Evaluation of SPN-Based Lightweight Crypto-Ciphers

Loïc Dalmasso, Florent Bruguier, Pascal Benoit, Lionel Torres

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 20, 2018, accepted December 6, 2018, date of publication January 14, 2019, date of current version January 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2889790

Evaluation of SPN-Based Lightweight Crypto-Ciphers

LOÏC DALMASSO, FLORENT BRUGUIER, PASCAL BENOIT, AND LIONEL TORRES
LIRMM, CNRS, University of Montpellier, 34095 Montpellier, France

Corresponding author: Loïc Dalmasso (firstname.name@lirmm.fr)

This work was supported in part by the French Agence Nationale de la Recherche (ANR) through the Project IDEFI-FINMINA under Grant ANR-11-IDFI-0017, in part by the European Union ERDF Funds (European Regional Development Fund), and in part by the Occitanie Region Funds through the Project GEPETOS VASCO/SECNUM under Grant FEDER FSE IEJ 2016-006259.

ABSTRACT Lightweight cryptography has recently emerged as a strong requirement for any highly constrained connected device; encryption/decryption processes must strike the balance between speed, area, power efficiency, and security robustness. The aim of this paper is to study the potential gains of the lightweight cryptography algorithms compared to the classic ones in hardware implementation. Advanced Encryption Standard (AES) as the standard, PRESENT and the very recently published GIFT are considered along with several optimized hardware versions of each one. Low- and high-security levels with 80- and 128-bit key length respectively are compared. They are all implemented on a Xilinx Kintex-7 FPGA, exploiting different slice configurations to evaluate their performances. The results show the expected benefits in terms of throughput and area, which allows selecting the best lightweight crypto-ciphers depending on the target device or application. In addition, correlation power analysis is performed on each cipher to estimate their resistance against side-channel analysis.

INDEX TERMS Lightweight cryptography, block cipher, substitution-permutation-network, advanced encryption standard (AES), PRESENT, GIFT.

I. INTRODUCTION

As the number of connected devices increases exponentially within the so-called "Internet of Things" (IoT), experts agree about the huge potential of cybersecurity threats they open. Cryptography offers efficient means to address confidentiality, integrity and authenticity issues of devices and communications. Although the Advanced Encryption Standard (AES) is the standard of block-based symmetric ciphering, it may not be suitable for very constrained systems, especially for end-nodes like simple connected sensors, mainly because of the processing, area and energy overheads it requires. To ensure resourceful security, several block ciphers have been published during the last decade. Among them, LED [1], MIDORI [2] and SKINNY [3] use an AES-like structure. PRESENT [4] and GIFT [5] are based on a Substitution-Permutation Network (SPN), as AES, but they differ in the way that they are bit oriented, opening new opportunities for hardware optimizations.

As of today, PRESENT is the reference of lightweight symmetric ciphering, and is included into ISO standards [6]. Several works have been done on PRESENT to optimize it regarding the area on FPGA using its different architectural features [7] or deriving Boolean S-Boxes [8] with ESPRESSO algorithm [9]. The smallest implementation is achieved in [10], reducing the total of S-Boxes by an 8-bit data path and by the use of a Karnaugh mapping. Furthermore, only two simple counters are used for the logic control, minimizing the area. Another small implementation of PRESENT is based on RAM blocks [11]. This approach provides a smaller slice count and can be performed into two ways: S-Boxes are implemented in slices or stored in RAM, as a lookup table. The second method uses a smaller slice count but increases the number of cycles and the complexity of the control logic.

Recently, Banik *et al.* [5] have proposed GIFT, claiming that it is the smallest lightweight cipher. It also corrects the well-known weaknesses of PRESENT, like the linear approximation [12]. Since GIFT is derived from PRESENT, which shares common properties with AES, it is relevant to compare them, with the same area-oriented implementation on FPGA technology, highlighting the throughput-to-area and the energy-per-bit characteristics.

Even if the lightweight block cipher is robust to classical cryptanalysis [4], [5], another important part of their

security level, is about their robustness against side-channel attacks. This type of attack, as CPA, analyzes physical leakage, emitted during cryptographic operation in a device and involving the secret key. This gain of secret information is only due to the implementation of the cryptographic algorithm. With a constant development of side-channel attacks and the growth of connected devices which interact more and more with data, this type of attack is a huge threat. Nowadays, side channel attacks are the most dangerous threats, since they require few knowledge and equipment. While the countermeasure methods [13]–[15] are the same for both classic and lightweight cryptography, new attacks have appeared for the latter, as [16] and [17] exploiting the bit permutation, the common diffusion function in lightweight block ciphers. Note that [16] and [17] are experimented on software implementations.

The main objective of this paper, is, based on a set of metrics, to fairly compare lightweight cryptography to the classic ones in hardware implementations. The ambition of this paper is also to evaluate their resistance against SCA, especially the common CPA. Of course in the literature many papers address lightweight cryptography algorithm and implementation [18], [19], but comparisons are often difficult and not based on the same technology, or same SCA methods [16], [17]. The main interest herein is to propose common criteria to make this evaluation. This study can be very useful, and new, for designers or embedded system architects, who want to select a given block cipher, to provide security on their very constrained devices. Low and high security levels can be balanced with 80 and 128-bit key length. Several optimized hardware implementations, using different slice configurations, are proposed to evaluate the performances in terms of throughput, area and power-consumption. To the best of our knowledge, it is also the very first one evaluating and comparing GIFT to its counterparts.

The remainder of this contribution is organized as follows: Section 2 summarizes the main aspects of PRESENT and GIFT algorithms. Their implementations are given in Section 3, followed by the comparison results in Section 4. Section 5 focuses on side-channel analysis, highlighting the setup of CPA attack on these algorithms. Results of attack are given in Section 6. Finally, concluding remarks are provided in Section 7.

II. ALGORITHMS

To propose optimized implementations and analyze the results, this background section focuses on the specifications and main properties of PRESENT and GIFT algorithms.

A. PRESENT

PRESENT [4] is an SPN based algorithm with 128- or 80-bit key and runs in 31 rounds. 80-bit version is recommended by its designers to be used in small designs, with strong area constraints. However, to provide fair comparisons, a 128-bit version has also been considered in this work, called PRESENT-128. Each round is based on three primary operations described below: AddRoundKey, Substitution and Permutation. Figure 1 shows the top-level architecture.

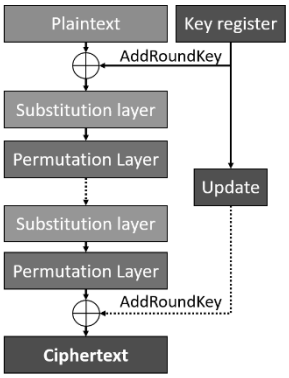

FIGURE 1. Top-level description of PRESENT algorithm.

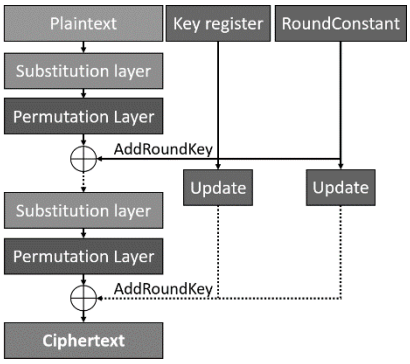

FIGURE 2. Top-level description of GIFT algorithm.

The 64-bit plaintext is first XORed with the round key. The result is then processed in the Substitution layer. This is a non-linear operation, also called "SBOX", that changes a 4-bit data in another one. The content of PRESENT SBOXes can be found in [4]. Finally, the Permutation layer remaps bits from bit position x of the input plaintext to bit position y of the output. All these steps are performed for each of 31 rounds.

During ciphering, the 64 left-most bits of the key register are used as round key. Then, the Key Schedule updates the 128-bit key register at each round. For PRESENT-80 only the Key Schedule operation is different.

B. GIFT

GIFT [5] is another SPN (relying on Substitution, Permutation layer and AddRoundKey as depicted in Figure 2) with 128-bit key and possibly 64- or 128-bit data. Compared to PRESENT, there are differences in the bit permutation layer, which depend on the data size, and the number of rounds (28-round for 64-bit and 40-round for 128-bit). These versions are called GIFT-64-128 and GIFT-128-128 respectively. In order to fairly compare PRESENT and GIFT, this paper is focused on GIFT-64-128 version.

Compared to PRESENT, it can be noticed that the XOR operation is there performed after the SBOX and Permutation layer. In addition, only 32-bit round key is mixed with 64-bit plaintext and 6-bit round constant. During ciphering,

the 32 LSBs of the key register are used as the round key. Then, the Key Schedule updates the 128-bit key register for each round.

III. IMPLEMENTATIONS

There are generally several possibilities to implement these algorithms, which allow balancing performance and area. Unrolled pipelined structures provide intrinsically the highest throughputs, but require much more resources. In the context of constrained devices, as the area is generally very limited, compact solutions are the preferred. The ''Round-based'' is a well-known implementation, where an entire round of the cipher is computed at every clock cycle. Only one round is implemented and re-used for other rounds (in a loop sequential fashion): AES (ECB mode), PRESENT and GIFT were all implemented in this way, encryption-only. Table 1 summarizes the block cipher parameters used in this study.

TABLE 1. Block cipher parameters.

Cipher	Key size	Data size	Rounds
AES-128	128	128	10
PRESENT-80	80	64	31
PRESENT-128	128	64	31
GIFT-64-128	128	64	28

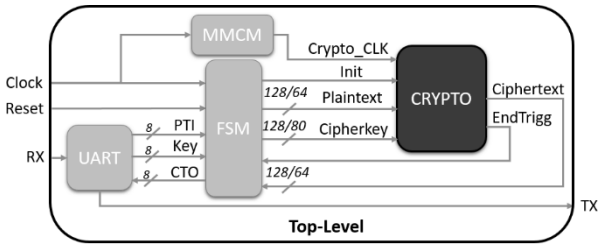

FIGURE 3. Top-level architecture.

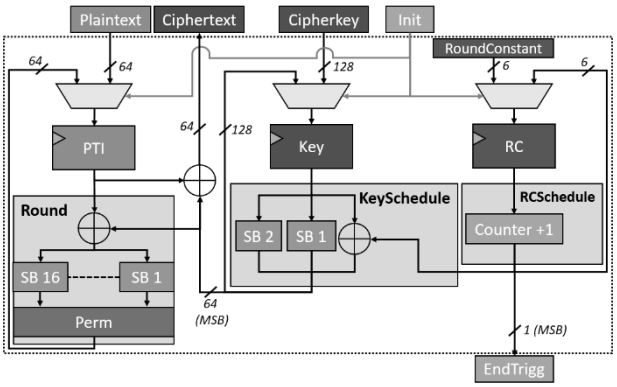

FIGURE 4. PRESENT-128 Round-Based implementation.

The top-level architecture is described in Figure 3. The plaintext and the cipher key are sent through the UART communication to the ''Crypto'' block. The cipher text is computed and transmitted to the user by the UART. A Mixed-Mode Clock Manager (MMCM) Xilinx module is used to generate the clock for the cryptography block.

The detailed architectures of PRESENT-128 and GIFT-64-128 are described in Figure 4 and 5 respectively.

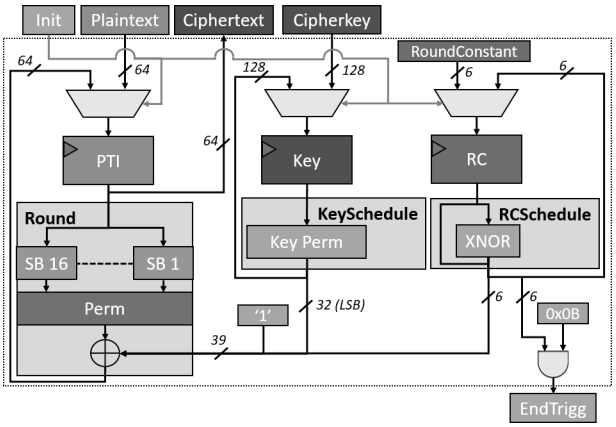

FIGURE 5. GIFT-64-128 Round-Based implementation.

The *Round Constant* input is fixed into the crypto block. The existing implementations generally wait for the end of a transmission of the plaintext and the key, to start the ciphering process, which implies one extra latency cycle to the whole process. Our implementation uses the *Init* input (Figure. 3) as a starting trigger. While the UART is transmitting the data, the *Init* signal, through the multiplexers, selects the plaintext, the cipher key and the round constant inputs, that are written in their dedicated registers (PTI, KEY and RC). The FSM maintains the crypto block into the first round state until the end of the transmission. Once done, the multiplexers, using the *Init* signal, select the feedback loop input (Figure. 4 and 5), which already processed the first round with the available data. Consequently, it allows saving one extra cycle to the total number of rounds for every block to cipher.

Each component of the Crypto block (*Round*, *KeySchedule* and *RCSchedule*) is implemented in a combinatory way and saves its result in a 64-, 128- and 6-bit register respectively, which is used as an input at the next iteration. Note that these are the same registers as in the initialization phase.

The *Round* contains the combinatory functions: the XOR operation, the substitution (*SBx*) and the permutation layers (*Perm*). In PRESENT, the plaintext is first XORed with the round key, contrary to GIFT, which performs this at the end of the round. Each SBOX is implemented with LUTs as read only memories (''ROM''), and 64-bit permutation is performed with wires only.

The round key is updated at each round by the *KeySchedule*. In PRESENT, the key is the 64 left-most bits of the key register. The new key is generated by two SBOXes at the MSB position and the *Round Constant*. In GIFT, the round key is the 32 LSB bits of the key register and only a 128-bit permutation, using wires, is needed to update the key.

The *RCSchedule* generates the *Round Constant*. In the proposed architectures, this block is also used as a simplified FSM. In PRESENT, this is a 5-bit counter, representing the current round number, from 1 to 31 (''11111'' in binary format). Adding one more bit, the new *Round Constant* on 6 bits allows to detect the end of the ciphering by triggering

on the MSB bit. In the same way, the end of GIFT is triggered on the last 6-bit *Round Constant* value (0x0B in hexadecimal format). No more bit is needed but one extra LUT is used.

In GIFT, the cipher text is available in PTI register at the last round. In PRESENT, another XOR operation is performed between the PTI register value and the last round key to generate the cipher text. When the ciphering is done, the trigger of end signal (*Endtrigg*) is sent to the top-level (Figure 3).

In order to compare our results with recent implementations in the literature as in [19], the Xilinx Kintex-7 family was chosen. In these FPGAs, every slice contains four 6-input LUTs, eight storage elements, as FFs, wide-function multiplexers and carry logic. In addition, some slices can use distributed RAM to store data and 32-bit registers to shift data. As these properties might impact area and performance, several implementations of the block ciphers are proposed to find the best slice configuration: A) "Portable" using LUTs and FFs without specific configurations; B) "FF" using FFs and slice MUXs; C) "SRL" using LUT as distributed memory and slice MUXs. All of these implementations use the same *Round*, *KeySchedule* and *RCSchedule* blocks. Only the input selection and the storage architecture change. Table 2 describes the resource used for each implementation.

TABLE 2. Implementation resource utilization.

Design	Input MUXs	Storage blocks	Xilinx Primitive
"Portable"	LUTs	FFs	No
"FF"	Slice MUXs		Yes
"SRL"		LUTRAMs	Yes

"Portable" version can be implemented on any FPGA of any vendor, hence its name. However, "FF" and "SRL" are exclusive Xilinx versions, because they use vendor's primitives.

IV. RESULTS

All the designs were done with VHDL, synthetized for Xilinx Kintex-7 XC7K70T FPGA using "AreaOptimized_High" parameter on Xilinx Vivado 2016.13. All results are given after place and route. Area is considered on a Slice basis, as the majority of related works.

Since one of our objectives is to quantify the benefits of lightweight ciphers over standard ones, we provide an implementation of AES that is compared to other ones. The technology of the FPGA and the operating frequency may have an important impact on the area and the throughput. In order to have fair results along this paper, a normalization at 100MHz was chosen, since it is the nominal frequency of the target platform (Nexys-4 board [21]). The plaintext and key loading phases are not considered, in order to focus only on the crypto-core. Table 3 shows different designs on different platforms, their area, throughput and throughput on area ratio.

TABLE 3. AES comparison with the state of the art.

Design	Platform	Area (Slices)	TP (Mbps)	TP/A (Mbps/Slices)
[19][1]	Kintex-7	95	41.55	0.437
[6]	Virtex-5	359	1280	3.565
"Portable"	Kintex-7	429		2.984
"FF"		429		2.984
"SRL"		463		2.765
[22][2]		132	320	2.42
[23][1,3]		94	80	0.85

[1] Using 8-bit data path.
[2] Using 32-bit data path.
[3] Encryption and decryption

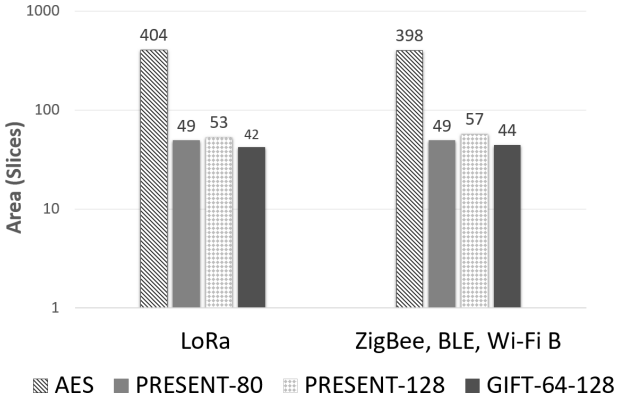

FIGURE 6. Block cipher comparison in terms of area at fixed throughput.

Since the slice structure of Xilinx 7 and 5 series is the same (four 6-inputs LUTs, multiplexers and eight FFs), the area can be fairly compared on a slice basis. The throughput (TP) is given in Mbits per second, and computed as (1).

$$TP = (Freq * Data_{Size})/Cycles \quad (1)$$

Cycles represents the number of cycles required to output one ciphered data, and *Data_size*, the size in bits of the data. *1/Freq* is the duration (in seconds) of one cycle, and generally determined with the critical path. Since the same algorithm is implemented with the same architecture in a round-based fashion, the throughputs are similar for [6], "Portable", "FF" and "SRL" implementations. Only [19] and [23], which lie on an 8-bit data path, exhibit a reduced TP compared to the other ones. Our AES implementation was not specifically optimized, which explains that the area is higher. However, the TP/A value is close to the state-of-the-art expectations, and can be used as a fair reference against PRESENT and GIFT. The energy-per-bit is computed as (2).

$$E_{bit} = (Cycles * Power)/(Freq * Data_{Size}) \quad (2)$$

Power is provided from the related Vivado report, using only the consumption of the crypto-core.

To assess the potential gains of lightweight crypto-ciphers, two approaches are taken into account: 1) Evaluate them at their maximum respective frequency; 2) Compare them at a fixed throughput (Figure 6).

TABLE 4. Comparison of block cipher according to the different architectures at maximum frequency.

Design	Freq (MHz)	Area			TP (Mbps)	TP/A (Mbps/Slices)	Energy/bit (pJ/bit)
		LUTs	FFs	Slices			
AES-128							
"Portable"	319	1587	344	429	4083.20	9.52	49.71
"FF"[1]	324	1474	344	429	4147.20	9.67	51.60
"SRL"[1]	242	1612	88	463	3097.60	6.69	50.68
PRESENT-80							
"Portable"	639	246	150	68	1319.22	19.40	40.93
"FF"[2]	741	205	150	101	1529.80	15.15	22.88
"SRL"[2]	391	279	0	130	807.22	6.21	23.54
PRESENT-128							
"Portable"	624	271	198	75	1288.26	17.18	42.69
"FF"[3]	740	210	198	123	1527.74	12.42	24.87
"SRL"[3]	400	297	0	151	825.80	5.47	27.85
GIFT-64-128							
"Portable"	636	181	198	50	1453.71	29.07	20.63
"FF"[3]	646	89	198	110	1476.57	13.42	14.22
"SRL"[3]	388	190	0	133	886.85	6.67	13.53

[1] Using 256 slice MUXs.
[2] Using 150 slice MUXs.
[3] Using 198 slice MUXs.

A. HARDWARE EVALUATION AT THE MAXIMUM FREQUENCY

As it might be expected, data in Table 4 show that AES has the best throughput, due to its largest data path and its smallest number of rounds. Nevertheless, its area size is on average 4 times larger than lightweight block ciphers, which results in a higher energy-consumption. Indeed, normalized at 100MHz, energy-per-bit of AES is twice as high as PRESENT-80/128 and about five times higher than GIFT-64-128. These results confirm that AES is definitively not suitable for very constrained devices.

PRESENT, which is the reference of lightweight symmetric ciphering, has a better performance on area trade-off in its 80-bit key version compared to the 128-bit, at the price of a lower security level. The recently published block cipher GIFT provides the best performances in terms of area, TP/A ratio, and energy-consumption, even better than the lightest version of PRESENT. For the same security level and according to the three implementations of GIFT and PRESENT-128, the energy-consumption is divided by 2 (normalized at 100MHz) and the size is reduced by 1.2 on average (in terms of slices). Since they have the same key and data length, the reduction in only due to the number of LUTs on combinatory logic. This difference mainly comes from the XOR operation, in "Round" block. Indeed, in PRESENT the 64-bit plaintext is XORed with 64-bit round key, whereas in GIFT only 39 bits ('1' at MSB position + 32-bit round key + 6-bit *round constant*) are XORed. Another reason lies in the *KeySchedule* block. PRESENT's is bigger because of the XOR operation between the key and the *round constant*. In GIFT, the *KeySchedule* is obtained by wire permutations, *i.e.* that it requires no LUT, which implies a smaller size for GIFT.

Comparing our PRESENT and GIFT implementations, "FF" version allows the best throughput, thanks to the reduction in the number of LUTs. However, since slices contain only four multiplexers, the required area is larger. In terms of area and TP/A, the "Portable" version is the best option. Note that the best TP/A, where area (A) is in slice, is not necessarily obtained with the maximal frequency, due to the place and route operations, which used more or less slices to guaranty the execution at the maximal frequency. In addition, the maximal frequency of GIFT is lower than PRESENT because the critical path is due to an extra LUT needed to trigger the end of an encryption. This trigger is mapped in PRESENT on the MSB bit of *round constant*, which requires no additional logic.

B. HARDWARE EVALUATION AT THE FIXED THROUGHPUT

One of the biggest motivations behind lightweight cryptography is to address the security issues in very constrained embedded devices, such as IoT end nodes. Generally, they use wireless communication protocols such as LoRa, ZigBee, BLE or Wi-Fi. The chosen radio generally limits the throughput: for this reason, it is interesting to assess the different lightweight crypto cores at fixed throughputs depending on the chosen protocol. The results in Figures 6 and 7 compare our best implementations ("Portable") at four different rates. For each protocol, the frequency has been chosen in order to cope with the maximum data throughput. Table 5 provides details of area comparison.

The above histograms clearly show the benefits of the lightweight crypto-ciphers against AES, in terms of both area and TP/area trade-off, around one order of magnitude (a 9 to 10 factor). Focusing on the two lightweight crypto cores, GIFT is the best one in all scenarios. It is about 20% smaller than PRESENT in terms of area and TP/A ratio. It can be noticed that the power-consumption was estimated around 1mW for all implementations, but since it is mainly due to idle dynamic power and static consumption,

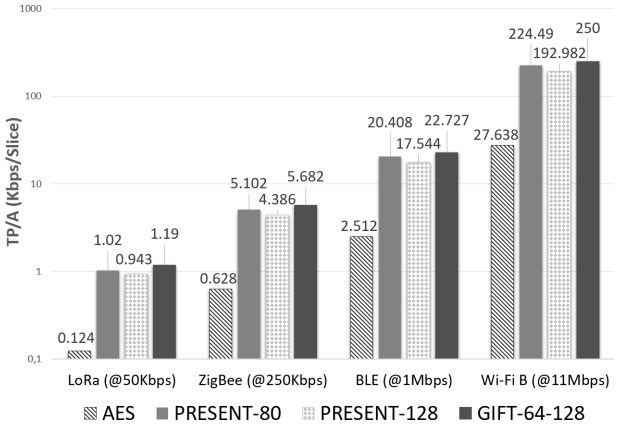

FIGURE 7. Block cipher comparison in terms of TP/A at fixed throughput.

TABLE 5. Details of block cipher comparison in terms of area at fixed throughput.

Protocols	Area		
	LUTs	FFs	Slices
AES-128			
LoRa	1454	344	404
ZigBee / BLE / WIFI	1457	344	398
PRESENT-80			
LoRa	188	150	49
ZigBee / BLE / WIFI	191	150	49
PRESENT-128			
LoRa	176	198	53
ZigBee / BLE / WIFI	214	198	57
GIFT-64-128			
LoRa	158	198	42
ZigBee / BLE / Wi-Fi	158	198	44

Note: AES is little smaller at 250 Kbps than 50 Kbps in number of slices. However, it is bigger in terms of LUTs. This is due to the place and route Vivado software, which works as a black box

this cannot be used as a discriminatory metric on this FPGA technology.

C. COMPARISON OF HARDWARE VERSIONS WITH THE STATE OF THE ART

To position this work to the related works, Table 6 compares the proposed designs of PRESENT and GIFT against the implementations in the literature. As justified previously, results are normalized at 100MHz and the loading phases of plaintext and key are not included.

According to Table 6, our PRESENT implementations are in the same range of performance. In addition, it can be noticed that the ''Portable'' version of PRESENT-80 reaches the best TP/A ratio thanks to its smaller *KeySchedule* block, compared to others implementations of PRESENT.

V. SIDE-CHANNEL ANALYSIS

Previously, results have shown that GIFT-64-128 has the best performance in terms of area, TP and TP/A compared to others. To complete this study, we focus on the security evaluation of the previous implementations of AES, PRESENT and GIFT. To have a fair comparison, this security evaluation focuses on CPA [24], a common attack for both classic and lightweight cryptography, using electromagnetic emanations. This work does not claim to introduce a new attack, but will serve as a metric to compare robustness.

A. EXPERIMENTAL PROTOCOL

All ciphers are implemented on Nexys-4 board [21] with Xilinx Artix-7 FPGA, running at 100MHz. In order to improve signal-to-noise ratio, the crypto-core is placed separately from the rest of the architecture implementation (described in Figure 3). Electromagnetic (EM) waves are captured by an EM probe and converted in electric signal, received by a 60dB Low-Noise Amplifier (LNA) to increase the signal level without degrading the signal-to-noise ratio. An oscilloscope (3.5GHz LeCroy) is used to plot the signal in order to find the best leakage point on the FPGA, where the EM radiations are the strongest.

The CPA is performed on the last round of algorithms, using the Hamming Distance (HD) model. The aim of this evaluation is to quantify the number of traces needed to find the correct key, with the Success Rate metric. It corresponds to the percentage of the correct key, that the attacker has found during the attack process, according to the number of traces used.

To have a fair comparison, our attack recovers 8-bit of the key at a time, for each cipher algorithm. Figure 8 shows the CPA attack principle at the last round.

The aim of the attack is to compute the Hamming Distance of the *Input* and *Output* of SBOX. The *Input* is generated using the reverse operation of the permutation layer and the SBOX, with hypothesis key values. *Output* is the ciphertext. The number of attack round is determined by analyzing the keyschedule of each cipher. For AES, the whole 128-bit key is used at a time. Only one round has to be attacked to recover the cipherkey. Figure 9, 10 and 11 describe the utilization of the key in PRESENT-80, PRESENT-128 and GIFT-64-128 respectively.

Both version of PRESENT use 64-bit key at each round. Its keyschedule operation is mainly a 61-bit shift left rotation. For -80 version, 2 roundkeys (64 bits + 16 bits) are needed to recover the cipherkey and 3 roundkeys (64 bits + 61 bits + 3 bits) for -128 version.

In case of GIFT, each round uses 32-bit key. Its keyschedule is a 32-bit rotation, so the cipherkey can be recovered with 4 roundkeys ($4 * 32$ bits).

B. CPA RESULTS

Figure 12 shows the Success Rate of CPA attacks.

According to these results, PRESENT is the most resistant against CPA, in its both versions (-128 is better about 1.1 times than -80). AES is in the third place and its resistant is divided by almost 3 compared to PRESENT-128. Finally, GIFT is the most vulnerable by almost a 4-factor than PRESENT-128.

In order to explain these results, Figure 13, 14 and 15 describe the attack model of AES, PRESENT and GIFT respectively.

TABLE 6. Present comparison with the state of the art.

Cipher	Platform	Ref		Area (Slices)	TP (Mbps)	TP/A (Mbps/Slices)
PRESENT-80	Spartan-6	[20] [1]		74	206.45	2.799
		[20] [1,3]		69	51.61	0.748
	Kintex-7	[19]		103	200	1.942
		This work	"Portable"	68	206.45	3.036
			"FF"	130		1.588
			"SRL"	101		2.044
PRESENT-128	Spartan-3	[11] [3]		83 +1 BRAM	6.02	0.073
	Spartan-6	[20] [1]		74	206.45	2.799
		[20] [1,3]		69	51.61	0.748
	Virtex-5	[10] [2]		62	22.94	0.370
		[6]		87	206.45	2.373
	Kintex-7	This work	"Portable"	75	206.45	2.753
			"FF"	123		1.678
			"SRL"	151		1.367
GIFT-64-128	Kintex-7	This work	"Portable"	50	228.57	4.571
			"FF"	110		2.078
			"SRL"	133		1.718

[1] Key is pre-computed and stored in memory.
[2] Using 8-bit data path.
[3] Using 16-bit data path.

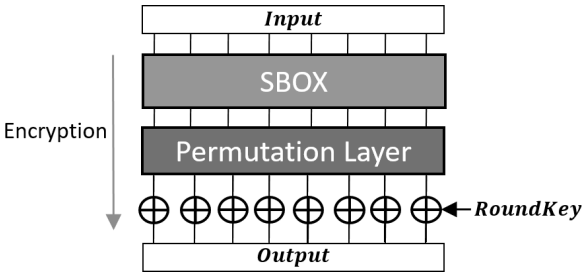

FIGURE 8. CPA attack principle at the last round.

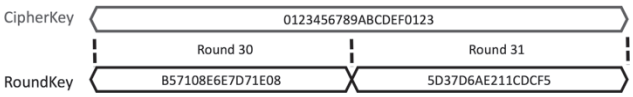

FIGURE 9. Last roundkeys of PRESENT-80.

FIGURE 10. Last roundkeys of PRESENT-128.

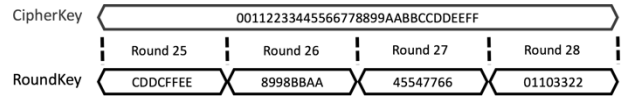

FIGURE 11. Last roundkeys of GIFT-64-128.

According to Figure 13 and 14, the attack model of AES and PRESENT is almost the same: for 8 bits of key (*RKx*), 8 bits are used. The main difference is the number of SBOX used during the attack. As a reminder, the SBOX of AES is on 8 bits and on 4 bits for PRESENT. Therefore, 2 SBOXes are used during the attack of PRESENT, instead of only one for AES. Moreover, a bigger SBOX implies bigger leaks.

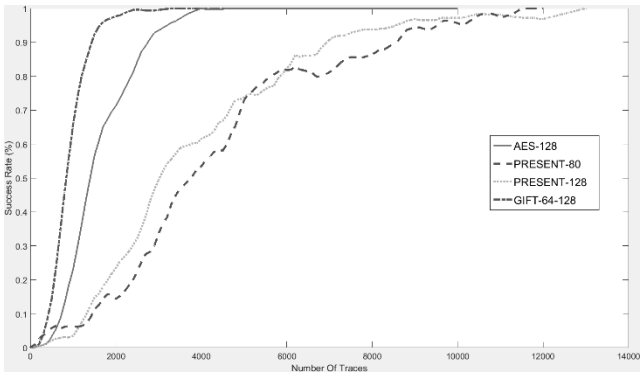

FIGURE 12. Success Rate results of AES, PRESENT-80, PRESENT-128 and GIFT-64-128.

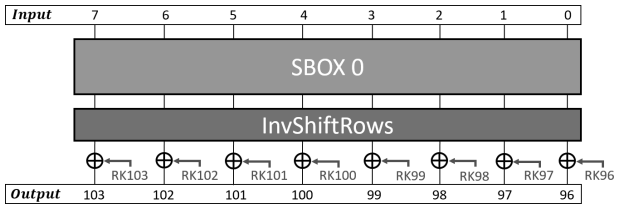

FIGURE 13. 8 LSB bits of the round function of AES.

Consequently, AES signs faster than PRESENT. The leak of two SBOXes of the latter one is still poor than that of one AES SBOX. This means, AES key is recovered with fewer traces, even before the smaller key of PRESENT-80.

By comparing both versions of PRESENT, since the only difference is the length of the key, PRESENT-80 is a little less resistant than PRESENT-128, as expected.

As PRESENT, the SBOX of GIFT is on 4-bit, but only two bits of key are used per SBOX. According to the Figure 15, to attack 8 bits of key (*RKx*), 4 SBOXes are needed, so a total of 16 bits. Note that the *RCx* are bits of the Round

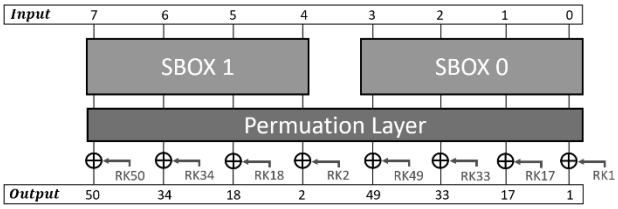

FIGURE 14. 8 LSB bits of the round function of PRESENT.

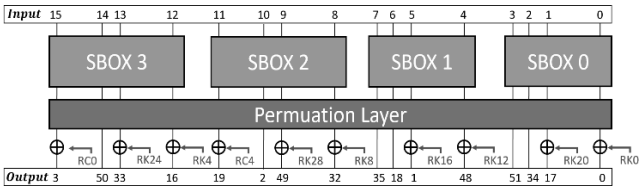

FIGURE 15. 16 LSB bits of the round function of GIFT-64-128.

TABLE 7. Execution time of CPA attack, with 13 000 traces.

Cipher	Number of attack round	Correlation time (s)
AES-128	1	19
PRESENT-80	2	44.3
PRESENT-128	3	68.6
GIFT-64-128	4	240.4

Constant of GIFT. These additional 8 bits improve the correlation process by eliminating wrong hypothesis. Consequently, the attack is less sensitive to noise, which reduces greatly the number of traces needed and makes GIFT the least resistant against CPA attack, even behind AES.

In our experimental setup, the data acquisition time is linear and requires 1 hour to capture 5000 traces, regardless of the algorithm. Then depending on the cipher used, attack processing time takes from a few seconds to minutes as summarized in Table 7. Note that correlation times are obtained using MATLAB scripts, executed on 2,5 GHz Intel Core i7 processor with 16 Go DDR3. Also, they depend on the number of traces to correlate. Values can be different on another platform but the main idea is the resulting trend. The correlation time of GIFT is around 12 time longer than AES. This gap is the result of the difference between the number of attack round and the complexity of the reverse operations. Because of its bitwise design and its use of key, GIFT needs several formatting operations, as decimal/ binary conversions, which increase the correlation time. PRESENT, which uses fewer formatting operations, is on average slower than AES, by a factor 2 to 4, depending on the version of PRESENT.

VI. CONCLUSION

There is an increasing need for security in very constrained devices and standards like AES are not well adapted, as they require too many resources. In order to provide fair metrics to select the best lightweight block cipher according to the constraints, this paper compared several ''Round-Based'' alternatives, especially PRESENT and the recently published block cipher GIFT, both based on SPN. According to this study, GIFT is the most effective, by reducing the area and increasing the throughput by about 20% for both. Regarding Side Channel Analysis, GIFT is the least robust of the studied ciphers. Its performances in terms of area and throughput deeply decrease its resistance against CPA, by a factor of 4 compared to PRESENT. Thanks to the evaluation of several implementations and their security level, this study demonstrates that PRESENT has the best area and security trade-off, among these ciphers and it is the best candidate to ensure security in very constrained devices.

REFERENCES

[1] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, ''The LED block cipher,'' in *Cryptographic Hardware and Embedded Systems* (Lecture Notes in Computer Science), vol. 6917. Berlin, Germany: Springer, 2011, pp. 326–341.
[2] S. Banik *et al.*, ''Midori: A block cipher for low energy,'' in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2015, pp. 411–436.
[3] C. Beierle *et al.*, ''The SKINNY family of block ciphers and its low-latency variant MANTIS,'' in *Advances in Cryptology-CRYPTO*, vol. 9815, M. Robshaw and J. Katz, Eds. Berlin, Germany: Springer, 2016, pp. 123–153.
[4] A. Bogdanov *et al.*, ''PRESENT: An ultra-lightweight block cipher,'' in *Cryptographic Hardware and Embedded Systems-CHES*, vol. 4727, P. Paillier and I. Verbauwhede, Eds. Berlin, Germany: Springer, 2007, pp. 450–466.
[5] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, ''GIFT: A small present—Towards reaching the limit of lightweight encryption,'' in *Proc. Cryptograph. Hardw. Embedded Syst. (CHES)*, Taipei, Taiwan, Sep. 2017, pp. 321–345.
[6] N. Hanley and M. ONeill, ''Hardware comparison of the ISO/IEC 29192-2 block ciphers,'' in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Aug. 2012, pp. 57–62.
[7] P. Yalla and J. P. Kaps, ''Lightweight cryptography for FPGAs,'' in *Proc. Int. Conf. Reconfigurable Comput. FPGAs*, Quintana Roo, Mexico, Dec. 2009, pp. 225–230.
[8] M. Sbeiti, M. Silbermann, A. Poschmann, and C. Paar, ''Design space exploration of present implementations for FPGAs,'' in *Proc. 5th Conf. Southern Program. Logic (SPL)*, São Carlos, Brazil, Apr. 2009, pp. 141–145.
[9] *Espresso*. Accessed: Nov. 16, 2018. [Online]. Available: https://ptolemy.berkeley.edu/projects/embedded/pubs/downloads/espresso/index.htm
[10] J. J. Tay, M. L. D. Wong, M. M. Wong, C. Zhang, and I. Hijazin, ''Compact FPGA implementation of PRESENT with Boolean S-box,'' in *Proc. 6th Asia Symp. Qual. Electron. Des.*, Aug. 2015, pp. 144–148.
[11] E. B. Kavun and T. Yalcin, ''RAM-based ultra-lightweight FPGA implementation of PRESENT,'' in *Proc. Int. Conf. Reconfigurable Comput. FPGAs*, Nov. 2011, pp. 280–285.
[12] A. Bogdanov and V. Rijmen, ''Linear hulls with correlation zero and linear cryptanalysis of block ciphers,'' *Des., Codes Cryptogr.*, vol. 70, no. 3, pp. 369–383, Mar. 2014.
[13] F.-X. Standaert, G. Rouvroy, and J.-J. Quisquater, ''FPGA implementations of the DES and triple-DES masked against power analysis attacks,'' in *Proc. Int. Conf. Field Program. Logic Appl.*, Madrid, Spain, Aug. 2006, pp. 1–4.
[14] G. Jing, Y. Xu, R. Liu, E. Si, N. Shang, and A. Wang, ''Power attack and protected implementation on lightweight block cipher SKINNY,'' in *Proc. 13th Asia Joint Conf. Inf. Secur. (AsiaJCIS)*, Aug. 2018, pp. 69–74.
[15] N. Gupta, A. Jati, A. Chattopadhyay, S. K. Sanadhya, and D. Chang, ''Threshold implementations of GIFT: A trade-off analysis,'' IACR Cryptol. ePrint Arch., Tech. Rep. 2017/1040, 2017, p. 16. [Online]. Available: http://eprint.iacr.org/2017/1040
[16] J. Breier, D. Jap, and S. Bhasin, ''SCADPA: Side-channel assisted differential-plaintext attack on bit permutation based ciphers,'' in *Proc. Des., Automat. Test Eur. Conf. Exhib.*, Mar. 2018, pp. 1129–1134.
[17] S. Patranabis, J. Breier, D. Mukhopadhyay, and S. Bhasin, ''One plus one is more than two: A practical combination of power and fault analysis attacks on PRESENT and PRESENT-like block ciphers,'' in *Proc. Workshop Fault Diagnosis Tolerance Cryptogr. (FDTC)*, Taipei, Taiwan, Sep. 2017, pp. 25–32.

[18] D. Dinu, A. Biryukov, J. Großschädl, D. Khovratovich, Y. Le Corre, and L. Perrin, "FELICS—Fair evaluation of lightweight cryptographic systems," in in *Proc. NIST Workshop Lightweight Cryptogr. (NIST)*, 2015.
[19] W. Diehl, F. Farahmand, P. Yalla, J. P. Kaps, and K. Gaj, "Comparison of hardware and software implementations of selected lightweight block ciphers," in *Proc. 27th Int. Conf. Field Program. Logic Appl.*, Sep. 2017, pp. 1–4.
[20] C. A. Lara-Nino, M. Morales-Sandoval, and A. Diaz-Perez, "Novel FPGA-based low-cost hardware architecture for the PRESENT block cipher," in *Proc. Euromicro Conf. Digit. Syst. Design*, Aug. 2016, pp. 646–650.
[21] *Digilent Nexys 4 Artix-7 FPGA Trainer Board*. Accessed: Jan. 11, 2019. [Online]. Available: https://www.xilinx.com/products/boards-and-kits/1-3yznp5.html#overview
[22] *AES Standard Core—Xilinx, Altera, Microsemi, Lattice and ASIC—Helion Technology*. Accessed: Jan. 11 2019. [Online]. Available: https://www.heliontech.com/aes_std.htm
[23] *AES Tiny Core—Xilinx, Altera, Microsemi, Lattice and ASIC—Helion Technology*. Accessed: Jan. 11, 2019. [Online]. Available: https://www.heliontech.com/aes_tiny.htm
[24] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2004, pp. 16–29.

LOÏC DALMASSO received the Engineering degree in electronics from Polytech Montpellier, France, in 2017. He is currently pursuing the Ph.D. degree in secured approaches for embedded systems with the Montpellier Laboratory of Informatics, Robotics and Microelectronics, University of Montpellier.

FLORENT BRUGUIER received the M.S. and Ph.D. degrees in microelectronics from the University of Montpellier, France, in 2009 and 2012, respectively. In 2015, he joined the ADAC Team as an Associate Professor. Since 2016, he has been in charge of the SECNUM Platform, a platform dedicated to the side-channel attacks. He has co-authored over 30 publications. His research interests include self-adaptive and secured approaches for embedded and high-performance systems.

PASCAL BENOIT received the Ph.D. degree in microelectronics from the University of Montpellier, France, in 2004, and the Habilitation degree from the Montpellier Laboratory of Informatics, Robotics and Microelectronics, University of Montpellier, in 2015. He was a Scientific Assistant with the Karlsruhe Institute of Technology, University of Karlsruhe, Germany. Since 2005, he has been a Permanent Associate Professor with the Montpellier Laboratory of Informatics, Robotics and Microelectronics, University of Montpellier. He has co-authored over 130 publications in books, journals, and conference proceedings. He holds five patents. His research interests include the Internet of Things, from smart sensors to gateways, energy efficiency, and security issues.

LIONEL TORRES received the master's and Ph.D. degrees from the University of Montpellier, in 1993 and 1996, respectively. From 1996 to 1997, he was an IP Core Methodology Research and Development Engineer with ATMEL. From 1997 to 2004, he was an Assistant Professor with Polytech Montpellier and Microelectronics (LIRMM), University of Montpellier. From 2007 to 2010, he was the Head of the Microelectronic Department, LIRMM, where he has been a Full Professor, since 2004. He is currently the Deputy Head of Polytech Montpellier, where he is in charge of research, industrial, and international relationship. Since 2015, he has been the Head of the Cluster of Excellence NUMEV (Digital and Hardware Solutions and Modeling for the Environment and Life Sciences). He has co-authored over 50 journal papers and 150 conference publications. He holds 10 patents. His research interests include system level architecture, with a specific focus on the security and cryptographic applications and nonvolatile computing based on the emerging technologies. He leads several European, national, and industrial projects in these fields.