit inputs as two 64-bit inputs or by adapting a different counting technique to fully utilize the space, respectively.

The ASCON-core [30] consists of two stages: Selective XOR and parameterized permutation. It runs iteratively to keep the permutation block (Section 2) busy once started. For every fixed number of rounds, the permutation block reads a new message. Between two iterations of the ASCON permutation, the input for the next iteration is computed by XORing the current sponge state with either another message, key, or a constant. The order of iterations is controlled by an FSM controller, which tracks the progress based on the status of the state registers to select the correct round number and the XOR operand corresponding to initialization, run, and finalization operations. The challenge of supporting multiple instances with an arbitrary round number, variable XOR operand, and a block size is overcome by splitting the current sponge state into two parts: A head comprising 128 **most significant bits** (**MSB**), and a tail comprising 192 **least significant bits** (**LSB**). These two parts get updated in parallel by combining the similarity in XOR operands and block size. Muxes are used to select the correct XOR operand based on the current finite state. Since various components in the proposed ASCON core are reused, the functionality to support multiple instances does not incur significant overhead when compared to a single ASCON-128a core instance. The different area, frequency, power, throughput, and efficiency overheads of the proposed architecture on FPGA and ASIC hardware platforms are shown in Table 2 and Table 3, respectively.

A. R. Alharbi et al. [31] propose a unified/scalable design of ASCON variants 128 and 128a, implemented in hardware on 7-Series FPGA devices (Kintex, Virtex, Spartan, and Artix). The proposed architecture is designed to perform the authenticated encryption with associated data iteratively using one permutation per clock-cycle which is controlled using a **finite state machine** (**FSM**)-based controller. To optimize memory, the proposed design uses input/output buffers instead of traditional memory to reduce the critical path delay of the design, and, in turn, cause reducing the operating frequency. The proposed design also focuses on iteratively implementing the permutation function by using buffers in the datapath to improve the design frequency. To combine the designs, the proposed design uses 128-bit buffer for the IV to handle different data blocks of ASCON-128 (64-bit data block) and ASCON-128a (128-bit data block). The preferred ASCON implementation is selected using a 1-bit "mode" signal with the controller. The proposed design is benchmarked on FPGA platforms, results in the tabulated figures in Table 3. The proposed ASCON design utilized 1,632, 1,497, 1,904, and 1,756 LUTs with power consumption/frequency of 239mW@335MHz, 236mW@331MHz, 219mW@309MHz, and 222mW@317MHz on Virtex-7, Kintex-7, Spartan-7, and Artix-7 FPGA devices, respectively.

Khan et al. [32] explore the hardware performance of ASCON for **artificial intelligence** (**AI**) enabled IoT devices. Unrolled and recursive strategies have been adopted for ASCON implementations on Virtex-4, Virtex-7, and Spartan-6 FPGA families. The unrolled scheme has been designed to achieve high throughput while the recursive scheme helps in reducing hardware costs. For the unrolled architecture, the encryption/decryption is performed using combinational circuits and ASCON permutation is deployed in an unrolled manner for initialization, run, and finalization phases. This results in higher throughput at the cost of high area overhead. Moreover, since the permutation function utilizes the same hardware for every stage, a recursive strategy is implemented by the authors to achieve high throughput. To successfully implement hardware re-utilization in ASCON permutation functions, the authors propose computing two permutation rounds per clock cycle, thus requiring a total of 24 clock cycles for encryption/decryption using ASCON-128 (as opposed to 36 clock cycles for 36 permutations) and 26 clock cycles for the ASCON-128a variant (as opposed to 40 clock cycles for 40 permutations). The area overhead is negligible for any additional XOR operations required. The overhead results of both unrolled and recursive implementations have been tabulated in Table 2.

Khan et al. [33] study the implementation of loop folded, loop unrolled, and fully unrolled accelerator architectures for ASCON on SAED 32 nm technology. In the study, the different design explorations highlight the tradeoff between area and throughput (performance) for resource-constrained devices, accommodating both ASCON-128 and ASCON-128a variants. The results are evaluated in terms of hardware utilization, power consumption, and execution time. From the results, the fully-unrolled accelerator is the best in implementations where high throughput is required such as applications in CCTV cameras for encrypted video surveillance. The other two architectures, loop folded/unrolled, are suitable for different implementations in IoT ecosystems, where high data processing is not required though efficiency is necessary - such as smart parking, smart traffic solutions, and energy consumption scenarios. The results for 32mn ASIC technology implementations are presented in Table 3.

3.2 Designs Focused on Security Robustness

An FPGA-based application of ASCON cipher in portable **Internet of Medical Things** (**IoMT**) devices is presented in [34], where the cipher is used to enhance the security of such devices using AEAD functionality. A round-based architecture of ASCON is designed for round calculation per clock cycle. The proposed design utilizes the dual output LUT (LUT6) feature of the Xilinx 7-series FPGA boards to implement the 5-bit S-box of ASCON for the optimized area. Implementing the 5-bit S-box of ASCON using LUT6 utilized only three LUTs in FPGA implementation, significantly optimizing the area when compared to other hardware implementations of ASCON. The hardware implementation is performed on the Xilinx Artix-7 FPGA family and the area (in terms of LUTs), throughput, frequency, and efficiency (throughput/area) results are shown in Table 2. This proposed implementation of ASCON cipher consumed 35% less area and 56% more efficiency when compared to the architecture of ASCON in [35].

Diehl et al. [36] compare the protected and unprotected implementation of ASCON against first-order DPA using **test vector leakage assessment** (**TVLA**) implemented using Flexible Open-source workBench **fOr Side-channel analysis** (**FOBOS**). The overhead results of the protected ASCON implementations are presented in Table 2.

FOBOS 2, an upgraded and optimized FOBOS, is proposed in [37] which is used to evaluate power measurements and SCA resistance for the hardware implementations of various lightweight ciphers with AEAD functionality on the Xilinx Artix-7 FPGA board. The results of power consumption, frequency, throughput, and energy/bit obtained using FOBOS 2 are tabulated in Table 2. The results show that ASCON performed better in terms of having the lowest power consumption (33.5

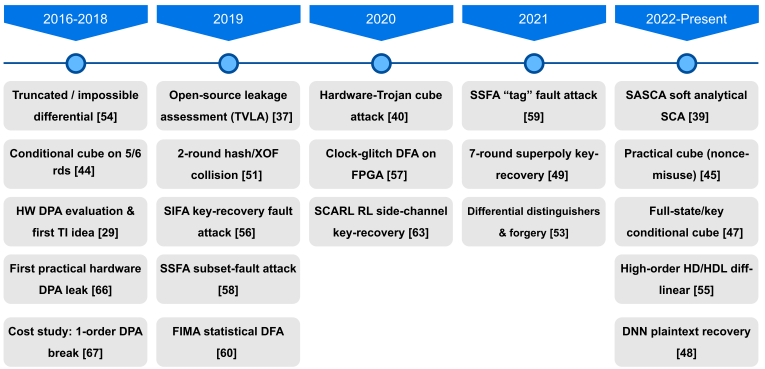

Fig. 4. A timeline of attacks mounted on ASCON in recent years.

mW at 50 MHz), and lowest incrementally increasing dynamic power with increasing frequency among the NIST standardization process candidates Spoc, Spook, and GIFT-COFB. The energy per bit of ASCON was 0.86 nJ/bit while the static power consumption was around 27 mW.

A CMOS/STT-MRAM-based hardware implementation of ASCON benchmarked on an ASIC hardware platform is proposed by Roussel et al. [38]. Such implementation is made resilient to power failure by replacing volatile CMOS flip-flops with non-volatile flip-flops to save the intermediate state of ASCON computations which can then be retrieved on startup. This hybrid CMOS/STT-MRAM implementation helps in reducing energy utilization between 11% to 48%, while incurring an area overhead of about 5.5% when compared to CMOS-only implementation of ASCON. These results are presented in Table 3.

4 Differential and Side-channel Cryptanalysis of ASCON

Various structural and mathematical vulnerabilities of cryptographic ciphers with authenticated encryption such as ASCON can be exploited to gather information from the permutation during encryption/decryption processes. Cryptanalysis, differential or side-channel, can be performed on this gathered information to recover the secret key and break the cipher. This section summarizes the works presenting cryptanalysis of ASCON in terms of algebraic attacks, cube/cube-like attacks, differential attacks, fault attacks, and power attacks. Most of these attacks target the reduced round versions of ASCON to recover the secret key successfully. Machine learning techniques have also been adopted to find differential distinguishers of the ASCON permutation from a random one to gather more information about its internal state and eventually recovering the secret key in an acceptable time. A timeline of the attacks mounted on ASCON is presented in Figure 4.

For the sake of brevity, discussions on algebraic and differential cryptanalysis have been moved to the online supplementary materials. Readers are encouraged to refer to them via the ACM Digital Library.

4.1 Fault Analysis

Ramezanpour et al. [39] successfully apply a **statistical ineffective fault analysis** (**SIFA**) using double-fault injection and key-dividing techniques on the S-box of ASCON in a software implementation to recover the secret key. The authors inject faults in any selected pair of S-boxes for every encryption performed in the last round of the finalization stage of ASCON. The faults are injected using a clock and/or voltage glitches are injected in a manner where they do not affect the result of the S-box. Then, the correct tag values resulting from induced ineffective faults are analyzed to gather information about the secret key. The probability of distribution is assumed to

be biased in the proposed attack and the attack is successful as long as there is sufficient data available for analysis (12.5 to 2500 correct tag values in the proposed study). Thus, the SIFA-based fault model requirements are less than other differential fault analysis techniques and are also noise tolerant. Fault attack countermeasures such as error detection and error-randomization techniques fail in the presence of SIFA as they rely on the incorrect output value in cipher operations under fault inductions. The best countermeasures against SIFA are those where the fault injection mechanisms can be detected or where the fault distribution is independent of secret data. Sensor-based techniques can detect fault injection mechanisms, however, they are limited in the fault mechanisms they can detect. The FPGA and ASIC hardware implementations are proposed as future work.

Surya et al. [40] implement a synchronous clock glitching strategy to induce delay faults in selected parts of the ASCON architecture. SASEBO-GII FPGA board is used for the hardware implementation and a **Digital Clock Manager** (**DCM**) is used to generate synchronous clock signals. In every encryption round, the faults are injected into the ASCON S-box via a high-frequency faulty clock signal resulting in faulty output in the ASCON linear layer. Surya et al. also try to implement an asynchronous clock glitching strategy, however, it incurs higher utilization overhead when compared to the synchronous method [40]. The fault injection is performed by feeding the faulty clock signal only to a few parts of the design to better observe the error distribution and propagation, and to emulate EM injections easily. Possible countermeasures to this kind of attack include a **threshold implementation** (**TI**) scheme and a unified masking approach to protecting the ASCON architecture. Hardware implementation overheads are given in Table 2.

Joshi and Mazumdar [41] perform a **subset fault analysis** (**SSFA**) on a software implementation of ASCON-128 by attacking the vulnerabilities in its S-box. The strategy tries to find correlations between the input and output bits of the ASCON S-box by determining which output bits become 0 for input bits set to 0. They also propose a key division strategy to decrease the search space for key recovery to 2^{64} for the worst case. Key masking before key whitening operation, error detection via partial decryption, or using a new and strengthened S-box design resilient to 1-bit SSFA are proposed as the countermeasures against SSFA. In [41], a fault attack called the preliminary attack which focuses on vulnerabilities in the key whitening function and the tag creation function of the finalization stage is also proposed. Both attacks are shown to retrieve the entire secret key using a key analysis methodology. For the preliminary attack strategy, the involution property of the XOR function is leveraged to recover the key value from the generated tag. The attack can be mounted on ASCON in three ways: (1) Injecting faults into three selected S-boxes which requires 374 fault injections to recover the full secret key; (2) Injecting faults into a single selected S-box followed by an instruction skip error (induced by another fault injection) which requires 256 fault injections to recover the entire secret key; and (3) Resetting a word register to 0 at the output of the substitution layer via 128 (for 1-bit faults), 16 (for 1-byte fault), or 2 (for 64-bit word fault) fault injection, respectively, to recover the entire key. Joshi and Mazumdar [42] also propose the following strategy to perform an SSFA on ASCON. The attack is performed in two phases. In Phase 1, the SSFA is performed using key partitioning, where the 128-bit key is first divided into n-bit subkeys, and their individual parities are calculated instead of using the key-bits directly. A set of key hypothesis for the ASCON S-box is considered and the parity of each set component is estimated. In Phase 2, key analysis using partition parity is performed once the correct parity of the key hypothesis is determined. The key combinations that do not satisfy the parity are eliminated. This helps in reducing the search space for each of the target 64 S-boxes, and key bits are recovered by solving linear equations for the remaining components of the key hypothesis set to get a 128-bit key. The process is repeated for subsequent sets of key hypothesis if the correct key is not determined.

Ramezanpour et al. [43] propose another statistical fault analysis attack called the **fault intensity map analysis** (**FIMA**) for a software implementation of ASCON. This attack can retrieve the entire 128-bit key of ASCON. The attack is designed to use different features such as faulty ciphertexts, SIFA-induced correct ciphertexts, and data-dependent bias to recover the secret key. It is also resilient several countermeasures such as error detection techniques for DFA where it can gather secret information from the increased sample size. Even with infective countermeasures, where a fault is injected in a wrong random round of ASCON, FIMA can recover the secret key with 453 data samples. Thus, compared to other fault analysis methods, FIMA is 6 times more powerful.

Ambili and Jose [44] propose an upgraded design of ASCON-128a using pseudo-randomness of **Cellular Automata** (**CA**) to make the cipher resilient against SIFA and SSFA, verified mathematically in a software implementation. CA is a technique where a particular cell updates its value every iteration depending upon its state and a set of predefined rules. In [44], a null boundary CA (where the farthest cell neighbor is set to 0) is used to protect the architecture of ASCON by implementing it in the pseudorandom function of ASCON permutation. The security against SIFA and SSFA is due to the induced randomness in the linear layer due to which the XOR/linear equations derived for erroneous bits cannot be solved reliably. The authors propose the practicality of their work in hardware implementation as future work.

Kaur et al. [45] propose low-cost error-detection mechanisms as countermeasures against fault attacks for the hardware implementations of ASCON. Parity, interleaved parity, and CRC-3-based techniques are formulated and applied to the 5-bit S-box of ASCON to detect natural and transient faults injected that may occur during the S-box operation to generate faulty outputs. Two kinds of error-detection implementations are introduced, either using Boolean logic or using the **Look-up Tables** (**LUTs**). The error coverage of the proposed error detection schemes is tested for 640, 000 injected faults and is determined to be over 99.99% [45]. The overhead results for the hardware implementations of ASCON on Spartan-7 and Kintex-7 FPGAs, protected using these error-detection mechanisms, show an increase in the area overhead up to 15% for both types of implementations. The results have been tabulated in Table 2. These mechanisms aim at detecting most of injected single and multiple-bit faults leveraged in DFA and SSFA, however, detecting SIFA could be challenging since the error detection is performed at the output of the S-box and not the input.

4.2 Power Analysis

The research work of [46] introduces a ML based **side-channel analysis with reinforcement learning** (**SCARL**) to obtain confidential data by using unsupervised learning to extract leakage models from power measurements. SCARL attempts to obtain the secret key by analyzing the power consumed by the non-linear S-box computations in the initialization phase of ASCON. An autoencoder to process power measurement samples and reinforcement learning along with actor-critic networks are used to cluster the power features. The hardware implementation of the ASCON-128 on the Artix-7 FPGA board is attacked using SCARL, where FOBOS is used to gather the power measurements of 64 S-box computations. The authors successfully demonstrate that their proposed SCARL strategy can recover the secret key of the implemented ASCON-128 cipher on Artix-7 FPGA by using power measurements obtained during 24, 000 encryption operations; the first 4 bits of the secret key are obtained using SCARL within 8 minutes.

The SCA countermeasure assessment based on power leakage is performed using FOBOS 2 and is presented in [32]. **Test vector leakage assessment** (**TVLA**) results for protected and unprotected ASCON architecture using Artix-7 and Spartan-6 FPGA boards. Significant leakage is noticed in the unprotected version vs. the protected version in which the values are within the threshold value; however, no confidential data is recovered through power leakage. The χ^2-test

is also performed for the protected and unprotected ASCON architecture for leakage assessment flow for fixed and random frequency classes using the same test vectors as that for TVLA. Similar to TVLA, the χ^2-test observes leakage in the unprotected ASCON while no leakage is observed in the protected version.

ASCON's initialization phase is the most vulnerable to power analysis attacks as only 2 input bits (out of 5) of the initial S-boxes are unknown (secret key bits) while the other 3 are known. In [29], in addition to optimized hardware implementations of ASCON, the authors propose countermeasures against side-channel analysis attacks, particularly first-order DPA attacks. This is achieved by using the TI scheme [47, 48], a masking technique where the calculations on critical data are indirectly carried out by modified transformations called shares. The proposed protected implementation of ASCON efficiently applies three shares like in the Keccak since ASCON uses an affine transformation of Keccak's χ function [48]. ASCON's linear layer can be implemented on each share. However, the non-linear S-box layer needs to be transformed such that it maintains the following properties (1) Correctness - the sum of the resulting output share matches the S-box output when applied to the sum of the input shares, (2) Non-completeness - each of the three S-box functions is independent of at least one input share, and (3) Uniformity - each S-box function is invertible. The research work [29] also implements a three-share TI version of the ASCON-fast and ASCON-x-low-area variants. The ASCON-fast-TI is a microcontroller-based implementation proposed as a cryptographic co-processor where the initial state sharing and randomness are performed by the microcontroller. The ASCON-x-low-TI variant directly uses the output of an available random number generator in the S-box operation per cycle. The hardware implementation results of the TI-protected implementation are listed in Table 2.

In [49], the **correlation power analysis** (**CPA**) and DPA attacks are mounted on the parallel implementations of ASCON-128 and the TI-protected ASCON-128, respectively. The CPA attack is successfully implemented on ASCON-128 by attacking the vulnerabilities at the end of the initialization phase while requiring fewer power traces to obtain half of the secret key. These vulnerabilities in the initialization round are leveraged again for the attack on the TI-protected ASCON-128 by using the difference of skewness as the third-order attack [49].

Similarly, the research work of [50] implements a TI-based protection scheme for ASCON against first-order DPA by executing one round in 7 clock cycles. The implemented scheme instantiates a single hybrid 2-share/3-share TI-protected 64-bit AND module which uses random 192 bits every clock cycle - 128 for resharing between 2-/3- shares, and the remaining 64 bits are used to maintain TI uniformity. This TI-protected ASCON implementation is shown to be resistant to first-order DPA by analyzing the results of the t-test leakage detection test [50].

5 Our Insights, Visions, and Conclusion

This survey is the first work on the current standard for lightweight cryptography, standardized in 2023. This study covers various hardware implementations proposed for NIST LWC winner ASCON in the recent years on FPGA and ASIC hardware platforms. These hardware implementations suggest improvements on the original design in terms of area, throughput, efficiency or energy/power utilizations for applications of ASCON in resource-constrained devices. Differential and side-channel cryptanalysis performed on ASCON on both hardware and software platforms have also been reviewed. The differential cryptanalysis techniques highlight the vulnerabilities present in the permutation function of the reduced-round ASCON but not of a full 12-round ASCON. The S-box design is also shown to be vulnerable to SFA due to its design where the secret key is retrieved using correlation between the input and the output bits. Power analysis attacks were also mounted on ASCON using machine learning strategies or DPA to gather secret information

from side-channel leakage, however, protected architecture of ASCON using TI is considered safe against such attacks.

An essential insight from our research is the potential benefit of incorporating a design-for-low-cost fault diagnosis in ASCON implementations. This approach prioritizes error detection and countermeasure integration during the initial design phase, rather than as a post-facto solution. By adding low-overhead error detection mechanisms from the outset, ASCON architectures can achieve a more robust and cost-effective defense against cryptographic attacks.

Looking forward, we identify a significant research opportunity in the area of combined fault and side-channel attack strategies, specifically targeted for ASCON. Despite that there has been little prior work and none considers ASCON [51]–[54], we believe that exploring combined attack methodologies will be crucial in advancing cryptographic security. Developing integrated countermeasures that address both fault and side-channel vulnerabilities simultaneously could mark a significant evolution in cryptographic defense mechanisms.

Furthermore, as quantum computing becomes more prevalent, quantum-resistant features in ASCON are becoming increasingly important. ASCON's resilience to quantum-based cryptanalytic attacks should be quantified and improved through research. Additionally, using cross-disciplinary approaches, including concepts from network security, software engineering, and behavioral science, might lead to unique insights about combined attacks. Lastly, research should also focus on the standardization process and policy implications of these attack strategies, collaborating closely with standardization bodies to develop guidelines for evaluating the security of cryptographic algorithms.

Developing integrated countermeasures that address both fault and side-channel vulnerabilities simultaneously could mark a significant advancement in cryptographic defense mechanisms, ensuring that ASCON and similar cryptographic standards remain secure in an increasingly complex digital landscape.

The authors also note that several emerging and related attack methodologies require consideration to ensure the cipher's security in future. Cryptanalysis using machine learning approaches, especially deep neural networks, posts an evolving threat that could enhance current power analysis attacks by leveraging automation strategies such as feature extraction and pattern recognition [55]. Alternatively, template and profiling attack strategies using advanced statistical methods could also be used to enhance traditional power cryptanalysis against ASCON implementations [56]. Emerging laser-based or electromagnetic pulse-based fault-injection techniques may present new challenges in addition to traditional differential fault analysis. In addition to the aforementioned methods, deep-learning-based fault analysis against AEAD schemes also pose potential threat to security of ASCON [57–59]. Moreover, as the number of IoT devices increases, and the attacks become sophisticated, studying composite attacks where multiple attack vectors/methodologies are combined, such as fault injection combined with ML-based SCA and so on, could also emerge as potential threats against ASCON. Thus, monitoring these evolving attack methodologies is essential to maintain secure applicability of ASCON in diverse IoT and deeply-embedded environments.

References

[1] A. Alabdulatif , N. N. Thilakarathne , Z. K. Lawal , K. E. Fahim, and R. Y. Zakari. 2023. Internet of nano-things (IoNT): A comprehensive review from architecture to security and privacy challenges. *Journal of Sensors*. 23, 5 (2023), 2807.

[2] A. Al-ahdal and N. Deshmukh. 2020. A systematic technical survey of lightweight cryptography on iot environment. *International Journal of Scientific & Technology Research* 9, 3 (2020).

[3] Amrita, C.P. Ekwueme, I.H. Adam, and A. Dwivedi. 2024. Lightweight cryptography for internet of things: A review. *Journal. EAI Endorsed Transactions on Internet of Things* 10 (2024).

[4] J. Soto-Cruz, E. Ruiz-Ibarra, J. Vzquez-Castillo, A. Espinoza-Ruiz, A. Castillo-Atoche, and J. Mass-Sanchez. 2025. A survey of efficient lightweight cryptography for power-constrained microcontrollers. *J-Technologies*, 13, 1 (2025), 3.

[5] J. Kaur, M. Mozaffari Kermani, and R. Azarderakhsh. 2022. Hardware constructions for lightweight cryptographic block cipher QARMA with error detection mechanisms. *IEEE Transactions on Emerging Topics in Computing* 10, 1 (2022), 514–519.
[6] J. Kaur, A. Sarker, M. Mozaffari Kermani, and R. Azarderakhsh. 2022. Hardware constructions for error detection in lightweight welch-gong (WG)-oriented streamcipher WAGE benchmarked on FPGA. *IEEE Transactions on Emerging Topics in Computing* 10, 2 (2022), 1208–1215.
[7] A. Aghaie, M. Mozaffari Kermani, and R. Azarderakhsh. 2018. Reliable and fault diagnosis architectures for hardware and software-efficient block cipher KLEIN benchmarked on FPGA. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37, 4 (2018), 901–905.
[8] S. Subramanian, M. Mozaffari Kermani, R. Azarderakhsh, and M. Nojoumian. 2017. Reliable hardware architectures for cryptographic block ciphers LED and HIGHT. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 36, 10 (2017), 1750–1758.
[9] P. Ahir, M. Mozaffari Kermani, and R. Azarderakhsh. 2017. Lightweight architectures for reliable and fault detection simon and speck cryptographic algorithms on FPGA. *ACM Transactions on Embedded Computing Systems* 16, 4 (2017), 109:1–109:17.
[10] A. Aghaie, M. Mozaffari Kermani, and R. Azarderakhsh. 2017. Fault diagnosis schemes for low-energy block cipher Midori benchmarked on FPGA. *IEEE Transactions on Very Large Scale Integrated (VLSI) Systems* 25, 4 (2017), 1528–1536.
[11] M. Mozaffari Kermani and R. Azarderakhsh. 2013. Efficient fault diagnosis schemes for reliable lightweight cryptographic ISO/IEC standard CLEFIA benchmarked on ASIC and FPGA. *IEEE Transactions on Industrial Electronics* 60, 12 (2013), 5925–5932.
[12] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schlffer. 2021. ASCON v1.2: Lightweight authenticated encryption and hashing. *J. Cryptol* 34 (2021), 1–42.
[13] T. Beyne, Y. L. Chen, C. Dobraunig, B. Mennink. 2021. Elephant v2. 1-55. [Online] Available: Retrieved from https://www.esat.kuleuven.be/cosic/elephant
[14] S. Banik, A. Chakraborti, A. Inoue, T. Iwata, K. Minematsu, M. Nandi, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo. 2020. GIFT-COFB. *J. Cryptology ePrint Archive* (2020), 732–768. [Online] Available: Retrieved from https://eprint.iacr.org/2020/738
[15] M. Hell, T. Johansson, A. Maximov, W. Meier, J. Sönnerup, and H. Yoshida. 2021. Grain-128AEADv2. *A submission to the NIST Lightweight Cryptography Standardization Process* (2021), 1–38. Available [Online]: Retrieved from https://grain-128aead.github.io
[16] C. Dobraunig, M. Eichlseder, S. Mangard, F. Mendel, and T. Unterluggauer. 2017. ISAP – Towards side-channel secure authenticated encryption. In *IACR Transactions on Symmetric Cryptology)* 2017, 80–105.
[17] Z. Bao, A. Chakraborti, N. Datta, J. Guo, M. Nandi, T. Peyrin, and K. Yasuda. 2021. PHOTON-beetle authenticated encryption and hash family. *A Submission to the NIST Lightweight Cryptography Standardization Process* (2021), 1–115. Available [Online]: Retrieved from https://www.isical.ac.in/lightweight/beetle.
[18] C. Guo, T. Iwata, M. Khairallah, K. Minematsu, and T. Peyrin. 2021. Romulus v1. 3. *A Submission to NIST Lightweight Cryptography* (2021), 1–57. Available [Online]: Retrieved from https://romulusae.github.io/romulus
[19] C. Beierle, A. Biryukov, L. Cardoso dos Santos, J. Groschdl, L. Perrin, A. Udovenko, V. Velichkov, and Q. Wang. 2020. Lightweight AEAD and hashing using the sparkle permutation family. *J. ToSC* (2020), 208–261.
[20] H. Wu and T. Huang. 2021. TinyJAMBU: A family of lightweight authenticated encryption algorithms (version 2). *A submission to the NIST Lightweight Cryptography Standardization Process* (2021), 1–40. Available [Online]: Retrieved from https://csrc.nist.gov/Projects/lightweight-cryptography/finalists
[21] J. Daemen, S. Hoffert, M. Peeters, G. Van Assche, and R. Van Keer. 2020. Xoodyak - a lightweight cryptographic scheme. *J. ToSC* (2020), 60–87.
[22] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, D. Stehl. 2017. CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In *Proceedings of the IEEE European Symposium on Security and Privacy*. 353–367.
[23] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehl. 2018. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 1 (2018), 238–268.
[24] P. A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. 2018. Falcon: Fast-fourier lattice-based compact signatures over NTRU. *Submission to the NIST's Post-quantum Cryptography Standardization Process* 36, 5 (2018), 1–75.
[25] D. J. Bernstein, A. Hlsing, S. Klbl, R. Niederhagen, J. Rijneveld, and Peter Schwabe. 2019. The SPHINCS+ signature framework. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security.* 2129–2146.
[26] H. Madushan, I. Salam, and J. Alawatugoda. 2022. A review of the NIST lightweight cryptography finalists and their fault analyses. *J. Electronics* 11, 24 (2022), 4199.

[27] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schlffer. 2015. Cryptanalysis of ascon. In *Proceedings of the Cryptographers' Track at the RSA Conference*. 371–387.
[28] A. Kandi, A. Baksi, T. Gerlich, S. Guilley, P. Gan, J. Breier, A. Chattopadhyay, R. R. Shrivastwa, Z. Martinasek, and S. Bhasin. 2023. Hardware implementation of ASCON. Available [Online]: Retrieved from https://csrc.nist.gov/csrc/media/Events/2023/lightweight-cryptography-workshop-2023/documents/accepted-papers/07-hardware-implementation-of-ascon.pdf
[29] H. Gross, E. Wenger, C. Dobraunig, and C. Ehrenhfer. 2017. Ascon hardware implementations and side-channel evaluation. *J. Microprocessors and Microsystems* 52 (2017), 470–479.
[30] X. Wei, M. El-Hadedy, S. Mosanu, Z. Zhu, W. -M. Hwu, and X. Guo. 2022. RECO-HCON: A high-throughput reconfigurable compact ASCON processor for trusted IoT. In *Proceedings of the 2022 IEEE 35th International System-on-Chip Conference*, 1–6.
[31] A. R. Alharbi, A. Aljaedi, A. Aljuhni, M. K. Alghuson, H. Aldawood, and S. S. Jamal. 2024. Evaluating ascon hardware on 7-series FPGA devices. *J. IEEE Access* (2024), 1–1.
[32] S. Khan, W. K. Lee, and S. O. Hwang. 2022. Evaluating the performance of ascon lightweight authenticated encryption for AI-enabled IoT devices. In *Proceedings of the 2022 TRON Symposium*. 1–6.
[33] S. Khan, K. Inayat, F. B. Muslim, Y. A. Shah, M. U. R. Atif, A. Khalid, M. Imran, and A. Abdusalomov. 2024. Securing the IoT ecosystem: ASIC-based hardware realization of ascon lightweight cipher. *International Journal of Information Security* 23, 6 (2024), 3653–3664.
[34] K. Raj and S. Bodapati. 2022. FPGA based light weight encryption of medical data for IoMT devices using ASCON cipher. In *Proceedings of the 2022 IEEE International Symposium on Smart Electronic Systems*. 196–201.
[35] S. Khan, W. K. Lee, and S. O. Hwang. 2021. Scalable and efficient hardware architectures for authenticated encryption in IoT applications. *IEEE Internet of Things* 8, 14 (2021), 11260–11275.
[36] W. Diehl, A. Abdulgadir, F. Farahmand, J.-P. Kaps, and K. Gaj. 2018. Comparison of cost of protection against differential power analysis of selected authenticated ciphers. *J. Cryptography* 2, 3 (2018), 1–26.
[37] A. Abdulgadir, W. Diehl, and J. P. Kaps. 2019. An open-source platform for evaluation of hardware implementations of lightweight authenticated ciphers. In *Proceedings of the International Conference on ReConFigurable Computing and FPGAs*. 1–5.
[38] N. Roussel, O. Potin, G. Di Pendina, J. -M. Dutertre, and J. -B. Rigaud. 2022. CMOS/STT-MRAM based ascon LWC: A power efficient hardware implementation. In *Proceedings of the 2022 29th IEEE International Conference on Electronics, Circuits and Systems*. 1–4.
[39] K. Ramezanpour, P. Ampadu, and W. Diehl. 2019. A statistical fault analysis methodology for the ascon authenticated cipher. In *Proceedings of the 2019 IEEE International Symposium on Hardware Oriented Security and Trust*. 41–50.
[40] G. Surya, P. Maistri, and S. Sankaran. 2020. Local clock glitching fault injection with application to the ASCON cipher. In *Proceedings of the 2020 IEEE International Symposium on Smart Electronic Systemsv*. 271–276.
[41] P. Joshi and B. Mazumdar. 2021. SSFA: Subset fault analysis of ASCON-128 authenticated cipher. *J. Microelectronics Reliability* 123 (2021), 114155:1–14.
[42] P. Joshi and B. Mazumdar. 2019. SSFA: Subset fault analysis on ASCON. *J. Cryptology ePrint Archive* (2019), 1370:1–3.
[43] K. Ramezanpour, P. Ampadu, and W. Diehl. 2019. FIMA: Fault intensity map analysis. In *Proceedings of the International Workshop on Constructive Side-Channel Analysis and Secure Design*. 11421 (2019), 63–79.
[44] K. N. Ambili and J. Jose. 2022. Reinforcing lightweight authenticated encryption schemes against statistical ineffective fault attack. *J. Cryptology ePrint Archive* (2022), 41–59.
[45] J. Kaur, M. Mozaffari Kermani, and R. Azarderakhsh. 2022. Hardware constructions for error detection in lightweight authenticated cipher ASCON benchmarked on FPGA. *IEEE Transactions on Circuits and Systems II: Express Briefs* 69, 4 (2022), 2276–2280.
[46] K. Ramezanpour, P. Ampadu, and W. Diehl. 2020. SCARL: Side-channel analysis with reinforcement learning on the ascon authenticated cipher. 1–25. [Online]. Available: Retrieved from https://arxiv.org/abs/2006.03995v1
[47] S. Nikova, C. Rechberger, and V. Rijmen. 2006. Threshold implementations against side- channel attacks and glitches. In *Proceedings of the International Conference on Information and Communications Security*. 529–545.
[48] B. Bilgin, J. Daemen, V. Nikov, S. Nikova, V. Rijmen, and G. Van Assche. 2014. Efficient and first-order DPA resistant implementations of Keccak. In *Proceedings of the Smart Card Research and Advanced Applications: 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers 12*, 187–199.
[49] N. Samwel and J. Daemen. 2017. DPA on hardware implementations of ascon and keyak. In *Proceedings of the Computing Frontiers Conference*, 415–424.
[50] W. Diehl, A. Abdulgadir, F. Farahmand, J.-P. Kaps, and K. Gaj. 2018. Comparison of cost of protection against differential power analysis of selected authenticated ciphers. *J. Cryptography* 2, 3 (2018), 1–26.
[51] F. Regazzoni, T. Eisenbarth, L. Breveglieri, P. Ienne, and I. Koren. 2008. Can knowledge regarding the presence of countermeasures against fault attacks simplify power attacks on cryptographic devices? In *Proceedings of the 2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems*. 202–210.

[52] F. Regazzoni, T. Eisenbarth, J. GroschŁadl, L. Breveglieri, P. Ienne, I. Koren, and C. Paar. 2007. Power attacks resistance of cryptographic S-Boxes with added error detection circuits. In *Proceedings of the 2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems*. 508–516.
[53] J. Dofe, H. Pahlevanzadeh, and Q. Yu. 2016. A comprehensive FPGA-based assessment on fault-resistant AES against correlation power analysis attack. *J. Electronic Testing* 32, 5 (2016), 611–624.
[54] H. Pahlevanzadeh, J. Dofe, and Q. Yu. 2016. Assessing CPA resistance of AES with different fault tolerance mechanisms. In *Proceedings of the 2016 21st Asia and South Pacific Design Automation Conference* (2016), 661–666.
[55] M. Degr, P. Derbez, L. Lahaye, and A. Schrottenloher. 2024. New models for the cryptanalysis of ASCON. *J. Cryptology EPrint Archive* (2024), 298.
[56] S.-C. You, M. G. Kuhn, S. Sarkar, and F. Hao. 2023. "Low trace-count template attacks on 32-bit implementations of ASCON AEAD", *J. TCHES* 4 (2023), 344–366.
[57] L. Weissbart and S. Picek. 2023. Lightweight but not easy: Side-channel analysis of the ascon authenticated cipher on a 32-bit microcontroller. *J. Cryptology EPrint Archive* (2023), 1598.
[58] M. Das and B. Mazumdar. 2024. Security analysis of ASCON cipher under persistent faults. *Cryptology EPrint Archive*. Retrieved from https://eprint.iacr.org/2024/2030
[59] A. Rezaeezade, A. Basurto-Becerra, L. Weissbart, and G. Perin. 2024. One for all, all for ascon: Ensemble-based deep learning side-channel analysis. *Lecture Notes in Computer Science* (2024), 139–157.
[60] B. Halak and J. Duarte-Sanchez. 2020. Cube attack on a trojan-compromised hardware implementation of ascon. In *Proc. IEEE Inter. SOCC*. 43–47.

Received 12 April 2023; revised 3 June 2025; accepted 7 June 2025