6-2023

Residual Vulnerabilities to Power side channel attacks of lightweight ciphers cryptography competition Finalists

Aurelien Mozipo
Portland State University

John M. Acken
Portland State University

Citation Details

Mozipo, A. T., & Acken, J. M. (2023). Residual vulnerabilities to power side channel attacks of lightweight ciphers cryptography competition finalists. IET Computers & Digital Techniques.

Received: 5 December 2022 | Revised: 23 February 2023 | Accepted: 27 April 2023

IET Computers & Digital Techniques

DOI: 10.1049/cdt2.12057

The Institution of Engineering and Technology WILEY

REVIEW

Residual vulnerabilities to power side channel attacks of lightweight ciphers cryptography competition finalists

Aurelien T. Mozipo[1] | John M. Acken[2]

[1]Department of Electrical and Computer Engineering, Portland State University and Intel Corporation, Portland, Oregon, USA

[2]Department of Electrical and Computer Engineering, Portland State University, Portland, Oregon, USA

Correspondence

Aurelien T. Mozipo.
Email: mozipo@pdx.edu

Abstract

The protection of communications between Internet of Things (IoT) devices is of great concern because the information exchanged contains vital sensitive data. Malicious agents seek to exploit those data to extract secret information about the owners or the system. Power side channel attacks are of great concern on these devices because their power consumption unintentionally leaks information correlatable to the device's secret data. Several studies have demonstrated the effectiveness of authenticated encryption with advanced data, in protecting communications with these devices. A comprehensive evaluation of the seven (out of 10) algorithm finalists of the National Institute of Standards and Technology (NIST) IoT lightweight cipher competition that do not integrate built-in countermeasures is proposed. The study shows that, nonetheless, they still present some residual vulnerabilities to power side channel attacks (SCA). For five ciphers, an attack methodology as well as the leakage function needed to perform correlation power analysis (CPA) is proposed. The authors assert that Ascon, Sparkle, and PHOTON-Beetle security vulnerability can generally be assessed with the security assumptions "Chosen ciphertext attack and leakage in encryption only, with nonce-misuse resilience adversary (CCAmL1)" and "Chosen ciphertext attack and leakage in encryption only with nonce-respecting adversary (CCAL1)", respectively. However, the security vulnerability of GIFT-COFB, Grain, Romulus, and TinyJambu can be evaluated more straightforwardly with publicly available leakage models and solvers. They can also be assessed simply by increasing the number of traces collected to launch the attack.

KEYWORDS

cryptography, internet of things, leakage currents, power consumption, security of data, telecommunication security

1 | INTRODUCTION

Due to the exponential rise of communication networks implemented on small Internet of Things (IoT) devices, there has been an urgent need to secure these networks to protect both consumers' and cloud service providers' private information. With the implementation of cryptographic algorithms, the need arises to protect them against malicious attacks. Power side channel attacks (SCAs) are of great concern on IoT devices. This is stemming from the fact that malicious agents can implement power measurements and run cryptanalysis algorithms such as differential power analysis (DPA) to extract secret information from the device. Although power SCAs have been extensively studied, they have been applied mainly to the advanced encryption standard (AES) for regular full power applications. The AES is not suited for IoT devices because of its complexity and power dissipation. Multiple lightweight, low-power, compact cipher algorithms have been proposed for such devices. Likewise, traditional countermeasures against a power SCA proposed for AES implementations yield relatively significant area, performance, and power overheads when implemented on lightweight ciphers such as SIMON, PRINCE, and PRESENT. But there are optimal countermeasures or modes of operation targeted for lightweight

ciphers that lead to acceptable results. Particularly, SIMON with a round unrolled datapath architecture that enhances vulnerability against a power SCA and yet increases throughput and reduces energy per encryption (pJ/encryption), has been presented in [1, 2]. Likewise, PRINCE with unrolled architecture implementation with countermeasures against power SCA has also been proposed [3].

Although these lightweight ciphers represent a viable and safe alternative to the power-hungry AES, their proliferation and the indecision in the industry around the choice of a common encryption technique and mode of operation have prompted the US National Institute of Standards and Technology (NIST) to undertake the creation of standard, resilient lightweight ciphers. They should encompass confidentiality, security, and authentication. They must either have built-in countermeasures to side channel attacks or show a strong resistance against power SCAs through the algorithm constructs [4].

1.1 | Relevant types of side channel attacks

There are numerous types of side channel attacks published in literature. The ones relevant to IoT devices and our study are local versus remote attacks and passive versus active attacks. With local attacks, the malicious agent has physical access to the target device to capture the measurements needed to perform side channel analysis. In the case of remote attacks, the agent can sense the leaked information remotely with no physical access to the device. Passive attacks occur when the device naturally and unintentionally leaks side channel information to the outside world, in the course of its normal operation. However, with active attacks, the malicious agent has to modify the device's intended behaviour to forcibly produce or alter the side channel information.

The following passive side channel information is most likely to be leaked by IoT devices implementing cryptographic algorithms, leading to the undermentioned types of attacks (Figure 1):

- Power consumption or temperature rise: local and remote power side channel attacks [3, 5].
- Electromagnetic emanation: local electromagnetic interference attacks [6].
- Programme execution time of circuit delay: remote and local timing attacks
- Scanning electron microscope (SEM) of device layout: local SEM attacks [7].

Active attacks force the alteration or leakage of the following side channel information, which leads to the undermentioned types of attacks (Figure 1):

- Execution time or circuit delay: glitch attacks, rowhammer [8] attacks, and microarchitecture attacks [9].
- Power and clock glitch: local power and clock glitch attacks [10].

1.2 | Relevant studies on the security of IoT communications

Much of this research is focused on studying the residual vulnerabilities to power SCA of the NIST lightweight ciphers cryptography competition (LWC) finalists. First, we are dedicating this section to introducing similar relevant prior art as well as the necessary background knowledge helpful to readers in understanding the concepts at hand. Many researchers have published studies to address the security challenges of lightweight cryptographic protocols. The authors of

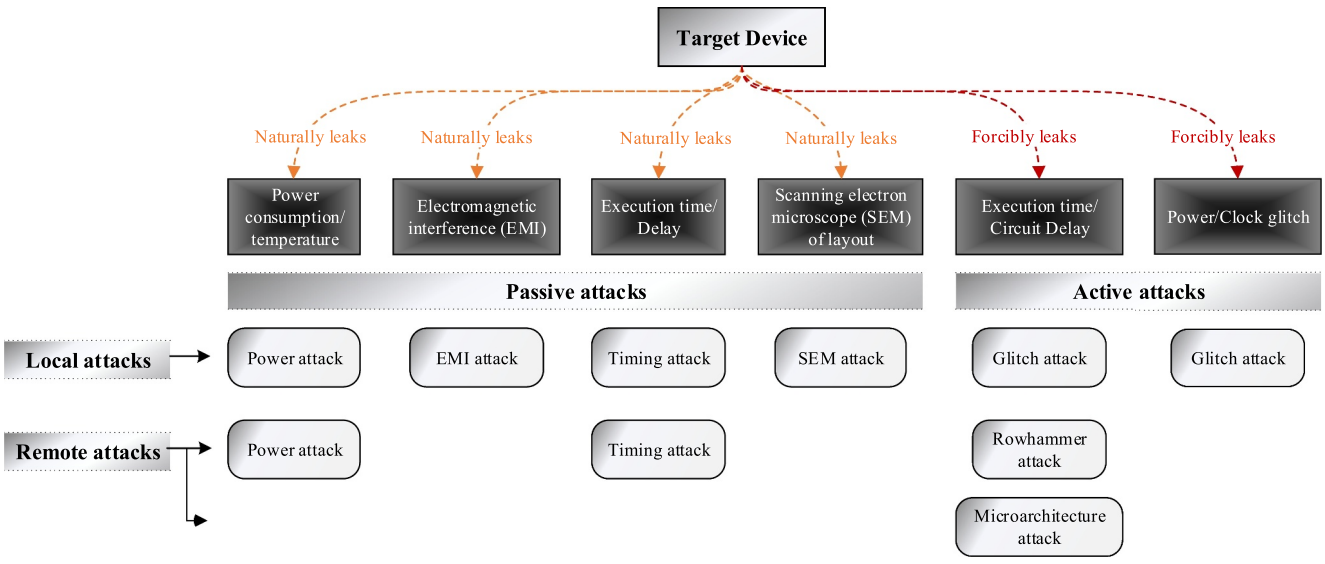

FIGURE 1 Classification of relevant types of side channel attacks.

[11] published a comparative survey of lightweight cryptographic algorithms, their strengths, weaknesses, and general security requirements, such as integrity, confidentiality, and authentication. [12] compared the 32 LWC second-round candidates for features such as performance and power. [13] focuses on surveying certain lightweight block ciphers that can easily be implemented in resource challenge devices; such ciphers include PRESENT, SIMON, and GRAIN. The authors of [14] propose a SCA categorisation system, particularly for enabling analysis of SCA on mobile devices. The study goal is also the facilitation of the development of new countermeasures.

Protecting the integrity of the communications between IoT devices goes beyond the protection of the device themselves. Malicious agents have also intercepted the communications and tried to exploit the weaknesses in the protocol. [15] proposes a survey of existing protocols and analyses methods to establish secure communications between IoT devices. Direct attacks on lightweight cipher implementations are also a threat to IoT devices' data. [16] proposes a differential attack on the family of lightweight block ciphers SKINNY. [17] has demonstrated a successful collision fault attack on GIFT with only 64 faulty ciphertexts.

To help understand the theory and algorithms behind power side channel cryptanalysis, the review in [18] and study in [19] provide foundations that summarise the concepts of power analysis distinguishers. They focus on distinguishers used in non-template attacks, including correlation power analysis (CPA), which is one of the most efficient distinguishers. They also introduce the notion of test vector leakage assessment (TVLA). TVLA, based on Welch's *t*-test, uncovers leakage of information without mounting an attack. Other distinguishers summarised in the paper are simple power analysis (SPA), differential power analysis (DPA), and mutual information analysis (MIA).

However, none of these prior studies address the issue of resistance to power side channel attacks of the LWC finalists, thus our analysis is the first with such a goal.

1.3 | Related surveys and work on side channel attacks of lightweight ciphers

In this section, we discuss surveys of IoT and mobile devices, as well as surveys on SCA distinguishers, applied to lightweight ciphers. We also present studies dealing with multiple cryptanalysis aspects of a single lightweight cipher. Multiple prior arts have also performed comparative studies of SCA on multiple lightweight ciphers, which we are also summarising in this section. Table 1 summarises the prior art covering surveys and studies on lightweight ciphers' vulnerabilities to power SCA, with references for readers.

TABLE 1 Previous surveys/work on side channel attacks on symmetric ciphers.

Publication year	Article	Main topic covered
2021	Khan M N [11]	Lightweight cryptographic protocols, focusing on IoT devices
2018	Spreitzer R [14]	Classification of side channel attacks, focusing on mobile devices
2020	Randolph M [18]	Exploration of the foundation of power SCA distinguishers.
2020	Fei Y [20]	Evaluation of WAGE vulnerability to CPA and comparison with LWC competition 2nd round candidates.
2022	Liu Z [21]	Root cause of power leakage, compared to AES, in three candidates of LWC competition.
2022	Abdulgadir A [22]	Study the impact on cost and performance, of applying Domain-oriented masking on three LWC competition finalists.
2022	Babinkostova L [23]	Study of side channel leakage of GIFT-COFB by applying CPA with the hamming distance model.
2016	Nalla Anandakumar, N [24]	Study SCA resistance of FPGA implementations of MAC-PHOTON.
2016	Biryukov A [25]	Analysis of the efficiency of common leak functions used in CPA to attack AES and seven lightweight ciphers.
2021	Zhang J [26]	Power attack method against the diffusion layer of GIFT implemented in an FPGA.
2017	Samwel N [27]	Presents first DPA attack on Keyak S-box and first CPA attack on Ascon S-box.
2022	Windarta S [28]	Analysis of cryptographic areas and cryptanalysis attacks of various hash functions suitable for lightweight ciphers.
2022	Batina L [29]	Side channel attack evaluation of software implementations of ASCON, Xoodyak, and ISAP
2021	Miteloudi K [30]	First application of ROCKY as a countermeasure against SCA.
2018	Diehl W [31]	Study of protection against DPA of a few authenticated ciphers.
2017	Heuser A [32]	Study of side channel analysis metrics used to determine resistance to SCA.

1.3.1 | Surveys on IoT and mobile devices

[11] Surveyed lightweight cryptographic protocols focusing on IoT devices. But, the SCA on these protocols is not a focus of the study. However, [14] presents a classification of side channel attacks focusing on mobile devices. They allow and facilitate the development of new countermeasures. But this paper fails to address most lightweight ciphers and certainly not the NIST LWC candidates, which is the focus of our study.

1.3.2 | Surveys and studies on SCA distinguishers applied to lightweight ciphers

The study of [20] evaluates and analyses authenticated lightweight cipher WAGE vulnerability to CPA and compares it against LWC second round candidates. [23] focuses on the study of side channel leakage of GIFT-COFB by applying CPA with the Hamming distance model. Then, they use the attack results to rate the reliability of several side-channel leakage assessment metrics: transparency order, revisited transparency order, and signal-to-noise ratio, amongst others. [24] studies the SCA resistance of FPGA implementations of MAC-PHOTON. They implement three concept architectures (iterative, folding, and unrolling), then analyse their security against the SCA. They also elaborate on MAC-PHOTON Threshold Implementation (TI) resistance against the first-order power analysis. [26] covers power attack methods against the diffusion layer of GIFT implemented in an FPGA. [27] presents the first DPA attack on a Keyak S-box and the first CPA attack on an Ascon S-box. The difference with our work is that we propose a method to attack the 320-bit state of ascon-128. In [30], they show the first application of ROCKY as a countermeasure against SCA, on four architectures of Xoodoo implemented in an FPGA.

Contrary to the above-mentioned studies and surveys that focus only on a single cipher, our study focuses on exposing residual vulnerabilities on multiple LWC finalists, namely all seven ciphers that do not have built-in SCA countermeasures.

1.3.3 | Surveys of the comparative studies of SCA on multiple lightweight ciphers

GIFT-COFB, Xoodyak, and Grain-128, three finalists of the LWC are covered in [21]. This paper studies the root cause of power leakage in those ciphers and compares it to AES. [22] studies the impact on cost and performance, of applying Domain-Oriented Masking on three LWC competition finalists: Elephant, TinyJambu, and Xoodyak. In [25], the authors analyse the efficiency of common leak functions used in CPA to attack symmetric ciphers. The study case is an implementation of AES and seven lightweight ciphers (Fantomas, LBlock, Piccolo, PRINCE, RC5, SIMON, and SPECK) in an 8-bit processor. None of these is amongst the finalists of the NIST LWC, which is the focus of our study. [28] focuses on the analysis of cryptographic areas and cryptanalysis attacks of various hash functions suitable for lightweight ciphers. They have also conducted a comparative study and presented research challenges on hardware and software implementations of those lightweight cryptography hash functions. However, this work does not focus on power SCA. [29] proposes a side channel attack evaluation of software implementations of ASCON, Xoodyak, and ISAP. [31] is a study of protections against DPA of a few authenticated ciphers (ACORN, ASCON, CLOC, SILC, JAMBU, and AES-GCM). In that paper, the authors use TVLA to demonstrate vulnerability to a first-order DPA and to demonstrate improved resistance of the protected versions. Then, they compare the cost of implementing countermeasures on those ciphers. [32] is a study of side channel analysis metrics used to determine resistance to SCA. Particularly, they attack the first, last, and both rounds of several 4-bit S-boxes ciphers (KLEIN, Midori, Mysterion, LED, Piccolo, PRESENT, PRIDE, PRINCE, RECTANGLE, and SKINNY) and 8-bit S-boxes ciphers (AES, Zorro, and Robin).

Amongst the above-mentioned studies that deal with the same ciphers of interest as us, the NIST LWC finalists, a maximum of three ciphers is analysed in any one paper. Therefore, none comprehensively covers the SCA vulnerabilities of all of them; which is what we address in this paper.

1.4 | Organisation of the paper

This paper is organised as follows: Section 2 presents a detailed account of our contribution to knowledge while Section 3 is the background summary which gives the theoretical knowledge necessary to understand the analysis throughout this paper. In sections 4 and 5, we discuss our evaluation of the residual vulnerabilities against power SCA of the seven LWC finalists that do not integrate a built-in countermeasure against side channel attacks. We conclude our analysis in Section 6.

2 | OUR CONTRIBUTION

The novelty of this research resides in the fact that we identify vulnerabilities to power SCA in seven (out of 10) LWC finalists and propose methodologies for attacking five of them. We also propose the leakage functions needed to perform CPA on those lightweight ciphers.

This study defines a method for attacking Ascon by reducing the key search space to a practically implementable size. We also propose a leakage function used in CPA to attempt to uncover the state. Leveraging a methodology shared in [43], we introduce two hamming distance-based leakage functions for attacking the first and last rounds of GIFT_-COFB. We highlight the Hamming distance-based leakage model for attacking GRAIN-128-AEADv2. The study also proposes methodologies for launching power SCA on PHOTON-Beetle, Romulus, and Schwaemm.

The study begins with a comprehensive comparative study and evaluation of the 10 LWC finalists to evaluate their hardware implementations' residual vulnerability against a power SCA. To our knowledge, a study of this kind has never been performed on these ciphers, so this will be the first proposal. Many generalised analyses of lightweight ciphers have been proposed. Some general studies focus on security aspects, performance, power consumption [12], area, and validations of the advertised features of confidentiality, authentication, and integrity [11]. Unlike [12], which proposed a general, broad survey targeting the 32 second-round candidates of the LWC competition, our research goes in-depth into the level of resistance to power side channel attacks, targeting the 10 candidates of the final round. We aim to provide the evaluators of these algorithms, the NIST community, and the IoT device designers with the tools that will help educate and inform on the weaknesses of those algorithms. The authors of [44] have launched a call to side channel security labs to propose an evaluation against side channel attacks of the 10 finalists. Hence, our comprehensive vulnerability evaluation is intended to serve as a lantern to those who aim to develop power side channel attack proposals against the 10 finalists to evaluate their robustness before the final selection by NIST. Some of the residual vulnerabilities uncovered are based on demonstrated, previously published literature. Others are based on our initial theoretical assessment.

3 | BACKGROUND

To address the critical issue of the standardization of lightweight ciphers, the NIST has initiated a competition to solicit lightweight ciphers suitable for low-power, compact, or otherwise highly constrained devices. After two preliminary selection rounds, the NIST reduced the initial 57 submissions to a final round of 10 candidates.

3.1 | Lightweight cipher competition finalists

Table 2 summarizes the main characteristics of the 10 proposals selected by the NIST for the final round of evaluations. They are based on authenticated encryption with associated data (AEAD), which are symmetric encryption algorithms that provide both confidentiality and authentication. The following three LWC competition finalist algorithms have integrated countermeasures against side channel attacks: ISAP, Elephant, and Xoodyak. Elephant implements masking using linear-feedback shift registers (LSFRs) [34]. ISAP features sponge-based rekeying [37]. Xoodyak's built-in countermeasure, called Cyclist, implements a DPA countermeasure by absorbing the session counter that is used for a nonce. It limits the number of selection functions an attacker can use [42]. However, the other ciphers, Ascon, GIFT-COFB, Grain, PHOTON-Beetle, Romulus, SPARKLE, and TinyJambu, do not feature such built-in side channel protections and will constitute the focus of this work.

3.2 | Security metrics for sample classes of attacks

Before diving into the cipher analysis, let us state some security metrics which are classes of attacks that constitute the basics of some vulnerabilities exposed in a few ciphers.[45]

3.2.1 | Chosen ciphertext attack and leakage in encryption only, with nonce-respecting adversary (CCAL1)

The malicious agent performs several encryption/decryption operations that leak the algorithmic implementation of the authenticated encryption scheme. Then, she or he chooses two new messages and receives the corresponding ciphertexts while measuring the leaked information. The system is considered insecure when the agent can match the ciphertext to the plaintext with a reasonable advantage. The CCAL1 security variant is when the chosen ciphertext has nonce-respecting and leakage is measured during encryption operations only.

3.2.2 | Chosen ciphertext attack and leakage in encryption only, with nonce-misuse resilience adversary (CCAmL1)

Same as CCAL1 but with a fresh challenge nonce.

3.2.3 | Chosen ciphertext attack and leakage in encryption only, with nonce-misuse resilience (CCAmL2)

Same as CCAL1 but with a fresh challenge nonce and leakage during both encryption and decryption.

3.2.4 | Ciphertext integrity with leakage during encryption only (CIL1), with nonce-respecting adversary

For this security metric, the malicious agent also performs encryptions/decryptions while capturing the leaked information. The implementation is considered secure if the malicious agent cannot guess a valid plaintext with good probability. The CIL1 security variant is non-respecting and leaks only during encryption.

3.2.5 | Ciphertext integrity with leakage during encryption and decryption (CIML2), with nonce misuse resistance

Similar to CIL1, except there is no constraint on nonces and leakage during both encryption and decryption.

TABLE 2 Security characteristics of the 10 finalists of the lightweight cipher cryptography competition.

	MAC, hash functions or primitives	Vulnerabilities and cryptanalysis features reinforcing security
Ascon [33] Type: Block cipher; Key size: 128	Ascon-hash, Ascon-HashA	• No countermeasure is implicitly implemented. However, the algorithm architecture offers protection against repeated nonces. • The Ascon round function is amenable to the application of masking.
Elephant [34] Type: Tweakable block cipher (elephant); Key size: 128	A variant of the protected counter sum MAC function	Masked using LSFR
GIFT-COFB [35] Type: Block cipher (GIFT-128) Key size: 128	No integrated hash functionality. If needed, the authors propose a 256-bit hash function from another research.	• Masked using LSFR • The GIFT ascon round function is amenable to the application of masking.
Grain-128AEADv2 [36] AEAD stream cipher; Key size:128	Based on non-linear feedback shift registers (NLFSR) and LSFR pre-output generator	None
ISAP: [37] Isap-A-128a, and Isap-A-128 Isap-K-128a, and Isap-K-128 Key Size:128	320-Bit Ascon-p permutation400-Bit Keccak-p [400] permutation	Sponge based rekeying
PHOTON-beetle authenticated encryption and hash family [38] Key size: 128	P_{256} ($PHOTON_{256}$ hash)	None
Romulus [39] Type: Tweakable block cipher (SKINNY); Key size: 128	Romulus-H	SKINNY round function is amenable to the application of masking.
SPARKLE (SCHWAEMM and ESCH [40] Type: Block cipher; Key size: 128	Esch	• No countermeasure • Collision resistant • Long trail strategy (LTS) provides security against differential and linear cryptanalysis
TinyJambu [41] Type: Block cipher; Key sizes: 128, 192, 256	Keyed permutation P_n N rounds of state update, based on nonlinear feedback shift register	The TinyJambu round function is amenable to the application of masking.
Xoodyak [42] Type: Stream cipher; Key size: ≥ 128	Xoodoo permutations	Built-in countermeasures: • Cyclist: a DPA countermeasure that absorbs the session counter used for a nonce. It limits the number of selection functions an attacker can use. • A key replacement scheme. Instead of a counter, a new key is generated and saved for the next instantiation of Xoodyak • A method similar to "Forget", which is a ratchet mechanism offered by cyclist: Prevents the recovery of the secret key before the use of the ratchet. • Xoodoo round function is amenable to the application of masking. However, masking is implemented in the LWC proposal.

4 | EVALUATION OF RESIDUAL VULNERABILITIES

Several studies have demonstrated the effectiveness of authenticated encryption with advanced data (AEAD), in protecting communications with IoT devices [46–48]. They provide security, authentication, and confidentiality, all in one algorithm implementation. However, the proposed LWC algorithms still displayed residual vulnerabilities against the power SCA, which we expose in the next few sections.

4.1 | Ascon-128/Ascon-128a

Ascon-128 and Ascon-128a are suites of lightweight ciphers that provide AEAD, in addition to hash functions Ascon-Hash and Ascon-Hasha and extendable output functions Ascon-Xof and Ascon-Xofa. The primary recommendation for the NIST competition is the suite set Ascon-128/Hash-128/Hash-Xof. The parameters for this authenticated encryption scheme include a key size and permutation length of 128 and 320 bits, respectively. The algorithm also features two permutations p^a and p^b used in the AEAD and the hash functions of lengths 12 and 6 in the AEAD, and lengths of 12 each in the hashing algorithm [33].

The construction of Ascon has an initialisation stage that generates the state by manipulating the encryption key (K), the initialisation vector (IV), the nonce (N), and the permutation p^a as follows:

$$S \leftarrow IV \,\|K\|\, N \tag{1a}$$

$$S \leftarrow p^a(S) \oplus \left(0^{320-k} \,\|\, K\right) \tag{1b}$$

$$p^a = p_C \circ p_S \circ p_L \tag{1c}$$

where S is the 320-bit state, K is the k-bit key, $k = 128$, p^a is a permutation with a rounds ($a = 12$), p_C is the constant addition layer, p_S is the substitution layer, p_L is the linear diffusion layer and $\|$ represents the concatenation operation.

4.1.1 | Proposed scheme for attacking Ascon-128/Ascon-128a

A power SCA works on the premise of developing a predictable relationship between the algorithm's internal operations, the encryption key, and other input/output data. Thus, based on the initialisation stage in the equations above and the Hamming distance model developed in [43], we propose the following leakage function for an attack on Ascon-128 using the correlation power analysis (CPA) distinguisher:

$$Leak_{ascon-128} = HD\left(p'^a(S) \oplus \left(0^{320-k} \,\|\, K\right), S\right) \tag{2}$$

where $HD(x,y)$ represents the Hamming distance between x and y.

The success of this leakage function in recovering the state largely depends on the signal-to-noise ratio (SNR) of the measurements, which in turn depends on the algorithm implementation. The authors of Ascon have stated that recovering the state during data processing may not directly lead to recovery of the secret key, and recovery of the state during the initialisation stage will lead to recovery of the secret key.

To reduce the complexity of guessing the 320-bit state S, we decompose the guessing phase into 64-bit substates to align with the structure of the 64-bit register words (x_0, x_1, x_2, x_3, x_4):

$$S = x_0 \,\|x_1\|x_2\|x_3\|\, x_4 \tag{3}$$

We can divide the Ascon state S into 5 64-bit words and guess each word by replacing the permutation p^a with the permutation p'^a defined as follows:

$$p'^a = p_C \circ p_L \tag{4}$$

By removing the substitution layer p_S from the permutation p^a, we are ensuring that the result of each substitution p'^a on x_i does not depend on the remaining 4 words. Given that the substitution layer p_S acts on a 5-bit column word across all 5 words x_i, it mixes the 5 64-bit words, and thus its output is no longer solely dependent on the 5-bit words x_i. Thus, the attack on the state is reduced into 64-bit operations, therefore reducing the search from 2^{320} to 5×2^{64}.

4.1.2 | On the confidentiality and integrity of Ascon under the security game "Chosen ciphertext attack and leakage in encryption only, with nonce-misuse resilience adversary (CCAmL1)"

The message processing part of Ascon is simple power analysis (SPA) secure under CCAmL1 assumptions. However, without DPA protected implementation of the verification phase [45], it is possible to successfully attack secure bootloading applications [49] by estimating valid messages without knowledge of the encryption key [50].

4.2 | GIFT-COFB

GIFT-128, which is the block cipher used in GIFT-COFB LWC, is a larger version of PRESENT [51]. Thus, the weaknesses of PRESENT against power SCA are also vulnerabilities of GIFT-COFB against power SCA. PRESENT implements a bit-oriented permutation layer and has a 64-bit block size. Each encryption/decryption round consists of layers AddRoundKey, sBoxLayer (substitution layer), and pLayer (permutation layer). One more key addition is performed after the encryption rounds. Similarly, each encryption round of GFIT-128 (and GIFT-COFB) consists of 3 three layers: SubCells (32-bit state

cell substitution), PermBits (bitwise permutations, different for each 32-bit state cell), and AddRoundKey (round key addition to the state).

Proposed leakage function for attacking GIFT-COFB

The round constructions are also similar to AES rounds, especially the last round which does not feature a MixColumn layer. Thus, we are proposing that GIFT-COFB can be attacked with a CPA targeting the first and/or last round with leak functions defined in the equations below:

$$Leak_{first\_round} = HD\left(PermBits(SubCells(P)) \oplus K_{r\_fisrst}, P\right) \tag{5}$$

$$Leak_{last\_round} = HD\left(inv\_SubCells\left(inv\_PermBits\left(C \oplus K_{r\_last}\right), C\right)\right) \tag{6}$$

where HD represents the Hamming distance, K_r is the round key, P (plaintext) is the input to the first round, and C (ciphertext) is the output of the last round. Successful uncovering of the encryption key in an AES implementation has been demonstrated with practical experiments, with similar leakage functions [43].

Additionally, an analysis performed on 4000 traces of PRESENT in an ASIC without any countermeasure yielded a test vector leakage assessment (TLVA) of 12.28 [52], which is higher than the threshold of 4.5 required by NIST to be accepted for secure cryptographic implementations. This means that the measured traces of GIFT-COFB implementations will be said to carry sensitive distinguishing information that could be exploited by a malicious agent to uncover the secret key.

4.3 | GRAIN-128-AEADv2

The authors of this algorithm proposal have argued that Grain-128a (the raw encryption algorithm of GRAIN-128a-AEADv2) is resistant to a fast correlation attack, the classical method that was designed to exploit the state of the LFSR inside the algorithm [36]. Although [53] have demonstrated successful attacks on smaller grain-like stream ciphers, those attacks do not apply to Grain-128a. Furthermore, a revised fast correlation attack from the same authors revealed that the Grain-128a state can be recovered with data and time complexity of 2^{114} [54]. However, this revised fast correlation attack does not apply to Grain-128a in authentication mode because only every other keystream bit can be recovered by the malicious agent [36].

But, GRAIN-128-AEADv2 is an AEAD stream cipher that derives from GRAIN-128-AEAD [36], which is a common stream cipher previously studied in the literature, from an SCA perspective [55]. For any successful SCA, the malicious agent needs to have a deterministic relationship between the input data and the encryption key. As demonstrated in [43, 55], the Hamming distance model is a very reliable method to estimate the power dissipation of the system for CPA. It is possible to define a leakage model based on the Hamming distance of the state.

Also, it has been demonstrated that one can construct a fast and automated process through Z3, a publicly available satisfiability modulo theory (SMT) solver, with the leakage model and the publicly available keystream, which leads to key recovery in a few seconds [55].

4.4 | PHOTON-beetle

PHOTON-Beetle authenticated encryption and hash are made of the sponged-based mode Beetle and the PHOTON256 permutation [38]. Although its mode is designed to be side channel resistant, PHOTON-Beetle is only strongly protected against SPA without averaging [45]. Given that the nonce repetition is prevented under the CCAL1 and CIL1 security hypothesis [45], the resistance to SPA is thus at its possible maximum. Thus, PHOTON-Beetle can be implemented in the flat, leveled architecture shown in Figure 2.

The following defines the variables used in the above Figure 2. N: nonce, K: master key, M: plaintext divided into m blocks of r bits each, with the last padded with 0's if it is smaller than r, $r = 128$ is the rate of the message absorption, T:

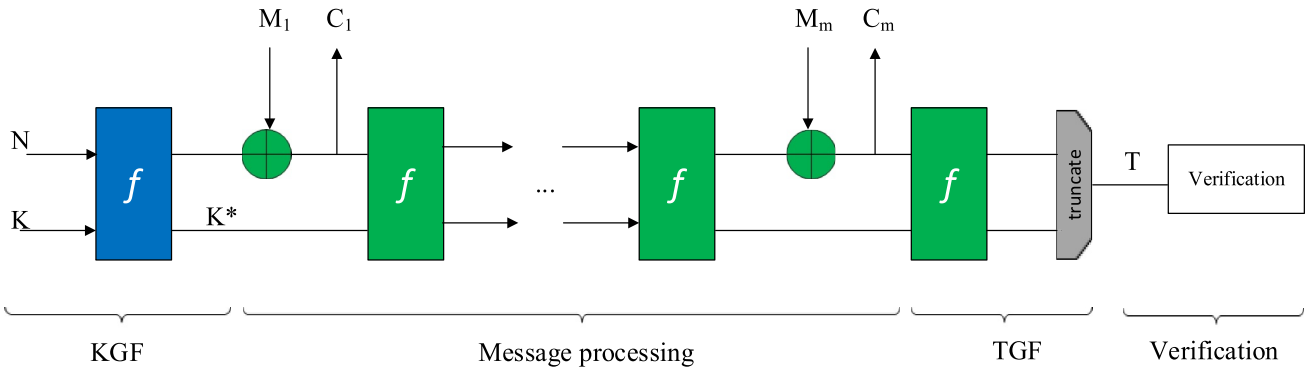

FIGURE 2 PHOTON-Beetle leveled implementation for an m-block message, with CCAL1 and CIL1 security targets.

tag, *f*: PHOTON256 permutation function [38], *KGF*: key generation function, and *TGF*: tag generation function.

However, the PHOTON-Beetle message processing section shows residual vulnerability against DPA with the following scenarios: define a fix nonce and ephemeral key K*, generate multiple plaintexts blocks M_1 or multiple ciphertexts C_1, uncover the capacity
section along with the plaintext M_1 or ciphertext C_1, and then perform the inverse permutation to uncover the key *K*. Thus, more uniform protection is needed to obtain a security level stronger than CCAL1 and CIL1 [45].

Comparing the vulnerability to a power SCA of PHOTON-Beetle S-Box versus Elephant S-Box and GIFT S-Box:

S-Box operations in symmetric cryptographic algorithms are frequently the main target of malicious agents who wish to extract information about the secret key. The authors of [56] have developed theoretical metrics to evaluate the vulnerability against a power SCA of 4x4 S-Box operations in PHOTON-Beetle and several other lightweight ciphers: revisited transparency order (VTO), confusion coefficient variance (CCV), and minimum confusion coefficient (MCC). Based on theoretical analysis, PHOTON-Beetle is the least vulnerable to a power SCA when evaluated with VTO and CCV. Even with the MCC metric, PHOTON still shows the second highest resistance to a non-profiled power SCA among the nine ciphers studied (including NIST LWC finalists GIFT and Elephant). Additionally, practical experiments evaluating the minimum number of traces to achieve 90% confidence of attack showed that PHOTON requires ~800 traces (for a non-profiled attack and noise level of $\log_2(\sigma^2) = 5$) and 300 (for a profiled attack and noise level $\sigma = 2$). For a non-profiled attack, PHOTON ranks second least vulnerable after the Elephant cipher. But for the profiled attack, the position compared to Elephant and GIFT is inconclusive, as it varies depending on the trace noise level [56].

However, although these results might indicate a low level of vulnerability for PHOTON, it is worth pointing out that the number of traces required to reach a high confidence level is nonetheless very low compared to what state-of-the-art attack scenarios are capable of today [43, 49].

In summary: barring the realistic aspect of implementing a practical attack targeting solely PHOTON-Beele S-box operations, a malicious agent will merely need to increase the number of traces to successfully uncover the encryption key.

4.5 | Romulus

Romulus is based on a tweakable block cipher modeled over the SKINNY family of ciphers. Precisely, the version proposed in the LWC competition, Romulus-N, implements a change in the number of rounds compared to SKINNY-128-384. Romulus-N adopts 40 rounds of encryption, which is the same as SKINNY-128-384+ [39]. Similar to GIFT-COFB, Romulus will be vulnerable to the same power SCA methodologies that have been demonstrated on its parent algorithm. Specifically, a power SCA run with a CPA distinguisher and the Hamming distance leakage function, on an unprotected SW implementation of SKINNY-128, has shown that the minimum traces to discover (MTD) is only 80 traces. This means that only 80 traces are required to attack an unmasked SKINNY-128, although a masked version could not be successfully attacked with 1000 traces [57]. However, Romulus' proposal does not integrate masking to protect against an SCA. Furthermore, a power SCA mounted on an HW implementation of SKINNY with a Hamming distance model showed a success rate of close to 100% with only 60 traces [32]. However, masking scheme implementation on SKINNY has shown an increase in the MTD to more than 1000 traces [57]. But, 1000 traces is not much of a deterrent with today's state-of-the-art computers and capture equipment because we have shown capabilities to mount SCAs with over 100 000 traces [43]. Thus, it still goes to show that Romulus implementations will need to be coupled with a countermeasure to be resistant to a power SCA.

4.6 | Sparkle (Esc/schwaemm)

The Sparkle proposal to the NIST LWC is a family of permutations closely related to the block cipher SPARX but with a fixed key and wider block size. The submission comprises the hash functions Esch256 and Esch384, based on the permutation family SPARKLE384 and SPARKLE512, which produce digests of 256 and 384 bits, and yield security levels of 128 and 192 bits respectively. The AEAD cipher family proposed is Schwaemm. The main implementation within the family is Schwaemm256-128, which accepts a key of length 128 bits, a nonce of length 256 bits, and produces a tag of length 128 bits. The encryption construction accepts the plaintext and outputs the ciphertext. Three other variants with different key, nonce, and tag lengths are proposed: 128-128, 192-192, and 256-256 [40].

Figure 3 represents Schwaemm authenticated encryption construction with 3 associated data blocks and 4 message blocks, showing the addition of the whitening block versus Beetle. The function *f* represents one of the permutations Sparkle256_s, Sparkle384_s, or Sparkle512_s; *s* represents the number of steps in the permutation, ρ is the combined feedback function, and $w_{c,r}$ is the whitening function as defined in [40].

The Schwaemm AEAD algorithm is based on a modified version of the Beetle mode for authenticated encryptions. Beetle is based on a duplexed sponge that provides additional security by using combined feedback to create a difference between the ciphertext output and the input of the permutation calls [58]. One of the main differences between Beetle and Schwaemm is that Schwaemm makes use of rate whitening, which consists of XORing the capacity to the rate before the permutation starts, as shown in Figure 3. However, half of the branches in the state are not modified. Another deviation from Beetle is making the Schwaemm key length the same as the capacity, which alters how the tag is handled.

Despite the differences between Beetle and Schwaemm and given that half of the branches in the state are identical, the security of the Schwaemm algorithm follows the security of the

underlying cryptographic algorithms from which it is derived. Specifically, the security of Sparkle is based on the security of sponge-based hashing and the Beetle mode. The differences mentioned above have no impact on the potential relation between the leaked trace and the encryption key. Thus, most vulnerabilities observed on Beetle still apply to Schwaemm [58].

Other vulnerabilities of schwaemm to power SCA

The addition of the whitening function to Schwaemm does not change leakage prevention under the CCAL1 and CIL1 security hypothesis. Thus, the resistance to SPA is maximum as with Beetle. However, Beetle is shown to be vulnerable to DPA under the scenario defined in section 4.5. Therefore, the Schwaemm algorithm will also be vulnerable to DPA when the capacity recovery step is changed to accommodate the inclusion of the combined feedback function ρ. Thus, instead of recovering the capacity straight up, we will need to perform the inverse combined feedback function to recover the capacity and then perform the inverse permutation to uncover the key K. Table 3 summarises the difference between Beetle and Schwaemm DPA vulnerability under the CCAL1 and CIL1 security games.

4.7 | TinyJambu

TinyJambu is a family of AEAD ciphers derived from Jambu that comprises three key size options: 256 bits, 192 bits, and 128 bits. They all feature a 128-bit keyed permutation, a message block size of 32 bits, and a state size of 128 bits, as shown in Figure 4 [41].

TinyJambu constructs features of the 128-bit keyed permutation P_n at every step of its operation: initialisation, associated data processing, plaintext processing, and tag generation steps. However, the number of permutation rounds, n, varies for each step. The nonlinear feedback shift register (NLFSR) and the elementary state update function (Figure 5), are executed n times for a permutation P_n. In 32-bit processors commonly used in IoT devices, 32 rounds of permutations can be implemented

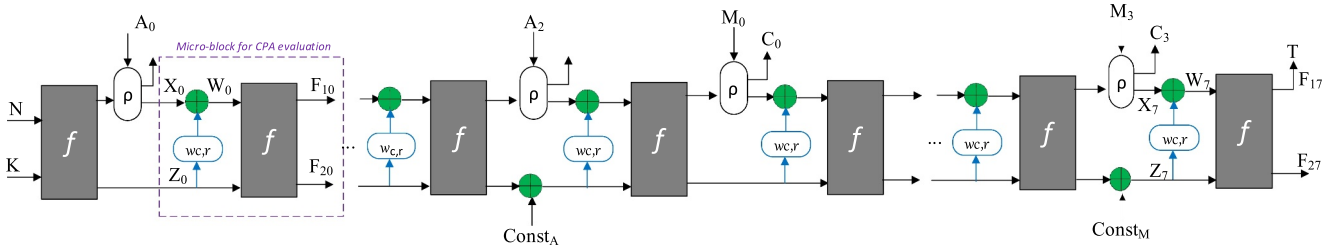

FIGURE 3 Schwaemm authenticated encryption construction with 3 associated data blocks and 4 message blocks, showing the addition of the whitening block versus Beetle [40].

TABLE 3 Difference between Beetle and Schwaemm scenarios to uncover DPA vulnerability.

Attack steps	Beetle [45]	Schwaemm
Step 1	Define a fix nonce and ephemeral key K*	Same as beetle
Step 2	Generate multiple plaintexts blocks M_1 or multiple ciphertexts C_1	Same as beetle
Step 3	Uncover the capacity section along with the plaintext M_1 or ciphertext C_1,	Perform inverse feedback function, then uncover the capacity section along with the plaintext M_1 or ciphertext C_1,
Step 4	Then perform the inverse permutation to uncover the key K	Same as beetle

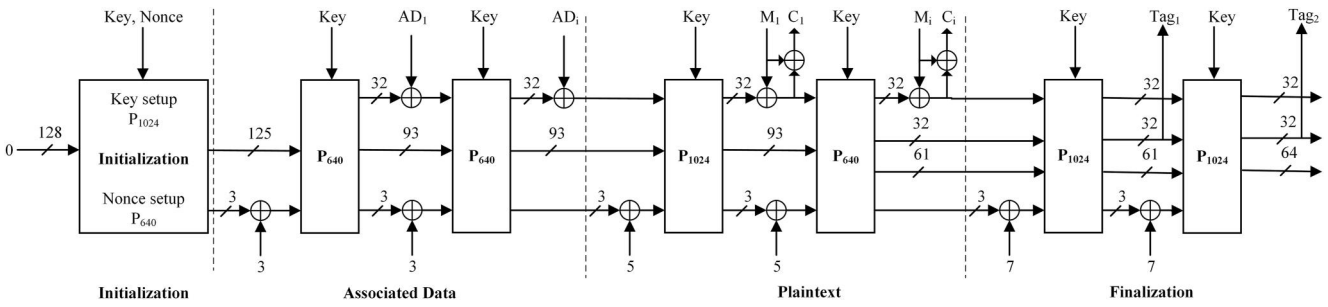

FIGURE 4 TinyJambu AEAD cipher, indicating the number of rounds of each permutation [41].

$$StateUpdate(S, K, i):$$
$$feedback = s_0 \oplus s_{47} \oplus \left(\sim(s_{70} \& s_{85})\right) \oplus s_{91} \oplus k_{i \bmod klen}$$
$$For\ j\ from\ 0\ to\ 126: s_j = s_{j+1}$$
$$s_{127} = feedback$$
$$end$$

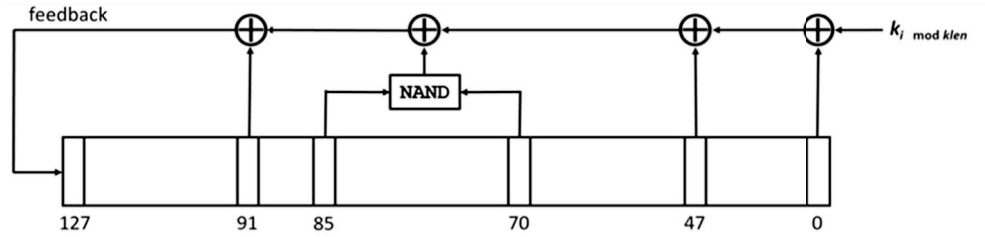

FIGURE 5 TinyJambu keyed permutation algorithm (top), and graphical feedback implementation, with the nonlinear feedback shift register (bottom) [41].

in parallel. Additionally, in a typical HW implementation, the key, nonce, and associated data are input on a 32-bit bus width to match the algorithm block size.

Scheme for attacking TinyJambu

Thus, implementations of unprotected TinyJambu with a block size of 32 bits [60] require the key to be accepted during the initialisation phase in at least 4 words of 32 bits each maximum. The process of accepting and storing the 32-bit data and then implementing parallel computations of 32 feedback bits with the NLFSR will generate power consumption that a malicious agent can exploit to run CPA. In the worst case, the key search space is reduced to a complexity of 4×2^{32}, meaning 17 billion key guesses are needed to fully uncover all 128 bits of the encryption key. With such a reduction, a traditional CPA can be carried out (with modern computers), analogous to the key search space reduction from 2^{128} to 16×2^8 of attacks on AES128 implementations [61]. Whether the computational complexity of such computations will result in a timely uncovering of the secret key is left to the next steps of this work. For a case of high-frequency implementation where the NLFSR computes 1 feedback bit per clock cycle, the algorithm implementation will additionally be vulnerable to SPA. If only one feedback bit is computed by the NLSFR in each clock cycle, the power consumption of the device will be different whether the feedback bit computed in Figure 5 results in a 1 or a 0. The computation result of the most significant bit (MSB) will then create a discernible power consumption difference that can be visually analysed by the malicious agent. Such SPA weakness, which borrows similarities to the conditional jump weakness in a data encryption standard (DES) algorithm and demonstrated in [62], allows the malicious agent to uncover the full state one bit at a time. Then, the full key can be deducted with the reverse computation of the initialisation steps.

In a nutshell, TinyJambu implementations, and particularly its initialisation phase, are vulnerable to CPA with key search space reduced from 2^{128} to 4×2^{32} and/or bit-by-bit simple power analysis attacks on its state when the feedback is computed one bit at a time. The above vulnerabilities are ubiquitous because the algorithm construct does not integrate any SCA countermeasure, such as masking or hiding. This thus makes it susceptible to leaking information that can be easily analysed with first-order DPA to uncover secret information. In fact, [60] shows an unprotected implementation of TinyJambu, on which an experiment with 10,000 traces yielded a test vector leakage assessment (TVLA) higher than 5. This is above the threshold of 4.5 widely accepted as the limit to which an implementation said is considered secure. This indicates that the implementation of an insecure TinyJambu leaks identifiable information with a probability greater than 99.999%.

5 | SUMMARY OF POWER SIDE CHANNEL ATTACKS VULNERABILITIES

The practical assessment of power SCA vulnerabilities that must be considered in evaluating the security of the seven lightweight ciphers is summarised in Table 4. For each cipher, we have shown the proposed integrated SCA countermeasure and our assessment of the residual vulnerabilities a malicious agent could exploit to extract secret information from the device. Most information is supported by prior art demonstrated with proven practical experiments, while others are novel concepts developed and demonstrated theoretically based on well-known general art concepts on power side channel cryptanalysis.

Ascon, Sparkle, and PHOTON-Beetle security vulnerability can generally be assessed with the security assumptions CCAmL1 and CCAL1/CIL1, respectively. However, the security vulnerability of GIFT-COFB, Grain, Romulus, and TinyJambu can be evaluated more straightforwardly with proposed leakage functions or publicly available leakage models (Hamming distance model). These latter four ciphers' security vulnerability can also be evaluated with a solver (satisfiability modulo theory) or with a more computer-intensive approach that consists of significantly increasing the number of traces collected to launch the attack.

TABLE 4 Residual vulnerability assessment of LWC finalist candidates.

Ciphers	Residual vulnerabilities
Ascon Type: Block cipher Key size: 128	Round reduced (7 out of 12) implementations are vulnerable to attacks [33]. Ascon is not considered secure under the CCAmL1 security game. Without DPA-protected implementation of the verification phase [45], it is possible to successfully attack secure bootloading applications [49] by estimating valid messages without knowledge of the encryption key [50]. The attack on the state can be reduced to 64-bit operations, therefore reducing the search from 2^{320} to 5×2^{64}.
GIFT-COFB Type: Block cipher (GIFT-128) Key size: 128	GIFT-COFB is vulnerable to CPA on a reduced number of rounds (11 vs. 40). However, the authors claim that a 40 round implementation is resistant to DPA [51]. GIFT-128 looks like a larger version of PRESENT, thus vulnerabilities of PRESENT can be present here as well. GIFT S-box is susceptible to CPA when assessed with the transparency order (TO) metric [59].
Grain-128AEADv2 AEAD stream cipher Key size:128	Grain-128AEADv2 is vulnerable to a power SCA with the hamming distance model [55]. It has been demonstrated that one can construct a fast and automated process through Z3, a publicly available satisfiability modulo theory (SMT) solver, with the leakage model and the publicly available keystream, that lead to key recovery in a few seconds [55]. Grain128AEADv2 is not resistant to fault attacks. The authors expect the users to implement protection mechanisms.
PHOTON-beetle authenticated encryption and hash family Key size: 128	The message processing section shows residual vulnerability against DPA under CCAL1 and CIL1 [45]. Targeting S-box with an increased number of traces may lead to the recovery of secret information.
Romulus Type: Tweakable block ciper (SKINNY) Key size: 128	Romulus is vulnerable to the same CPA as SKINNY, with leak a function defined as the hamming distance of the input/output of the target round. A power SCA mounted on an HW implementation of SKINNY with a hamming distance model showed a success rate of close to 100% with approximately 60 traces only [32].
SPARKLE (SCHWAEMM and ESCH Type: Block cipher Key size: 128	Sparkle and beetle share similar residual vulnerabilities to a power SCA. Sparkle is vulnerable to a power SCA under the security game CCAL1 and CIL1. The main difference lies in the fact that instead of recovering the capacity straight up as with beetle [45], we need to perform the inverse combined feedback function to recover the capacity, then perform the inverse permutation to uncover the key K, with schwaemm.
TinyJambu Type: Block cipher Key sizes: 128, 192, 256	An unprotected implementation of TinyJambu yielded a TVLA higher than 5, which is above the threshold of 4.5 [60]. TinyJambu implementations, and particularly its initialisation phase, are vulnerable to CPA with key search space reduced from 2^{128} to 4×2^{32}, and/or bit by bit simple power analysis attacks on its state when the feedback is computed one bit at a time.

We can further note that the ISAP, Elephant, and Xoodyak modes of operation provide built-in approaches to preventing side channel attacks against algorithm implementations. For instance, one of the most powerful tools used in power SCA, DPA, operates by accumulating information on the secret key by measuring the power consumption of the device during multiple encryption operations on different data. To counter this, ISAP has integrated a sponge-based rekeying in the encryption and MAC parts, which generates a fresh key for each new input. Doing so significantly decreases the vulnerability of ISAP implementations against a power SCA [37]. This is demonstrated in [45], where it is shown that this out-of-the-box security meets the highest security level defined by the authors, which is CCAmL2.

6 | CONCLUSION

Power side channel attacks are of great concern on IoT devices because malicious agents have physical access to the device and thus can run cryptanalysis algorithms after the products are deployed. The finalists selected by NIST at the LWC competition each have their residual vulnerabilities, of which we brought to light a few relevant ones. The expectation is that this comprehensive study will be useful to SCA vulnerability testers/analysers. Furthermore, these finalist ciphers or related variants have been previously proposed and used in applications. Therefore, regardless of the outcome of the LWC competition final selection, future IoT IC designers can leverage this work to evaluate the resilience of their products during the design phase.

AUTHOR CONTRIBUTION

Aurelien T. Mozipo: Conceptualisation, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing. **John M. Acken**: Conceptualisation, Methodology, Supervision, Visualisation, Writing – review & editing.

ACKNOWLEDGEMENTS

No funding institution.

CONFLICT OF INTEREST STATEMENT

No conflict of interest.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

ORCID

Aurelien T. Mozipo https://orcid.org/0000-0002-2740-8036

REFERENCES

1. Singh, A., et al.: Energy efficient and side-channel secure cryptographic hardware for IoT-edge nodes. IEEE Internet Things J. 6(1), 421–434 (2018)
2. Singh, A., et al.: Energy efficient and side-channel secure hardware architecture for lightweight cipher SIMON. In: 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 159–162. IEEE (2018)
3. Takemoto, S., Nozaki, Y., Yoshikawa, M.: Statistical power analysis for IoT device oriented encryption with glitch canceller. In: 2019 IEEE 11th International Workshop on Computational Intelligence and Applications (IWCIA). IEEE (2019)
4. NIST. Computer Security Research Center. https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf
5. Zhao, M., Suh, G.E.: FPGA-based remote power side-channel attacks. In: 2018 IEEE Symposium on Security and Privacy (SP). IEEE (2018)
6. Carlier, V., et al.: Electromagnetic side channels of an FPGA implementation of AES, REPORT 2004/145. In: CRYPTOLOGY EPRINT ARCHIVE (2004)
7. Kison, C., Frinken, J., Paar, C.: Finding the AES bits in the haystack: reverse engineering and sca using voltage contrast. In: International Workshop on Cryptographic Hardware and Embedded Systems, pp. 641–660. Springer, Berlin (2015)
8. Van der Veen, V., et al.: Drammer deterministic rowhammer attacks on mobile platforms. In: Proc. Conf. Comput. Commun. Security, pp. 1657–1689. CCS), Vienna (2016)
9. Yaron, Y., Falkner, K., Flush-Reload: A high resolution, low noise, L3 cache side channel attack. In: Proc: USENIX Security Symp, pp. 719–732. San Diego (2014)
10. O'Flynn, C.: Fault injection using crowbars on embeded systems, Report 2016/810, 2016. [Online]. Available:. In: IACR Cryptology ePrint Archive. https://eprint.iacr.org/2016/810
11. Khan, M.N., Rao, A., Camtepe, S.: Lightweight cryptographic protocols for IoT-constrained devices: a survey. In: IEEE Internet of Things Journal, vol. 8(6), pp. 4132–4156 (2021). https://doi.org/10.1109/JIOT.2020.3026493
12. Fotovvat, A., et al.: Comparative performance analysis of lightweight cryptography algorithms for IoT sensor nodes. IEEE Internet Things J. 8(10), 8279–8290 (2021). https://doi.org/10.1109/JIOT.2020.3044526
13. Philip, M.A., Vaithiyanathan: A survey on lightweight ciphers for IoT devices. In: 2017 International Conference on Technological Advancements in Power and Energy, pp. 1–4. TAP Energy) (2017). https://doi.org/10.1109/TAPENERGY.2017.8397271
14. Spreitzer, R., et al.: Systematic classification of side-Channel Attacks: a case study for mobile devices. IEEE Communications Surveys & Tutorials 20(1), 465–488 (2018). https://doi.org/10.1109/COMST.2017.2779824
15. Granjal, J., Monteiro, E., Silva, J.Sá: Security for the internet of things: a survey of existing protocols and open research issues. IEEE Communications Surveys & Tutorials 17(3), 1294–1312 (2015). https://doi.org/10.1109/COMST.2015.2388550
16. Yang, D., Qi, W.F., Chen, H.J.: Impossible differential attacks on the SKINNY family of block ciphers. IET Inf. Secur. 11(6), 377–85 (2017). https://doi.org/10.1049/iet-ifs.2016.0488
17. Liu, S., Guan, J., Hu, B.: Fault attacks on authenticated encryption modes for GIFT. IET Inf. Secur. 16(1), 51–63 (2022). https://doi.org/10.1049/ise2.12041
18. Randolph, M., Diehl, W.: Power side-channel attack analysis: a review of 20 years of study for the layman. Cryptography 4, 15–2 (2020). https://doi.org/10.3390/cryptography4020015
19. Mangard, S., Oswald, E., Standaert, F.X.: One for all–all for one: unifying standard differential power analysis attacks. IET Inf. Secur. 5(2), 100–10 (2011). https://doi.org/10.1049/iet-ifs.2010.0096
20. Fei, Y., et al.: Correlation power analysis and higher-order masking implementation of WAGE. In: Selected Areas in Cryptography: 27th International Conference, Halifax, NS, Canada (Virtual Event), pp. 593–614. Springer International Publishing (2020). Revised Selected Papers 27 2021
21. Liu, Z., Schaumont, P.: Root-Cause Analysis of Power-Based Side-Channel Leakage in Lightweight Cryptography Candidates. NIST 5th Lightweight Cryptography Workshop (2022). https://csrc.nist.rip/presentations/2022/root-cause-analysis-of-power-based-side-channel-le
22. Abduladir, A., et al.: Side-Channel Resistant Implementation of Three Finalists of the NIST Lightweight Cryptography Standardization Process: Elephant, TinyJambu, and Xoodyak". NIST 5th Lightweight Cryptography Workshop (2022). https://csrc.nist.rip/presentations/2022/side-channel-resistant-implementations-of-three-lw
23. Unger, W., et al., TVLA: Correlation Power Analysis and Side-Channel Leakage Assessment Metrics". NIST 5th Lightweight Cryptography Workshop (2022). https://csrc.nist.rip/presentations/2022/tvla-correlation-power-analysis-side-channel-leaka
24. Nalla Anandakumar, N.: SCA resistance analysis on FPGA implementations of sponge based MAC−PHOTON. In: Bica, I., Naccache, D., Simion, E. (eds.) Innovative Security Solutions for Information Technology and Communications. SECITC 2015. Lecture Notes in Computer Science, vol. 9522. Cham (2015). https://doi.org/10.1007/978-3-319-27179-8_6
25. Biryukov, A., Dinu, D., Großschädl, J.: Correlation power analysis of lightweight block ciphers: from theory to practice. In: Manulis, M., Sadeghi, A.R., Schneider, S. (eds.) Applied Cryptography and Network Security. ACNS 2016. Lecture Notes in Computer Science, vol. 9696. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39555-5_29
26. Zhang, J., et al.: Power analysis attack on a lightweight block cipher GIFT. In: Liu, Q., et al. (eds.) Proceedings of the 9th International Conference on Computer Engineering and Networks. Advances in Intelligent Systems and Computing, vol. 1143. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-3753-0_55
27. Samwel, N., Daemen, J.: DPA on hardware implementations of Ascon and Keyak". In: Proceedings of the Computing Frontiers Conference, pp. 415–424 (2017). https://doi.org/10.1145/3075564.3079067
28. Windarta, S., et al.: Lightweight cryptographic hash functions: design trends, comparative study, and future directions. IEEE Access 10, 82272–82294 (2022). https://doi.org/10.1109/ACCESS.2022.3195572
29. Batina, L., et al.: Side-Channel Evaluation Report on Implementations of Several NIST LWC Finalists". https://repository.ubn.ru.nl/handle/2066/253567
30. Miteloudi, K., et al.: Evaluating the ROCKY countermeasure for side-channel leakage. In: 2021 IFIP/IEEE 29th International Conference on Very Large Scale Integration (VLSI-SoC), pp. 1–6. Singapore (2021). https://doi.org/10.1109/VLSI-SoC53125.2021.9606973
31. Diehl, W., et al.: Comparison of cost of protection against differential power analysis of selected authenticated ciphers. In: 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 147–152. Washington (2018). https://doi.org/10.1109/HST.2018.8383904
32. Heuser, A., et al.: Lightweight ciphers and their side-channel resilience. IEEE Trans. Comput. 69(10), 1434–1448 (2017). https://doi.org/10.1109/tc.2017.2757921
33. Dobraunig, C., et al.: Ascon". https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/ascon-spec-final.pdf
34. Beyne, T., et al.: Elephant v2" https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/elephant-spec-final.pdf

35. Subhadeep, S., et al.: GIFT-COFB". https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/gift-cofb-spec-final.pdf
36. Hell, M., et al.: Grain-128AEADv2 – A Lightweight AEAD Stream Cipher". https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/grain-128aead-spec-final.pdf
37. Dobraunig, C., et al.: ISAP v2.0". https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/isap-spec-final.pdf
38. Bao, Z., et al.: PHOTON-beetle Authenticated Encryption and Hash Family". https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/photon-beetle-spec-final.pdf
39. Guo, C., et al.: Romulus, v1.3". https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/romulus-spec-final.pdf
40. Beierle, C., et al.: Schwaemm and Esch: Lightweight Authenticated Encryption and Hashing Using the Sparkle Permutation Family". https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/sparkle-spec-final.pdf
41. Wu, H., Huang, T., TinyJambu: A Family of Lightweight Authenticated Encryption Algorithms. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/tinyjambu-spec-final.pdf
42. Daemen, J., et al.: Xoodyak, a lightweight cryptographic scheme. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/xoodyak-spec-final.pdf
43. Mozipo, A.T., Acken, J.M.: Power Side Channel attack of AES FPGA implementation with experimental results using full keys. In: 2021 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), pp. 1–6 (2021). https://doi.org/10.1109/DTS52014.2021.9497976
44. Lightweight Cryptography in Hardware and Embedded Systems. Evaluation of Finalists in the NIST LWC Process." https://cryptography.gmu.edu/athena/index.php?id=LWC
45. Bellizia, D., et al.: Mode-level vs. implementation-level physical security in symmetric cryptography. In: Annual International Cryptology Conference, pp. 369–400. Springer, Cham (2020)
46. Nguyen, N.D., Bui, D.H., Tran, X.T.: A Lightweight AEAD encryption core to secure IoT applications. In: 2020 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), pp. 35–38. IEEE (2020)
47. Rostampour, S., et al.: An authentication protocol for next generation of constrained Iot systems. IEEE Internet Things J. 9(21), 21493–504 (2022). https://doi.org/10.1109/jiot.2022.3184293
48. De Santis, F., Schauer, A., Sigl, G.: ChaCha20-Poly1305 authenticated encryption for high-speed embedded IoT applications, In Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 692–697. IEEE (2017)
49. O'Flynn, C., Chen, Z.D.: Side channel power analysis of an AES-256 bootloader. In: CCECE, pp. 750–755. IEEE (2015)
50. Berti, F., et al.: On leakage-resilient authenticated encryption with decryption leakages. IACR Trans. Symmetric Cryptol(3), 271–293 (2017). https://doi.org/10.46586/tosc.v2017.i3.271-293
51. Banik, S., et al.: GIFT: a small present. In: International Conference on Cryptographic Hardware and Embedded Systems, pp. 321–345. Springer, Cham (2017)
52. Slpsk, P., et al.: Karna: a gate-sizing based security aware EDA flow for improved power side-channel attack protection. In: 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 1–8. IEEE (2019)
53. Zhang, B., Gong, X., Meier, W.: Fast correlation attacks on Grain-like small state stream ciphers. IACR Transactions on Symmetric Cryptology, 58–81 (2017). https://doi.org/10.46586/tosc.v2017.i4.58-81
54. Todo, Y., et al.: Fast correlation attack revisited. In: Annual International Cryptology Conference, pp. 129–159. Springer, Cham (2018)
55. Baksi, A., Kumar, S., Sarkar, S.: A new approach for Side Channel analysis on stream ciphers and related constructions. IEEE Trans. Comput. 71(10), 2527–2537 (2021). https://doi.org/10.1109/tc.2021.3135191
56. Li, H., et al.: Transparency order versus confusion coefficient: a case study of NIST lightweight cryptography S-Boxes. Cybersecur 4(1), 35 (2021). https://doi.org/10.1186/s42400-021-00099-1
57. Ge, J., et al.: Power attack and protected implementation on lightweight block cipher SKINNY. In: 2018 13th Asia Joint Conference on Information Security (AsiaJCIS), pp. 69–74. IEEE (2018)
58. Chakraborti, A., et al.: Beetle Family of Lightweight and Secure Authenticated Encryption Ciphers. Cryptology ePrint Archive (2018)
59. Unger, W., et al.: Side-channel leakage assessment metrics: a case study of GIFT block ciphers. In: 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 236–241 (2021). https://doi.org/10.1109/ISVLSI51109.2021.00051
60. Abdulgadir, A., et al.: Side-channel resistant implementations of a novel lightweight authenticated cipher with application to hardware security. In: Proceedings of the 2021 on Great Lakes Symposium on VLSI, pp. 229–234 (2021)
61. NewAE Technology Inc. ChipWhisperer® by https://wiki.newae.com/Main_Page
62. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Annual International Cryptology Conference. Springer, Berlin (1999)

How to cite this article: Mozipo, A.T., Acken, J.M.: Residual vulnerabilities to power side channel attacks of lightweight ciphers cryptography competition finalists. IET Comput. Digit. Tech. 1–14 (2023). https://doi.org/10.1049/cdt2.12057