INL/CON-21-62581-Revision-0

Side-channel Leakage Assessment Metrics: A Case Study of GIFT Block Ciphers

July 2021

William Thomas Unger, Robert J Erbes, Liljana Babinkostova, Mike Borowczak

Idaho National Laboratory

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

INL/CON-21-62581-Revision-0

Side-channel Leakage Assessment Metrics: A Case Study of GIFT Block Ciphers

William Thomas Unger, Robert J Erbes, Liljana Babinkostova, Mike Borowczak

July 2021

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Side-channel Leakage Assessment Metrics: A Case Study of GIFT Block Ciphers

William Unger
Boise State University
Boise, United States
williamunger@u.boisestate.edu

Liljana Babinkostova
Boise State University
Boise, United States
liljanababinkostova@boisestate.edu

Mike Borowczak
University of Wyoming
Laramie, United States
Mike.Borowczak@uwyo.edu

Robert Erbes
Idaho National Laboratory
Idaho Falls, United States
robert.erbes@inl.gov

***Abstract*—Determination of an adequate level of security and providing subsequent mechanisms to achieve it, is one of the most pressing problems regarding embedded computing devices. While there are some solutions available for resource-rich computer systems, direct application of these solutions to resource-constrained environments are often unfeasible. The fundamental problem for such resource-constrained systems is the fact that current cryptographic algorithms utilize significant energy consumption and storage overhead. Both the cryptographic algorithm and its physical implementation affect the resilience of a cryptosystem against side-channel attacks. A side-channel attack represents a process that exploits leakages in order to extract sensitive information such as the key. This paper focuses on Correlation Power Analysis (CPA) which is side-channel attack based on the power consumption leakage. In 2016 the U.S. Commerce Department's National Institute of Standards and Technology (NIST) initiated the call for proposals of new cryptographic algorithms to strengthen the cryptographic defense of networked devices against cyberattacks and to protect the data created by those innumerable device. This work evaluates S-boxes used by NIST candidates PICCOLO, GIFT, and PRESENT, as well as several S-box variants that demonstrated sufficient weaknesses against classical cryptanalysis, for a quantitative comparison in terms of resiliency to CPA attack. Three well-known theoretical metrics are evaluated: transparency order (TO and RTO), non-linearity, and signal-to-noise (SNR) ratio, aiming to characterize the resistance of these S-boxes against adversaries exploiting physical leakages. Experimental results from attacks on an 8-bit XMEGA were obtained via the ChipWhisperer platform and of all the S-boxes evaluated, GIFT64 with a PICCOLO S-box was found to be the most susceptible to CPA. Results showed that variations in TO and RTO were not sufficient to ensure practical CPA resistance and that among S-boxes with equal non-linearity there were no significant differences in the TO and SNR variants.**

***Index Terms*—Correlation Power Analysis, Transparency Order, Non-linearity, Signal-to-Noise Ratio**

I. Introduction

Side-channel attacks, introduced in 1996 by P. Kocher [1], exploit side-channel leakages such as power consumption from a device to extract secret information. Side-channel attacks can be classified into two categories: profiled and non-profiled. The profiled attacks require access to a device, which is a strong assumption and may not always be possible in practice. Type of non-profiled attacks include Differential Power Analysis (DPA), Correlation Power Analysis (CPA) [2] and Mutual Information Analysis (MIA) [3]. The question we ask in this paper is whether certain features make the system more vulnerable to non-profiled attacks such as Correlation Power Analysis? It is highly desirable to have metrics that can indicate a system's vulnerability to this attack as it could guide computer architects in making design decisions and security, power, and performance trade-offs. In order to give a clearer insight on the data leakage, we propose to use the Correlation Power Analysis (CPA) based on the Hamming Weight model and our experimental work includes several theoretical metrics such as Transparency Order (TO) [4], Revisited Transparency Order (RTO) [5], Signal-To-Noise Ratio [6], DPA Signal-To-Noise Ratio (DPA-SNR) [7], and Non-Linearity [8]. Then we show that efficient attacks can be conducted against unprotected implementations of several substitution–permutation network (SPN) based lightweight ciphers such as GIFT [9], [10], PICCOLO [11], and PRESENT [12].

The paper is organized as follows. Section II provides background on SPN ciphers, Correlation Power Analysis, and several metrics which have been presented in literature for ranking constants used within our chosen cipher. Section III describes our specific implementation of CPA attack, and our overall research question and Section IV defines our test environment. Finally, Section V presents our results and Section VI our conclusions.

II. Background

A. Substitution - Permutation Network (SPN) Structure

An $\mathcal{SP}$-network is an iterated block cipher. This means that a certain sequence of computations, constituting a *round*, is repeated a specified number of times. The computations in each round are defined as a composition of specific functions (substitutions and permutations) in a way that achieves Shannon's principle [13] of confusion and diffusion. The *Advanced Encryption Standard (AES)* block cipher [14] is an example of an $\mathcal{SP}$-network. Existing SPN ciphers are not suitable for devices where memory, power consumption or processing power is limited. Lightweight SPN ciphers, such as GIFT [9], [10], PICCOLO [11], and PRESENT [12] provide a solution for running cryptography on low resource devices. There are two versions of GIFT, GIFT-64 is a 28-round SPN cipher and GIFT-128 is a 40-round SPN cipher, both versions

Work supported through the INL Laboratory Directed Research & Development (LDRD) Program under DOE Idaho Operations Office Contract DE-AC07-05ID14517.

have a key length of 128-bit. The GIFT-128 cryptographic scheme is a building block for GIFT-COFB (Authenticated Encryption with Associated Data), one of the 3-round finalists in the ongoing NIST lightweight cryptography standardization process. There are two versions of GIFT, GIFT-64 is a 28-round SPN cipher and GIFT-128 is a 40-round SPN cipher, both versions have a key length of 128-bit.

In GIFT-64 each round consists of an S-Box round function (Substitution Box), a P-Layer round function (permutation layer), and add round key function that introduces content from the private key. The overall structure of 2-rounds GIFT-64 is depicted in Figure 1 .

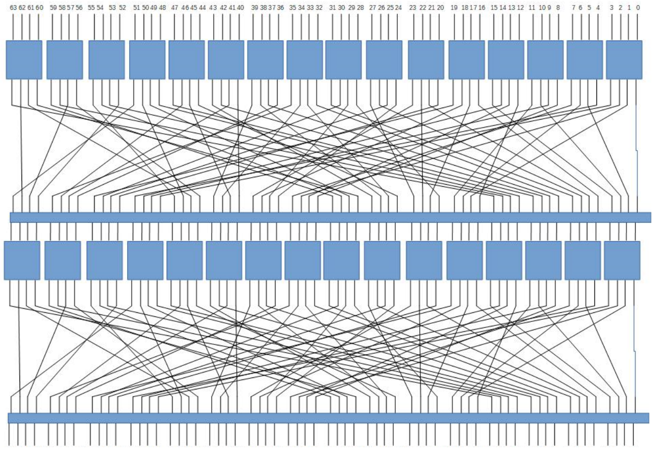

Fig. 1. 2-rounds of GIFT-64

Due to our approach focusing on the characteristics of s-boxes, which serve the same purpose in the different algorithms, our approach is somewhat independent of the specific algorithm. As such, we used the GIFT-64 structure and applied several different S-boxes in order to explain and illustrate our results.

B. *Correlation Power Analysis (CPA) Basics*

The concept of CPA was formalized and studied in [2].This power model is able to derive the private keys of SPN ciphers by using the power consumption of a cryptographic device while the device is undergoing the encryption process. CPA computes the correlation of the actual power draw of the device with the predicted power draw of the device over all possible sub-key guesses to rank the guesses and predict the sub-key. This process is then repeated for all sub-keys in the round key and for as many round keys as needed. In the CPA attack there are two common types of power models used in computing the predicted power draw: hamming distance (HD) between two relevant values which is typically used against hardware, and Hamming weight (HW) of a particular value which is typically used against software.

The CPA methodology consists of the following steps [15]:

- Identify Point of Interest (POI)
- Capture Power Traces
- Compute Intermediary Values
- Classify Hypothetical Power Consumption
- Compare Measurements with Predictions using the Pearson's Correlation Coefficient

This correlation coefficient between two samples R_i and G_i is given by

$$r = \frac{\sum(R_i - R_{avg}) * (G_i - G_{avg})}{\sqrt{\sum(R_i - R_{avg})^2 * \sum(G_i - G_{avg})^2}} \tag{1}$$

The details of our implementation of the CPA model are presented in section III.

C. *Metrics for Side-Channel Assessment*

From the designer's point of view it is important that the S-boxes are chosen carefully to have high resistance against side-channel attacks in addition to classical cryptanalytic attacks. Thus, the natural question is how to measure the resistance of S-boxes against side-channel attacks. Several metrics such as Signal-to-Noise Ratio (SNR), Transparency Order (TO) and Non-linearity have been introduced and studied [4], [5], [7], [16] regarding the S-box resistance against side-channel attacks. In this section we give a brief overview of the metrics used in our study.

We denote by "+" the addition of integers in $\mathbb{Z}$ and by "$\oplus$" the addition mod 2. For a pair of vectors $a = (a_1, a_2, \dots, a_m)$ and $b = (b_1, b_2, \dots, b_m)$ from $\mathbb{F}_2^m$, the scalar product $a \cdot b$ is defined as $a \cdot b = \oplus_{i=1}^m a_i \cdot b_i$. We can view the S-box as a function $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ that maps n input bits to m output bits. Denote

$$F(x) = (F_1(x), F_2(x), \dots, F_m(x))$$

where $F_i : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ for $1 \leq i \leq m$. The *derivative* of F with respect to a vector $a \in \mathbb{F}_2^n$ is the function

$$D_a F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$$

such that $D_a F(x) = F(x) + F(x + a)$. The function $W_F(u, v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{v \cdot F(x) + u \cdot x}$ is called the *Walsh transform* of F. The Walsh transform takes as inputs a function F and constants u, v and outputs an integer and is defined as

$$W_F(u, v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{v \cdot F(x) + u \cdot x} \tag{2}$$

Transparency Order (TO) [4]. The TO of a function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ function F is

$$TO(F) = \max_{\beta \in \mathbb{F}_2^n} \Bigg(|n - 2H(\beta)| - \frac{1}{2^{2n} - 2^n} \sum_{a \in \mathbb{F}_2^{n*}} | \sum_{v \in \mathbb{F}_2^n, H(v)=1} (-1)^{v \cdot \beta} W_{D_a F}(0, v)| \Bigg) \tag{3}$$

In order to provide better quantitative security criterion another form of TO was proposed in [5].

Revisited Transparency Order (RTO) [5]. The RTO of a function $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ function F is

$$RTO(F) = \max_{\beta \in \mathbb{F}_2^n} \left(n - \frac{1}{2^{2n} - 2^n} \sum_{a \in \mathbb{F}_2^{n*}} \left| \sum_{j=1}^{n} \sum_{i=1}^{n} (-1)^{\beta_i \oplus \beta_j} C_{F_i,F_j}(a) \right| \right) \quad (4)$$

Cross-Correlation Spectrum. As a generalization of the Walsh Transform, Cross-Correlation Spectrum is used in some of the newer forms of TO. The Cross-Correlation Spectrum is defined as

$$C_{f_1,f_2}(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f_1(x) \oplus f_2(x \oplus a)} \quad (5)$$

The following metric is used to measure resistance against linear cryptanalysis [16] and has a relationship to SCA success rate. Namely, higher non-linearity results in a cryptographic system being more susceptible to SCA attacks [16].

Non-Linearity [8]. Non-Linearity is defined as

$$NL(F) = 2^{n-1} - \frac{1}{2} \max_{u \in \mathbb{F}_2^n, v \in \mathbb{F}_2^{n*}} |W_F(u,v)| \quad (6)$$

Signal to Noise Ratio (SNR) [6]. The SNR is a probabilistic measurement of the quotient of the signal and noise in a cryptographic implementation. The quotient is defined as

$$SNR = \frac{Var(Signal)}{Var(Noise)} \quad (7)$$

Commonly expressed in decibels as $20\log(SNR)$, the higher the SNR, the stronger the signal or information in the signal relative to the noise or distortion.

DPA Signal-To-Noise Ratio (DPA-SNR) [7]. The DPA-SNR of a function $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ is

$$DPA - SNR(F) = n2^n \left(\sum_{a \in \mathbb{F}_2^n} \left(\sum_{i=0}^{n-1} \left(\sum_{x \in \mathbb{F}_2^n} (-1)^{F_i(x) + x \cdot a} \right) \right) \right) \quad (8)$$

III. Experimental Setup

A. General Implementation of CPA Using Hamming Weight

In our research we chose the point of interest (POI), also known as leakage point, to be after the S-Box function in rounds 2, 3, 4, and 5 of the cipher.

A correlation is computed for each possible value of a targeted sub-key used in the round, and the sub-key with the highest correlation becomes our prediction for that sub-key. As the secret key content is introduced at the end of each round, we target the first round key by predicting the hamming weight of the output from the S-Box function in the second round. This is repeated for all of the sub-keys in a given round key, and for as many round keys is needed in order to recover the full key.

A sample voltage capture for execution of a portion of GIFT64 on the XMEGA is shown in Figure 2. The x-axis indicates time increments and the y-axis is the voltage reading at each point in time. In this figure, the spike in the trace corresponds to the ending of one round and the beginning of another.

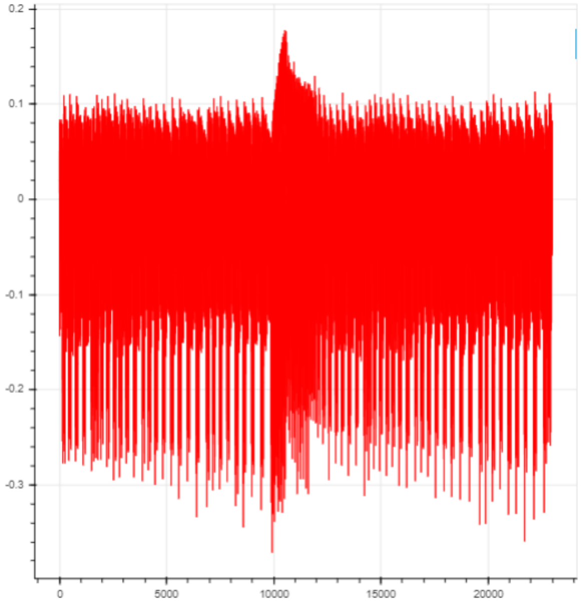

Fig. 2. A sample voltage trace

B. Specific Implementation of CPA Against GIFT64

For our case study on GIFT64, each sub-key are nibbles, so each sub-key has the possible values of 0,1,2,...,15 being all 2^4 possible combinations. For each of the sub-keys in the round key we compute a correlation value for that given possible sub-key. After computing the different correlation values, one for each possible sub-key, we state that the sub-key with the maximum correlation value is our predicted value for that sub-key. That process is repeated for every 4-bit value in the round key, in which the round key is 64-bits long. Because of the way we do this computation we do not care about any round constants and this method will compute them regardless.

After recovering the first 4 round keys, we then use all 4 round keys and reverse the key scheduler to recover the 128-bit private key. We need 4 sequential round keys because of how the GIFT64 cryptographic scheme is structured. In GIFT64, 32-bits of each round key are 'active' while the other 32-bits are either 0 or round constants. In order to recover the full 128-bit private key 4 sequential rounds are needed as 4 sets of 32-bits can recover the full 128-bit private key.

IV. Execution Environment

For our research we used the ChipWhisperer ecosystem [17] to provide the device under test (DUT), control board, and oscilloscope.

The DUT was hosted on the ChipWhisperer CW308 UFO board, and consisted of an ATXMEGA128D4 8-bit RISC micro-controller. The hardware environment used for data collection in shown in Figure 3.

We incorporated a C-language implementation of the GIFT64 algorithm using 8-bit data types into the “simple serial” firmware provided with the ChipWhisperer. All experimental data was collected using this implementation and hardware, apart from the S-Box constants which were modified with each test case.

The CPA attacks themselves and data analysis was performed using Python.

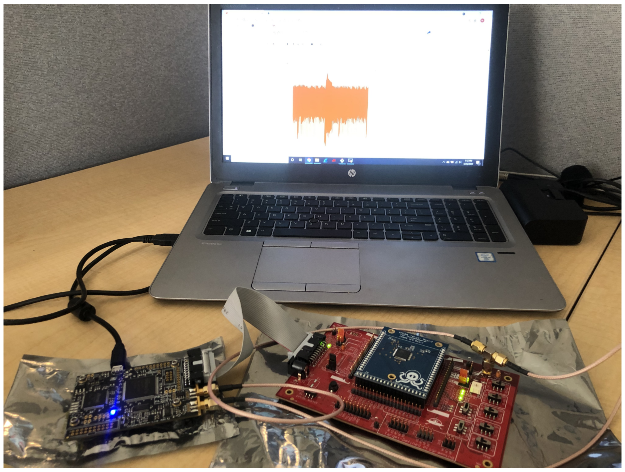

Fig. 3. Execution Hardware - Left:ChipWhisperer Lite - Right:XMEGA on CW308 UFO Board

In this study, we compared the mean success rate of CPA attacks on GIFT64 using several different S-Box look up tables. In order to do this comparison we need to have collections of CPA attacks against each S-Box implementation and a method to compare the results.

CPA Success Rate When conducting a CPA attack, the CPA success rate notes whether an attack successfully recovered the key used by the cryptographic algorithm running on the DUT. For our uses, we only consider first order attacks. The success rate is defined as:

$$SR = \begin{cases} 1, \text{If extracted key is the correct key} \\ 0, \text{Otherwise} \end{cases}$$

In our research, we attempt the CPA attack multiple times and we measure the mean success rate which is the average success rate of attempted key extraction.

Experiment An experiment is a collection of CPA attack results, with the the input to the experiment being a pool of plaintext/voltage array pairs, a threshold cap, and the known private key. An experiment is defined as a loop in which each iteration adds a randomly chosen plaintext/voltage array pair to a data-set, and then the CPA attack is executed using that data-set. The result of the CPA attack, the Success Rate, is stored along with the count of plaintext/voltage array pairs used to perform the attack. The experiment continues looping, until either the size of the data-set reaches the threshold cap or until a success limit of 5 consecutive successful CPA attacks are observed. In our study we used a threshold cap of 150 iterations, and pool of plaintext/voltage array pairs containing 2,000 entries. Example pseudocode is shown below:

Algorithm 1: Experiment Pseudocode

```
Result: Success rate for each trace count
count = 0;
successCount = 0;
list initialized;
results structure initialized;
while $count < Threshold$ do
    Add random plaintext-voltage array pair to list;
    Conduct CPA attack using list;
    count++;
    results[count] = CPA Success Rate (1/0);
    if $Successful$ then
        successCount++;
        if $successCount == 5$ then
            Mark remaining results successful;
            return results;
        end
    end
    else
        successCount = 0;
    end
end
return results;
```

The success limit was implemented in part to to speed up computation and in part through recognition that in most cases, once the CPA attack is successful within a given experiment it is likely to continue being successful with continued addition of more information. We chose the constant 5 to be a success limit based upon early experimental data, but the best parameter to ensure success stability is an open question.

A threshold cap of 150 was chosen based upon trial and error. The goal was to chose a cap which would halt execution of an experiment but also allow each experiment to capture the full progression to a 100% mean success rate. Each of our experiments achieved 100% mean success rate before the reaching the 150th iteration.

The output of an experiment is a set of ordered pairs (x, y) stored in a *Results* array in which the ’x’ value is the size of the data-set used for the CPA attack and the ’y’ value is either 0 or 1 depending on if the CPA attack was successful or not. This means that the Results from an experiment is an array in which the index is the trace/count size and the element in the array is the success rate.

Trial A trial is a collection of Result arrays from many experiments. In our study we consider a trial to be a collection of 100 experiments in which the trial results output has ordered pairs similar to the output of the experiment, but the 'y' values hold the mean success rate of the 100 executions of the experiments.

Algorithm 2: Trial Pseudocode

```
Result: Mean Success Rate for Each Trace Count
count = 0;
results structure initialized;
while count < 100 do
    Execute Experiment;
    Store results of experiment in the results structure;
    count++;
end
Average the results of the experiments for each trace
  count;
Return the average mean success rates;
```

For our analysis we chose seven S-Boxes with varying values for the metrics presented previously. The description of these S-Boxes are shown in the table below consisting of three samples from literature and four S-Boxes generated to purposely vary the metric values of the S-Boxes used in this study.

TABLE I
S-Box Definitions

Input	PICCOLO	GIFT	PRESENT	S2	S4	S1	S3
0	E	1	C	5	1	0	0
1	4	A	5	1	2	E	F
2	B	4	6	7	3	7	E
3	2	C	B	6	4	6	D
4	3	6	9	4	5	4	C
5	8	F	0	0	6	5	B
6	0	3	A	2	7	2	A
7	9	9	D	E	8	1	9
8	1	2	3	3	9	3	8
9	A	D	E	F	A	F	7
A	7	B	F	B	B	A	6
B	F	7	8	A	C	B	5
C	6	5	4	8	D	8	4
D	C	0	7	9	E	9	3
E	5	8	1	C	F	C	2
F	D	E	2	D	0	D	1

For each S-Box we computed several trials and our data will be explored in the Experimental Results section.

V. Experimental Results

In our case study we consider the GIFT64 structure but we substitute 7 differing S-Boxes while keeping everything else constant. Our S-Boxes were chosen to have varying metric scores so we could find correlation between given metric scores and our mean success rate of trials. The S-Boxes we analyzed have the following metric scores shown in Table II below:

TABLE II
S-Boxes with computed metric values

S-Box	NonLinearity	SNR	DPA-SNR	TO	RTO
PICCOLO	4	39.401	3.108	3.666	3.333
GIFT	4	39.348	2.399	3.466	3.066
PRESENT	4	34.665	2.129	3.533	3.266
S2	2	39.968	2.946	3.4	3.266
S4	0	39.252	2.484	2.933	2.933
S1	0	38.582	2.579	3.266	3.133
S3	0	34.148	2.484	2.933	2.933

Note that all of the metric scores are deterministic except for the classical SNR for which we took the average of 100 executions of the SNR measurement. For each of the S-Boxes we conducted several trials and there was not much deviation on repeating trials on the same S-Box so for our study we will display graphs of mean success rate on the first trials for each of the S-Boxes in our case study. The S-Boxes shown have mean success rate shown in Figure 4.

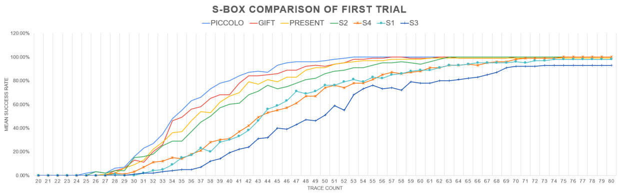

Fig. 4. A comparison on mean success rate of the S-Boxes analyzed

Upon analysis of Figure 4, Table I, and Table II we see that the non-linearity value is a strong indicator of how the mean success rate will be in our data. We note that in our data having a high non-linearity value will be more susceptible to SCA attack and having a lower non-linearity will lead to more resistance to attack.

If the non-linearity of two S-Boxes is the same based upon our data we note that SNR can be a good measure on the indication of mean success rate. That is why we split up the mean success rate graph into multiple graphs with the same value for non-linearity. That can be shown in Figure 5 and Figure 6 below in which the non-linearity values are the same but we can do comparison based upon the SNR values.

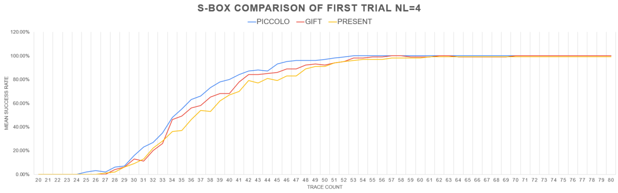

Fig. 5. A comparison on mean success rate of the S-Boxes analyzed with non-linearity $NL = 4$

In Figure 5 we see that the PRESENT S-Box is the most CPA resistant of the non-linearity 4 S-Boxes considered. We also note that PRESENT has a lower SNR value than the other two S-Boxes used in literature.

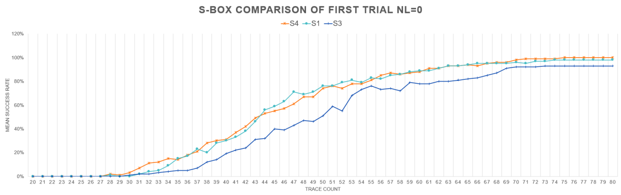

Fig. 6. A comparison of mean success rate of the S-Boxes analyzed with non-linearity $NL = 0$

In Figure 6 we see a clear distinction that the S3 S-Box is sufficiently lower than the other two with non-linearity equal to 0. We also know that the SNR values of S3 is significantly lower than that of the S1 and S4 leading to a correlation in the SNR value with mean success rate assuming their non-linearity values are equal.

We also note that the DPA-SNR has some similar findings that the classical SNR has but neither is a 'perfect' metric indicator for mean success rate assuming non-linearity is equal.

VI. Conclusion

The study of CPA attacks using several theoretical metrics allows the characterization of the S-boxes of several lightweight cryptographic systems. In this paper, we have demonstrated the capabilities of performing CPA on several GIFT-based algorithms with different S-Boxes. Our targeted points of interests are the output of S-Box function of the algorithm during the second, third, forth and fifth rounds of encryption. We attacked the unprotected software implementations of the well-known lightweight cipher, GIFT64 based on seven different S-boxes. The unprotected implementation of GIFT64 with the PICCOLO S-box is more susceptible to CPA attack than GIFT64 with any other S-box as shown in in Table II. Further, our simulations showed that the versions of transparency order (TO and RTO) combined with the signal-to-noise ratio (SNR and DPA-SNR) variants are not sufficient to ensure the practical security of the implementation of the cryptographic systems. Our analysis (e.g., Table 2) suggests that among S-boxes with equal non-linearity, there is no significant differences in terms of TO and SNR variants. It would be interesting to study how these metrics impact the efficiency of other SCA attacks such as Template attacks, Linear Regression attacks or Deep-Learning based SCA.

Finally, one limitation of the work presented is that the attack presented has been conducted against a software implementation of GIFT rather than a hardware implementation. This limitation should be considered for future research.

Acknowledgment

We would like to acknowledge high-performance computing support of the R2 compute cluster (DOI: 10.18122/B2S41H) provided by Boise State University's Research Computing Department.

References

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," Springer, pp. 388–397, 1999.

[2] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *International workshop on cryptographic hardware and embedded systems*, 2004, pp. 16–29.

[3] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual information analysis," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2008, pp. 426–442.

[4] E. Prouff, "Dpa attacks and s-boxes," in *International Workshop on Fast Software Encryption*. Springer, 2005, pp. 424–441.

[5] H. Li, Y. Zhou, J. Ming, G. Yang, and C. Jin, "The notion of transparency order, revisited," *The Computer Journal*, vol. 63, no. 12, pp. 1915–1938, 2020.

[6] R. E. Ziemer and W. H. Tranter, *Principles of Communication Systems, Modulation and Noise*, ser. ohn Wiley & Sons. New York, 1995.

[7] S. Guilley, P. Hoogvorst, , and R. Pacalet, "Differential power analysis model and some results," in *Smart Card Research and Advanced Applications Vi*. Springer, 2004, pp. 127–142.

[8] K. Nyberg, "Differentially uniform mappings for cryptography," *EUROCRYPT'93, Lect. Notes Comput. Sci.*, vol. 765, no. 1, pp. 55–64, 1993.

[9] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "Gift: a small present," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2017, pp. 321–345.

[10] S. Banik, A. Chakraborti, T. Iwata, M. Minematsu, M. Nandi, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "Gift-cofb," *Submission to NIST Competition*, vol. 1, 2019.

[11] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "Piccolo: an ultra-lightweight blockcipher," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2011, pp. 342–357.

[12] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, and C. Vikkelsoe, "Present: An ultra-lightweight block cipher," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2007, pp. 450–466.

[13] C. E. Shannon, "A mathematical theory of communication," in *Bell System Technical Journal*, vol. 27, 1948, pp. 379–423.

[14] J. Daemen and V. Rijmen, "Rijndael: The advanced encryption standard," *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 26, no. 3, pp. 137–139, 2001.

[15] Z. Wang, P. Zhang, C. Chen, and H. Hu, "Pre-processing of power traces in power analysis," *Chin. J. Commun. Technol.*, vol. 50, no. 4, pp. 765–770, 2017.

[16] A. Biryukov, D. Dinu, and J. Großschädl, "Correlation power analysis of lightweight block ciphers: From theory to practice," in *International Conference on Applied Cryptography and Network Security*. Springer, 2016, pp. 537–557.

[17] C. O'Flynn and D. Chen, "Chipwhisperer: An open-source platform for hardware embedded security research," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 2014, pp. 243–260.