Diffusional Side-Channel Leakage From Unrolled Lightweight Block Ciphers: A Case Study of Power Analysis on PRINCE

Ville Yli-Mäyry, Rei Ueno, *Member, IEEE*, Noriyuki Miura, *Member, IEEE*, Makoto Nagata, *Senior Member, IEEE*, Shivam Bhasin, *Member, IEEE*, Yves Mathieu, Tarik Graba, Jean-Luc Danger, *Member, IEEE*, and Naofumi Homma, *Senior Member, IEEE*

***Abstract*— This study investigates a new side-channel leakage observed in the inner rounds of an unrolled hardware implementation of block ciphers in a chosen-input attack scenario. The side-channel leakage occurs in the first round and it can be observed in the later inner rounds because it arises from path activation bias caused by the difference between two consecutive inputs. Therefore, a new attack that exploits the leakage is possible even for unrolled implementations equipped with countermeasures (masking and/or deglitchers that separate the circuit in terms of glitch propagation) in the round involving the leakage. We validate the existence of such a unique side-channel leakage through a set of experiments with a fully unrolled PRINCE cipher hardware, implemented on a field-programmable gate array (FPGA). In addition, we verify the validity and evaluate the hardware cost of a countermeasure for the unrolled implementation, namely the Threshold Implementation (TI) countermeasure.**

***Index Terms*— Low-latency block ciphers, side-channel attacks, unrolled implementation, countermeasures, PRINCE.**

I. Introduction

LOW-LATENCY block ciphers, such as PRINCE [1], MANTIS [2], and QARMA [3], have attracted considerable interest in recent years owing to their ability to perform encryption operations with extremely low latency. Conventional block ciphers are often implemented with a loop architecture, which usually processes one round in one or a few clock cycles and repeats the process until the last round of the cipher. Thus, such ciphers have a compact implementation, albeit with some latency. By contrast, low-latency ciphers are usually implemented with unrolled architectures, which process all the rounds in one or a few clock cycles.

As in the case of conventional ciphers, the threat of side-channel attacks must be considered for low-latency ciphers. Side-channel attacks on block ciphers usually assume that the target cipher is implemented with a loop architecture that stores intermediate results (i.e., round outputs) in registers synchronously. Owing to the widespread use of loop architectures, few studies have investigated the security of unrolled architectures with respect to side-channel attacks. In addition, unrolled architectures are expected to be somewhat resistant to side-channel attacks because unlike loop architectures, they do not use register elements to store intermediate results [4]. Therefore, it is more difficult for the attacker to determine the power consumption in order to perform a side-channel attack. However, studies have shown that conventional first-order attacks, such as correlation power analysis (CPA), are successful even against unrolled implementations [5]–[7].

Attacks on unrolled implementations have been performed against the first round(s) in a known-input or chosen-input scenario, because of the difficulty associated with guessing the values or switching times of the functions of the last round, which are computed by a large combinatorial circuit. In [6], [7], a first-order attack against PRINCE was explored in the known-input scenario. In [5], it was shown that a known-input CPA attack is successful against unrolled PRINCE hardware and the first two rounds should be protected to prevent such attacks. In [7], the authors highlighted the difficulty in applying the conventional countermeasure (i.e., threshold implementation (TI)-like countermeasure [8], [9]) to unrolled architectures without register elements. In [6], a chosen-input CPA attack against an unrolled PRINCE architecture was shown to be successful with significantly fewer traces compared to the corresponding known-input scenario. Meanwhile, previous studies have not ascertained the impact of such chosen-input scenarios or the corresponding valid countermeasures. In particular, it is critical to determine the number of rounds in which the countermeasures are to be applied because the total latency and power consumption of the protected implementation are proportional to this number in the case of unrolled architectures. Therefore, a deeper

Manuscript received June 2, 2020; revised September 9, 2020; accepted October 3, 2020. Date of publication October 23, 2020; date of current version December 1, 2020. This work was supported in part by the Japan Science and Technology Agency (JST) Core Research for Evolutional Science and Technology (CREST) under Grant JPMJCR19K5, Japan, and in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant 17H00729 and Grant 19K24336. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ulrich Rührmair. *(Corresponding author: Ville Yli-Mäyry.)*

Ville Yli-Mäyry, Rei Ueno, and Naofumi Homma are with the Research Insititute of Electrical Communication, Tohoku University, Sendai 980-8577, Japan, and also with the JST CREST, Tokyo 102-0076, Japan (e-mail: ville@riec.tohoku.ac.jp).

Noriyuki Miura and Makoto Nagata are with the Department of Computer Science and Systems Engineering, Kobe University, Kobe 657-8501, Japan.

Shivam Bhasin is with the Physical Analysis and Cryptography Engineering Lab, Temasek Laboratories, Nanyang Technological University, Singapore 637371.

Yves Mathieu, Tarik Graba, and Jean-Luc Danger are with the Institut Mines-Telecom/TELECOM ParisTech, 75009 Paris, France.

Digital Object Identifier 10.1109/TIFS.2020.3033441

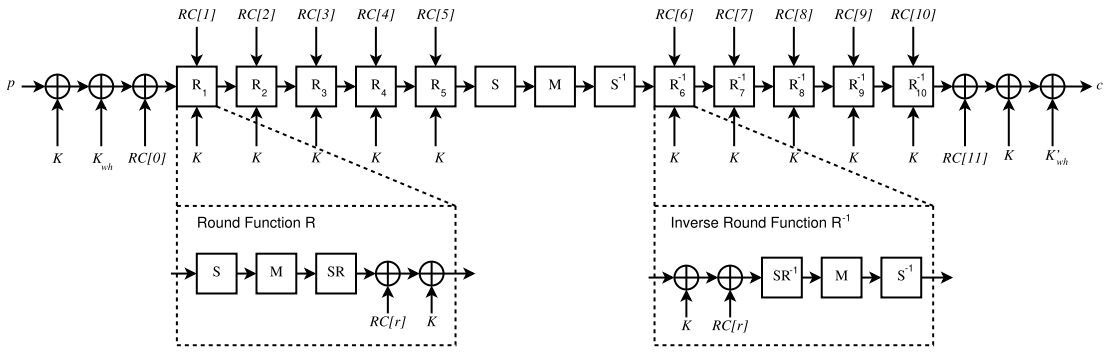


Fig. 1. PRINCE architecture.

understanding of side-channel leakage in unrolled architectures is essential.

This study investigates a unique side-channel leakage observable in the inner rounds of an unrolled block cipher implementation in a chosen-input attack scenario. In particular, a first-order leakage appears in the case of unrolled implementations, even if side-channel countermeasures are applied to the first few rounds. Such a leakage statically arises from the path activation bias caused by the difference between two inputs in the first round, and it is observable during the computation of the subsequent inner rounds. Owing to the nature of this leakage, attacks are possible even for an unrolled implementation with a valid countermeasure in the first round.

The availability of the side-channel leakage is validated by a set of experiments with fully unrolled PRINCE hardware-implemented on a field-programmable gate array (FPGA). In particular, to show that the leakage does not arise from implementation flaws and/or the dynamic characteristics of the circuit implementation (e.g., glitches and coupling effects), we employ three partial implementations of PRINCE, the first few rounds of which are not implemented in hardware but pre-computed in software. Through these experiments, we confirm that the leakage arises from the diffusive property of the input differences and demonstrate that at least the first four rounds of the cipher should be protected.

In addition, we verified the validity and evaluated the hardware cost of possible countermeasures against the above-mentioned attack. For ease of comparison of the side-channel resistance evaluation, in this work we use an extension of a TI-based countermeasure presented in the previous study [7]. More concretely, we consider a possible countermeasure that combines TI with register elements (i.e., pipeline registers or deglitchers) following the insights in [7]. A deglitcher is an element that suppresses glitches, and has commonly been implemented using a latch and a delay path with the same length as the critical path of a function, as shown in [10]. The delay path can also be implemented by a series of buffer elements as, for example, in [11]. In this study, we evaluate the countermeasure by applying TI and flip flops as the deglitching elements to the first round(s) of PRINCE hardware.

The remainder of this paper is organized as follows. Section II reviews related studies and preliminary information on low-latency ciphers. In particular, we describe the structure of PRINCE as a typical low-latency cipher implemented with a fully unrolled architecture. Section III introduces the new side-channel leakage considered in this paper. First, we present the leakage model specified for unrolled architectures. Then, we describe a chosen-input attack based on the leakage. Here, we consider correlation power analysis (CPA) as a typical attack method. In addition, we demonstrate the possibility and limitation of the above-mentioned attack through a set of experiments with three FPGA PRINCE implementations. Section IV verifies the validity and evaluates the cost of masking countermeasures specified for the unrolled PRINCE implementation as a case study. Finally, Section V concludes the paper by summarizing our findings and discussing directions for future works.

II. Preliminaries and Related Work

First, we briefly describe PRINCE, a typical low-latency block cipher. Compared with conventional block ciphers, such as AES, PRINCE achieves extremely low latency in encryption and decryption when implemented with a fully unrolled architecture. Its design has a symmetrical structure, which allows both encryption and decryption using the same circuit and operation flow. Subsequent low-latency block ciphers, such as MANTIS, have many properties similar to those of PRINCE. In addition to the present study, previous studies related to side-channel attacks on low-latency block ciphers have mainly focused on PRINCE as a typical target.

Figure 1 shows the structure of the PRINCE cipher, where the block size and key size are 64 bits and 128 bits (two 64-bit keys, K_{wh} and K), respectively. The encryption/decryption operation consists of initial key addition, 10 round functions (R), and final key and round constant additions. Each round function consists of five sub-functions: S-Layer (S), M-Layer (M), ShiftRows (SR), AddRoundConstants, and AddRoundKey. The first five rounds perform the five sub-functions in the above-mentioned order, while the last five rounds perform the inverses of these sub-functions in the reverse order. There is an intermediate function block consisting of three sub-functions executed in the order of S-Layer, M-Layer, and inverse S-Layer (S^{-1}) between the first five rounds and the last five rounds.

A first-order CPA attack against a PRINCE implementation without any countermeasure was originally presented in [5].

The CPA employed a pair of known inputs for calculating the dynamic power consumption (i.e., Hamming distance) in the S-Layers of the first and second rounds in an unrolled architecture, and it recovered the entire key from the measured traces. In [6], an extended attack that partially fixes the input values was presented to improve the signal-to-noise ratio (SNR) in the first and second rounds in the observed power traces. Previous studies have also discussed the difficulty in recovering the key from the subsequent rounds and concluded that the first two rounds should be protected as a countermeasure.

In [7], the threshold implementation (TI) scheme [8], [9], which is known as a masking scheme with proven security, was applied to PRINCE. The evaluation in [7] suggested that TI has two drawbacks when applied to an unrolled architecture. The first drawback is that TI requires register elements for intermediate state values to decompose the circuit functions. Accordingly, two designs were proposed. One is the true threshold implementation with registers that discard the unrolled structure and make the computation of one encryption/decryption slower by some clock cycles. The other is unrolled TI, which retains the unrolled design and thus does not satisfy the requirements of the original TI design (i.e., non-completeness). The authors mentioned that such an unrolled design suffers from side-channel leakage and that register elements must be used for providing proven security. The second drawback is the large overhead incurred by implementing the TI scheme. Compared with the straightforward unrolled design without any countermeasure, the area with the countermeasure increased from 8,512 GE to 48,012 GE (564%) for the area-optimized design while the latency with the countermeasure increased from 9.0 ns to 13.2 ns (147%) for the speed-optimized design. Here, note that the unrolled TI scheme was applied to only the first and last rounds of PRINCE; however, leakage still occurred. In this sense, a valid countermeasure with an unrolled design has not been developed thus far.

The chosen-input attack scenarios considered in this paper have been used in power analysis against some implementations of block ciphers (e.g., DES [12] and AES [13]). Against DES software, previous attacks, such as those in [14]–[16], have overcome some masking methods by fixing the inputs in a particular manner. Against an AES software implementation in which the inner rounds were not protected, Lu *et al.* [17] demonstrated a first- and second-order differential power analysis (DPA) attack that exploits some inner round intermediate values by fixing certain parts of the inputs. Reparaz and Gierlichs [18] demonstrated a chosen-input DPA attack in the third round of a DES software implementation. The above-mentioned chosen-input techniques were used for keeping the inputs (or outputs) partially constant to estimate the intermediate values in the middle rounds more easily. Note that the previous attacks estimate the secret data from the intermediate values and side-channel leakage of the same (middle) rounds.

Attacks that can target the inner rounds of block ciphers, such as algebraic side-channel analysis (ASCA) [19] and soft analytical side-channel analysis (SASCA) [20], have also been reported. These techniques exploit the input/output value distributions of round functions (e.g., S-boxes), and can be used to attack the inner rounds, where the derivation of the distinguisher function is generally computationally difficult. ASCA/SASCA attacks are reported for software implementations on microcontrollers processing intermediate values one byte at a time. In particular, ASCA assumes that only one (8-bit) S-box is processed at a time, which effectively eliminates algorithmic noise in the observed leakage signal. Such attacks are not applicable to our scenario (i.e., unrolled hardware implementation), where all the internal operations are performed in parallel using the entire data block width (i.e., 64 bits for PRINCE). This implementation does not allow the attacker to obtain the distributions of any *single* byte-wise function trivially. Note also that SASCA is a profiled attack, which makes it difficult to compare the presented unprofiled attack with SASCA in a fair manner.

Previous attacks, such as blind SCA [21], improved blind SCA, [22], and quadrivariate improved blind SCA [23], improve upon classical side-channel analysis methods by relaxing the condition that the input (or output) data of the cipher has to be known by attackers. The attacks can exploit even masked implementations. However, such attacks mainly target software implementations of ciphers. The previous methods are effective under the assumption that the implementations process S-boxes and other functions byte by byte at any observation point with the reduction of the algorithmic noise to zero. This enables the attacker to observe the leakage in any processed block and deduce the Hamming distance of the functions' input/outputs, for example. By contrast, the implementation assumption of this study is completely different. In the case of unrolled hardware implementation, the processing is parallelized and multiple functions are evaluated at any point in a trace. Further, even if the input can be chosen, the intermediate values in the middle rounds cannot be controlled to suppress the algorithmic noise. Therefore, an attacker will experience considerable difficulty in adapting the previous attacks to our case.

In [24], a related technique was presented to employ chosen ciphertexts for establishing a correlation between the Hamming distance of the AES first- and second-round outputs and the measured power consumption. The attack requires concrete calculation of the partial values in the second round. In this study, however, we do not exploit the leakage from register elements assumed in [24].

By contrast, the attack presented in this paper exploits the intermediate values (i.e., the bias in switching) of the first round that manifest themselves as side-channel leakage during the processing of the inner rounds (e.g., second, third, and fourth rounds). In other words, no concrete values must be calculated except for the first-round S-box output change determined from a key guess, and the attack complexity does not change regardless of the "depth" of the inner round that the attack targets. Simply using the Hamming distance of the first-round S-boxes is sufficient to exploit this leakage. Such an attack scenario does not appear in loop architectures. In the case of loop architectures, when a countermeasure is applied to the round function, all the rounds of the cipher are implemented with that countermeasure, and there is no vulnerable inner round to attack. Meanwhile, in the case of unrolled

implementations, a countermeasure is applied independently to each round function. To realize a countermeasure with lower latency, the number of rounds to be protected should be carefully determined. The next section presents a case study on PRINCE hardware to show why such leakage exists in an unrolled implementation as well as how many rounds are affected.

III. Diffusional Side-Channel Leakage

This section describes how the first-round leakage appears in the inner rounds, which can be exploited by side-channel attacks such as CPA. Whereas conventional CPA attacks on inner rounds require additional computational effort for guessing partial keys, the above-mentioned leakage source can be exploited with the same computational cost as that of a first-round attack. For example, a conventional CPA attack on the third round of PRINCE requires at least (i) a 64-bit key guess for calculating the first-round intermediate values, (ii) a 16-bit key guess to compute the partial values going to the third round, and (iii) a 4-bit key guess in the third round to compute a value in a single S-box output. Meanwhile, the chosen-input CPA presented in this study can be performed with 16 sets of 4-bit partial key guesses because the first-round leakage appears in the inner rounds owing to the path activation caused by consecutive inputs (not owing to glitches and coupling effects) in the case of unrolled architectures. We show that the new side-channel leakage originates from the leakage model of unrolled architectures (including pipelined ones), which can be explained by the same principle as that of differential cryptanalysis [25].

A. Leakage Model of Unrolled Architectures

Let $X_i^{(r)}$ be the intermediate state of the S-Layer output of the r-th round in the i-th encryption. In unrolled architecture, one block is usually input and processed for each clock cycle. Therefore, the side-channel leakage of the r-th round in the i-th encryption, $\mathcal{L}_i^{(r)}$, can be represented with the Hamming distance (HD) model as

$$\mathcal{L}_i^{(r)} = HD(X_i^{(r)}, X_{i-1}^{(r)}) + \varepsilon_i^{(r)}, \tag{1}$$

where $HD(X_i^{(r)}, X_{i-1}^{(r)})$ denotes the Hamming distance between $X_i^{(r)}$ and $X_{i-1}^{(r)}$, and $\varepsilon_i^{(r)}$ is an independent noise source from a Gaussian distribution with zero mean and variance σ^2. Here, for a straightforward unrolled architecture (i.e., one with neither pipelining nor deglitchers), the variance of $\varepsilon_i^{(r)}$ is likely to increase with r. Owing to the circuit being completely unrolled, in later rounds, the differences in routing lengths and propagation delays become more prominent, which increases the noise. Here, if a pipeline register or a deglitcher is inserted into the output of the $(r-1)$-th round, the variance of $\varepsilon_i^{(r)}$ is decreased and we can observe $HD(X_i^{(r)}, X_{i-1}^{(r)})$ more accurately owing to the suppression of glitches.

Let us consider the model in the r-th round, $(r \geq 3)$,[1] which indicates that a countermeasure can be applied to the first $(r-1)$ rounds. Let $\Delta X_i^{(r)}$ be the difference between $X_i^{(r)}$ and $X_{i-1}^{(r)}$ (i.e., $\Delta X_i^{(r)} = X_i^{(r)} \oplus X_{i-1}^{(r)}$). Using $\Delta X_i^{(r)}$, we can rewrite Eq. (1) as

$$\begin{aligned}\mathcal{L}_i^{(r)} &= HW(X_i^{(r)} \oplus X_{i-1}^{(r)}) + \varepsilon_i^{(r)} \\ &= HW(\Delta X_i^{(r)}) + \varepsilon_i^{(r)},\end{aligned} \tag{2}$$

where $HW(\Delta X_i^{(r)})$ denotes the Hamming weight of $\Delta X_i^{(r)}$. Here, if $r \geq 3$, we cannot directly compute $\Delta X_i^{(r)}$ (i.e., the hypothetical power consumption corresponding to $\mathcal{L}_i^{(r)}$) because the computation of $\Delta X_i^{(r)}$ requires a large key guess space if the target cipher is a modern block cipher such as PRINCE. Further, $\mathcal{L}_i^{(r)}$ including $\Delta X_i^{(r)}$ has an intrinsic property (i.e., diffusion property) of the block cipher exploited by differential cryptanalysis, which is different from the conventional side-channel leakage derived from round-based and word-serial architectures.

B. Chosen-Input Attack on Unrolled Architectures

The attack exploits the switching bias in the r-th round caused by the output difference of the S-Layer in the first round. More precisely, in the attack, plaintexts are chosen for biasing $\Delta X_i^{(r)}$ similar to the difference probability to provide the correlation between the output difference of the S-box in the first round and $\mathcal{L}_i^{(r)}$. To perform the attack with a feasible computational complexity, we should employ inputs that change only one S-box output in the first round. For this purpose, the attack is described as follows:

1) Generate the i-th chosen plaintext that makes only the j-th S-box output change in the first round.
   For the case of PRINCE, the i-th chosen plaintext for attacking the j-th key nibble $(0 \leq j \leq 15)$ is generated as
   $$P_{i,j} = \text{0x0000\_0000\_0000\_000}p_{i,j} \lll (4j), \tag{3}$$
   where $p_{i,j}$ is a 4-bit random number and $\lll (4j)$ denotes the $4j$-bit cyclic shift to the left. If we input $P_{1,j}, P_{2,j}, \ldots$ to the unrolled architecture of PRINCE, the j-th S-box output is changed (activated) in the first round, while the others should be unchanged (i.e., no switching occurs).
2) Measure the power traces by inputting $\ldots, P_{i-1,j}, P_{i,j}, \ldots$ and obtain $\mathcal{L}_{i,j}^{(r)}$, which is the r-th-round side-channel leakage from the inputs $P_{i,j}$ and $P_{i-1,j}$.
3) Calculate the Hamming distance of the hypothetical intermediate values in the first round, $\Delta x_{i,j}$, derived from the i-th and $(i-1)$-th outputs of the j-th S-box. For the case of PRINCE, the Hamming distance is given by
   $$\begin{aligned}\Delta x_{i,j} &= S(p_{i,j} \oplus k_j^{(1)} \oplus RC_j^{(1)}) \\ &\quad \oplus S(p_{i-1,j} \oplus k_j^{(1)} \oplus RC_j^{(1)}),\end{aligned} \tag{4}$$
   where $S(\cdot)$ denotes the S-box function, and $k_j^{(1)}$ and $RC_j^{(1)}$ are the j-th nibble of the key guess and round constant in the first round, respectively.[2] In the CPA,

[1] Note that our attack can be also applied to the first and second rounds. However, as the first- and second-round leakage can be exploited directly by the conventional attack [5], we focus on $r \geq 3$ in this paper.

[2] The sum of the secret and whitening keys is guessed as the first-round key.

the correlation coefficient between $\Delta x_{i,j}$ and $\mathcal{L}_{i,j}^{(r)}$ for identifying the j-th key nibble is given by

$$\rho_j = \frac{\sum_i \left(w_{i,j} - \mu_{w_j}\right)\left(\mathcal{L}_{i,j}^{(r)} - \mu_{\mathcal{L}_j^{(r)}}\right)}{\sqrt{\sum_i \left(w_{i,j} - \mu_{w_j}\right)}\sqrt{\sum_i \left(\mathcal{L}_{i,j}^{(r)}\mu_{\mathcal{L}_j^{(r)}}\right)}}, \quad (5)$$

where $w_{i,j} = HW(\Delta x_{i,j})$, and μ_{w_j} and $\mu_{\mathcal{L}_j^{(r)}}$ are the means of $w_{i,j}$ and $\mathcal{L}_{i,j}^{(r)}$ with regard to i for each j, respectively.

In Steps 2 and 3, the value of r in $\mathcal{L}_{i,j}^{(r)}$ might be determined according to the number of protected rounds. For example, $r = 3$ if a countermeasure is applied up to the second round.

Next, as an example, we explain why and how the attack on the first round works with the r-th-round leakage from our chosen plaintext.

The basic principle of the attack is similar to that of differential cryptanalysis. Here, the bias of difference in the inner rounds is exploited under the assumption that a small input difference is induced. Note that the intermediate values themselves are scrambled by a round function. Although a full-round modern block cipher is designed to be sufficiently resistant to differential cryptanalysis, our attack exploits the leakage from switching biases in the first few rounds, which is determined by the diffusion property of round functions. In other words, the amount of switching bias in the first few rounds can be explained by the differential characteristics of the cipher and it can be observed via side-channel leakage.

Then, we describe how the differential characteristics (or activated paths) are correlated to the first-round S-box output difference and how they can be exploited by a side-channel attack. The number of activated paths in the r-th round (i.e., $HW(\Delta X_i^{(r)})$) corresponds to the *branch number* of round functions. The branch number of a linear transformation is a measure of its diffusion power [26]. If the branch number in the r-th round is biased with respect to the input difference, our attack can exploit the bias through side-channel information as $\mathcal{L}_i^{(r)} = HW(\Delta X_i^{(r)}) + \varepsilon_i^{(r)}$.

Let $F(x)$ be a linear transformation acting on x given as w-bit word vectors, where w is the input length, and let $W(x)$ be the weight of a vector x defined as the number of nonzero w-bit words (note that it is **not** the number of nonzero **bits** as in the Hamming weight). In the case of PRINCE, this gives us a lower bound for nibble activation in the M-layer. The *branch number* BN over linear transformation F is $\min_{x\neq 0}(W(x) + W(F(x)))$.

Next, we consider the PRINCE cipher as an example. PRINCE operates on 4-bit words in the S-layer and SR functions, and we let $w = 4$ in this case. We recall from the specification of PRINCE [1] that the 64-bit linear transformation M' is defined as a64 × 64-bit matrix multiplication that is constructed with 16 × 16-bit matrices $\hat{M}^{(0)}$ and $\hat{M}^{(1)}$ as

$$M' = \begin{bmatrix} \hat{M}^{(0)} & 0 & 0 & 0 \\ 0 & \hat{M}^{(1)} & 0 & 0 \\ 0 & 0 & \hat{M}^{(1)} & 0 \\ 0 & 0 & 0 & \hat{M}^{(0)} \end{bmatrix}. \quad (6)$$

TABLE I

UPPER AND LOWER BOUNDS OF BRANCH NUMBER FOR DIFFERENT WORD WEIGHTS W

Word weight W		1	2	3	4
Branch number	lower bound	3	2	1	1
	upper bound	4	4	4	4

The definitions of $\hat{M}^{(0)}$ and $\hat{M}^{(1)}$ are

$$\hat{M}^{(0)} = \begin{bmatrix} M_0 & M_1 & M_2 & M_3 \\ M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_0 & M_1 & M_2 \end{bmatrix},$$
$$\hat{M}^{(1)} = \begin{bmatrix} M_1 & M_2 & M_3 & M_4 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_0 & M_1 & M_2 \\ M_0 & M_1 & M_2 & M_3 \end{bmatrix}, \quad (7)$$

where M_0, M_1, M_2, and M_3 are defined as

$$M_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (8)$$

As can be seen from the definition, the output of M' consists of four 16-bit blocks. Because activating multiple 16-bit blocks cannot produce the minimum weight, it follows that the BN of M' is derived as $\min(BN(\hat{M}^{(0)}), BN(\hat{M}^{(1)}))$.
The BN of both $\hat{M}^{(0)}$ and $\hat{M}^{(1)}$ is 4 in PRINCE. Using the above-mentioned matrices, we can derive the upper and lower bounds for all the numbers of activated nibbles with different word weights: $W(x) = 1, 2, 3, 4$.

Table I lists the upper and lower bounds for PRINCE. The noteworthy case from the perspective of this study is when one nibble is activated in the input ($W(x) = 1$). It follows from this result that when one S-box in the first round is activated, at least three S-boxes and at most four S-boxes are activated in the second round.

The values of $\hat{M}^{(0)}$ and $\hat{M}^{(1)}$ that generate the lower bound of the activated paths when $W = 1$ are given when the S-box output (i.e., the input for the M' function) difference is 1 bit. Meanwhile, an upper bound of 4 is given when the S-box output difference is 2 or more bits. From this observation, the number of values switched in the linear layer and functions is linearly correlated to the number of changed bits in the S-box output in the first round.

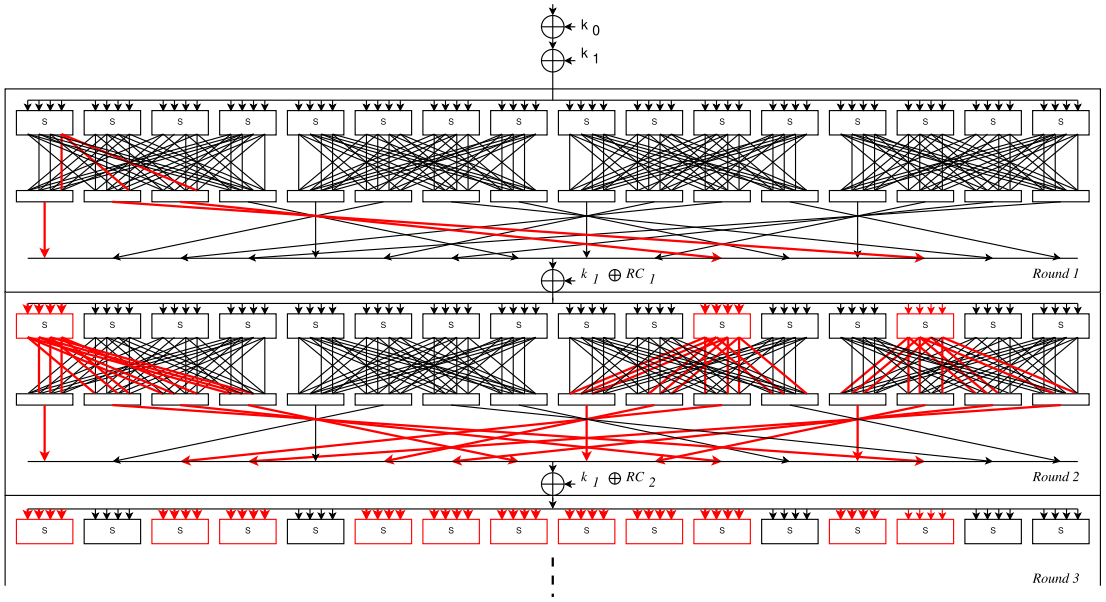


Fig. 2. Examples of activated paths in PRINCE when the difference between the input data pair is 1 bit. The red lines represent the potentially activated paths.

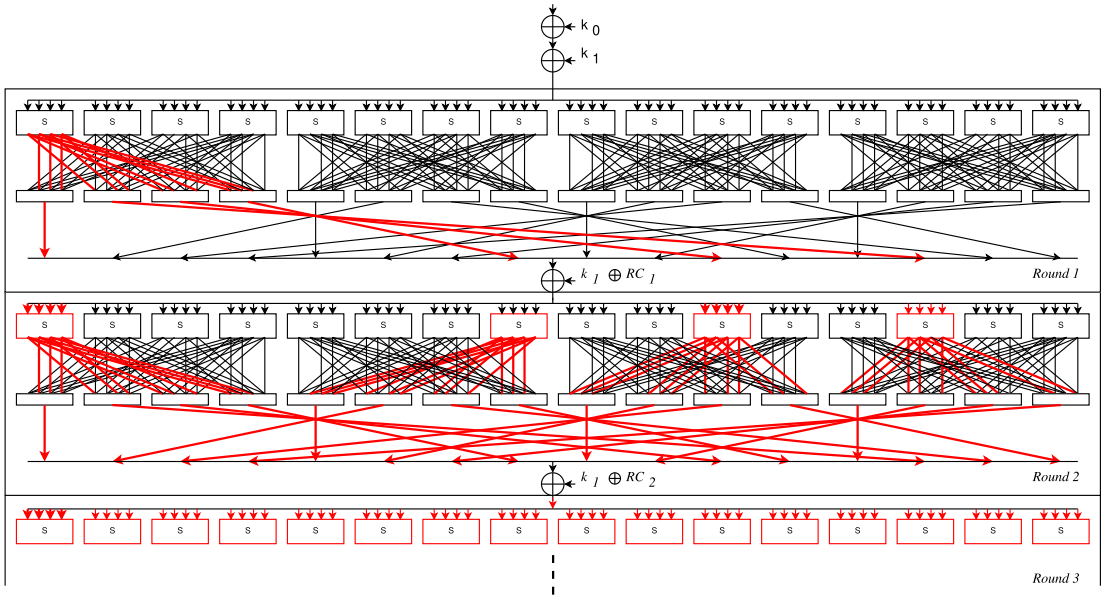


Fig. 3. Examples of activated paths in PRINCE when the difference between the input data pair is 4 bits. The red lines represent the potentially activated paths.

Figures 2 and 3 show the block diagrams of the first two rounds of PRINCE, including how the internal functions in the round are connected, where the S-layer of PRINCE is broken down into 16 individual functions with 4-bit inputs and outputs. Here, two examples are presented to show how the first-round input changes affect the later rounds. In Figs. 2 and 3, the red lines represent paths (potentially) activated when the differences of the first-round S-box outputs are 1 and 4, respectively. We can confirm from the figures that the difference of the first-round S-box output affects the number of values switched in the later rounds. Fig. 2 shows that three S-boxes in the second round are activated because of one bit change in the first-round S-box output and 13 S-boxes are potentially activated in the third round. By contrast, Fig. 3 shows that four S-boxes are potentially activated in the second round and all the S-boxes are potentially activated in the third round. However, note that this figure indicates the number of *potential* activations. The paths are not always activated (as shown in the figure) after the first-round S-box because the intermediate values also depend on (1) the outputs of other functions and (2) the values of key k_1 and round constants RC_1 and RC_2. It is difficult to compute the precise number of activated paths after the first round without knowing the individual key values. However, we can show that on average,

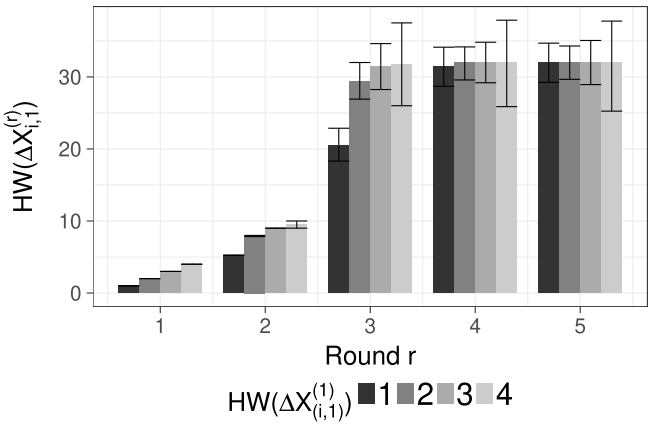


Fig. 4. Average number of bit switches in S-layer outputs for different Hamming distances of input pairs in the first-round S-box.

the switching bias is correlated to the HD of the first-round S-box output, that is, a larger HD in the first-round S-box causes more switching in the first few rounds compared to that in the case of smaller HDs.

Let $\Delta X_{i,j}^{(r)}$ be the output difference of the r-th-round S-Layer corresponding to $P_{i,j}$ and $P_{i-1,j}$. Again, the r-th-round leakage in our attack is represented by $\mathcal{L}_{i,j}^{(r)} = HW(\Delta X_{i,j}^{(r)}) + \varepsilon_{i,j}^{(r)}$. Now, we consider the relation between $\Delta X_{i,j}^{(1)}$ and $\mathcal{L}_{i,j}^{(r)}$, especially in the cases of $r = 3$, 4, and 5. Let $DP_{AVK}(\Delta X_{i,j}^{(1)} \rightarrow \Delta X_{i,j}^{(r)})$ be the average difference from $\Delta X_{i,j}^{(1)}$ to $\Delta X_{i,j}^{(r)}$, where the key is assumed to be given from a uniform distribution. Note that $\Delta X_{i,j}^{(1)}$ is given by $0\text{x}0000\_0000\_0000\_000\Delta x_{i,j} \ll < (4j)$. As described above, it is difficult to compute precise values in an exhaustive manner because $\Delta X_{i,j}^{(3)}$, $\Delta X_{i,j}^{(4)}$, and $\Delta X_{i,j}^{(5)}$ are 64-bit states in which the difference of $\Delta X_{i,j}^{(1)}$ is diffused to the entire state.[3] Therefore, we compute the average differences in the intermediate values with keys drawn from a uniform distribution using a Monte-Carlo-type method.[4]

Figure 4 shows the average, minimum, and maximum values of $HW(\Delta X_{i,1}^{(3)})$, $HW(\Delta X_{i,1}^{(4)})$, and $HW(\Delta X_{i,1}^{(5)})$ under the simulation condition that $HW(\Delta X_{i,1}^{(1)}) = w_{i,1} \in \{1, 2, 3, 4\}$, which correspond to $DP_{AVK}(\Delta X_{i,1}^{(1)} \rightarrow \Delta X_{i,1}^{(3)})$, $DP_{AVK}(\Delta X_{i,1}^{(1)} \rightarrow \Delta X_{i,1}^{(4)})$, and $DP_{AVK}(\Delta X_{i,1}^{(1)} \rightarrow \Delta X_{i,1}^{(5)})$, respectively. Here, the minimum and maximum values are represented by the min-max lines, while the average values are given by the bar graphs. Note that the cases of $HW(\Delta X_{i,1}^{(1)}) = 0$ are omitted because the difference is trivial and it cannot be used for the attack. From the result, for example, we confirm that ρ_1 with $r = 3$ in Eq. (5) would have the highest value if the key guess is correct because $HW(\Delta X_{i,1}^{(3)})$ (and $\mathcal{L}_{i,1}^{(3)}$) is biased depending on $w_{i,j}$.

We also confirm that the input difference of the first round is propagated to the third round, which generates the exploitable leakage owing to the diffusion properties of PRINCE. In addition, while $HW(X_{i,1}^{(4)})$ has a smaller bias than $HW(X_{i,1}^{(3)})$, it is possible to exploit it by our attack. Note that although this figure shows the result of a single S-box, the results of the other S-boxes are similar. The result of Round 5 in Fig. 4 shows that the bias of $HW(\Delta X_{i,1}^{(5)})$ is quite small, which implies that the difference is more scrambled, and the above-mentioned (first-order) attack may experience difficulty in exploiting it for key recovery.

C. Experimental Validation

We confirmed the existence of the aforementioned leakage (related to the first round) from the inner rounds by conducting experimental attacks on three FPGA implementations of PRINCE. Figure 5 shows the the three partial PRINCE implementations used in this experiment, in which we pre-computed the first two, three, or four rounds of PRINCE using software outside the device under evaluation, and implemented the remaining rounds in the hardware according to the PRINCE specification. Here, we pre-computed the first rounds to realize a type of ideal countermeasure. In other words, the computation of the removed rounds cannot contribute to the leakage observed in the hardware implementation, as it is physically removed.[5] Hence, we can completely remove the leakage from other sources, and can exclusively perform and evaluate our attack using the leakage from the inner rounds (i.e., $\mathcal{L}_{i.j}^{(3)}$, $\mathcal{L}_{i.j}^{(4)}$, and $\mathcal{L}_{i.j}^{(5)}$), which corresponds to scenarios in which the first few rounds are protected and the attacker is only able to exploit the leakage in the inner rounds.

Figure 6 shows the experimental setup, with a SAKURA-X board [27] and an Agilent DSO6104A oscilloscope controlled by a laptop PC. The technical details of the power consumption trace acquisition are summarized in Table II. Figure 7 shows an example of power traces and a CPA result obtained from the FPGA implementation.

Table III lists the numbers of key nibbles recovered from $\mathcal{L}_{i,j}^{(3)}$, $\mathcal{L}_{i,j}^{(4)}$, and $\mathcal{L}_{i,j}^{(5)}$, that is, implementations in which two, three, and four rounds were pre-computed, respectively. The results show that we can retrieve a number of sub-keys from $\mathcal{L}_{i,j}^{(3)}$ and $\mathcal{L}_{i,j}^{(4)}$ by CPA using $HW(\Delta X_{i,1}^{(1)})$, which is not implemented in the form of hardware in this experiment. This result clearly indicates that the existence of the first-round information in $\mathcal{L}_{i.j}^{(3)}$ and $\mathcal{L}_{i.j}^{(4)}$ is due to the diffusive effects of the cipher (expressible as difference probability) and not due to either glitches or coupling effects. Furthermore, we cannot recover any key nibble from $\mathcal{L}_{i,j}^{(5)}$, which implies that the difference was sufficiently scrambled in the fifth round and $HW(\Delta X_{i,j}^{(5)})$ was not sufficiently biased in all cases of j.

[3]More precisely, the trail from $\Delta X_{i,j}^{(1)}$ to $\Delta X_{i,j}^{(3)}$ contains two S-Layers and two M-Layers, which implies that at least a 64-bit exhaustive search is required to compute $DP_{AVK}(\Delta X_{i,j}^{(1)} \rightarrow \Delta X_{i,j}^{(3)})$ for all $\Delta X_{i,j}^{(3)}$ owing to the diffusion property of the M-Layer.

[4]Such a Monte Carlo method is rarely used for evaluating the differential (characteristic) probability in the context of differential cryptanalysis. However, it is sufficient for our demonstration to evaluate the bias of $HW(\Delta X_{i,j}^{(r)})$ with a fixed $\Delta x_{i,j} (\neq 0)$.

[5]One side effect of the experimental attacks is that the input signals for the round after the pre-computation are re-aligned. However, this alignment does not change the conclusion of our experiments regarding the demonstration of the presence of new side-channel leakage. Also note that the effect of applying countermeasures requiring registers such as TI is the same as our experimental setting.

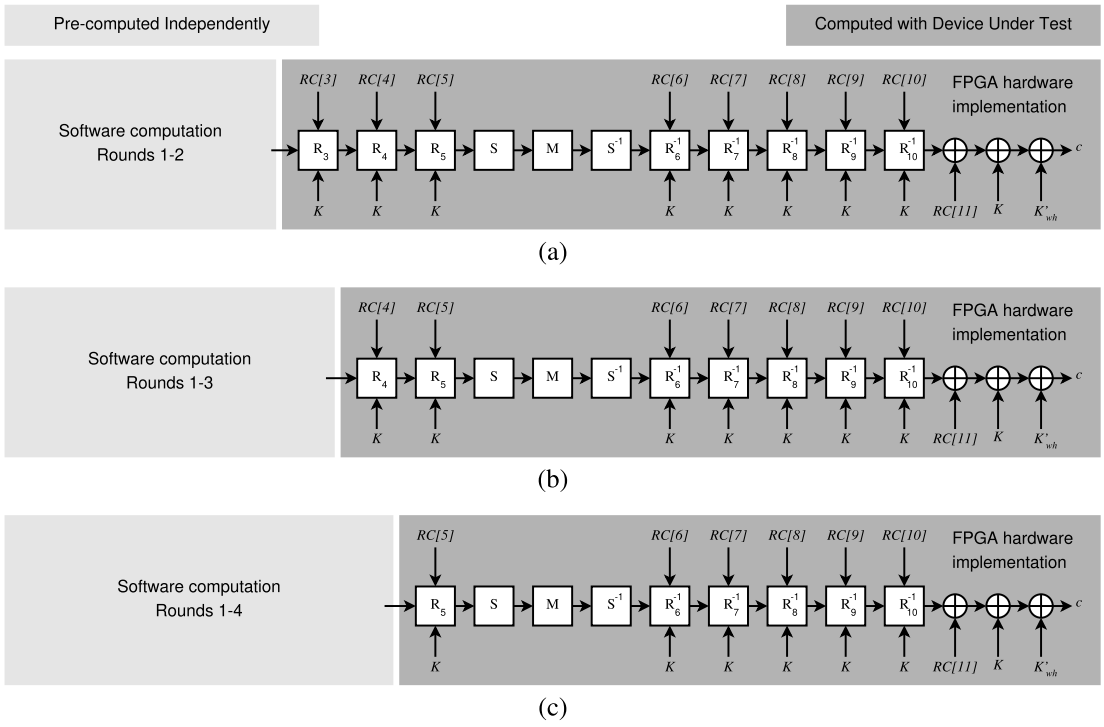


Fig. 5. Three PRINCE implementations for evaluating leakages: (a) $\mathcal{L}_{i,j}^{(3)}$, (b) $\mathcal{L}_{i,j}^{(4)}$, and (c) $\mathcal{L}_{i,j}^{(5)}$, where the first two, three, and four rounds were computed in software outside the device, respectively, and the remaining rounds were implemented in hardware.

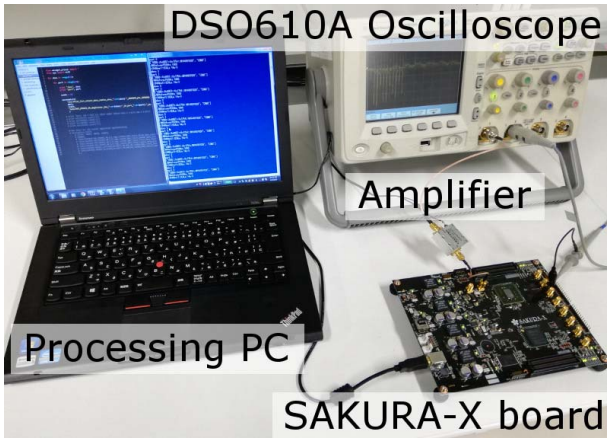


Fig. 6. Experimental environment.

TABLE II
EXPERIMENTAL ENVIRONMENT DETAILS

Digital oscilloscope	Agilent DSO6104A
Bandwidth	DC to 1 GHz
Samples	1000 8-bit samples/trace
Sampling point	Resistor (1Ω) attached to VDD
Implementation platform	Xilinx Kintex-7 on SAKURA-X

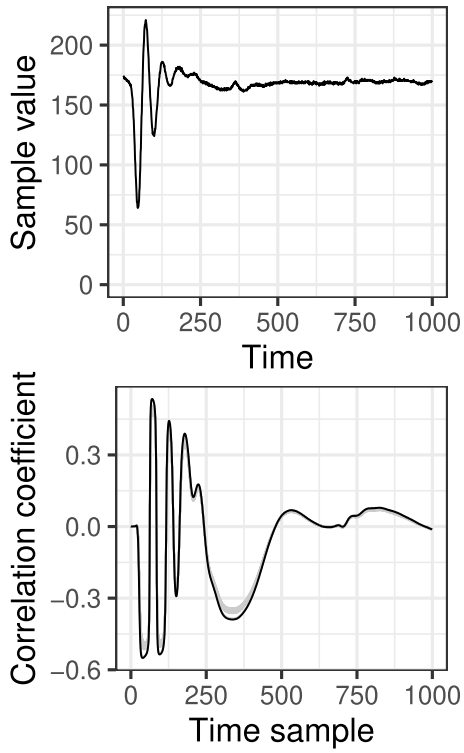


Fig. 7. Example of power trace (top) and successful CPA result (bottom). Correlation coefficients for correct and incorrect key guesses are shown in black and gray lines, respectively.

The result is consistent with the concepts of differential cryptanalysis: it becomes increasingly difficult to find biases in the intermediate state caused by the inputs when the number of mixing functions (rounds) increases. Thus, we can confirm the existence of the first-round leakage in the inner rounds owing to the diffusion property of the cipher.

A more detailed analysis was conducted in which the amounts that were switched in the fourth round were calculated for different S-box outputs from the first round in the

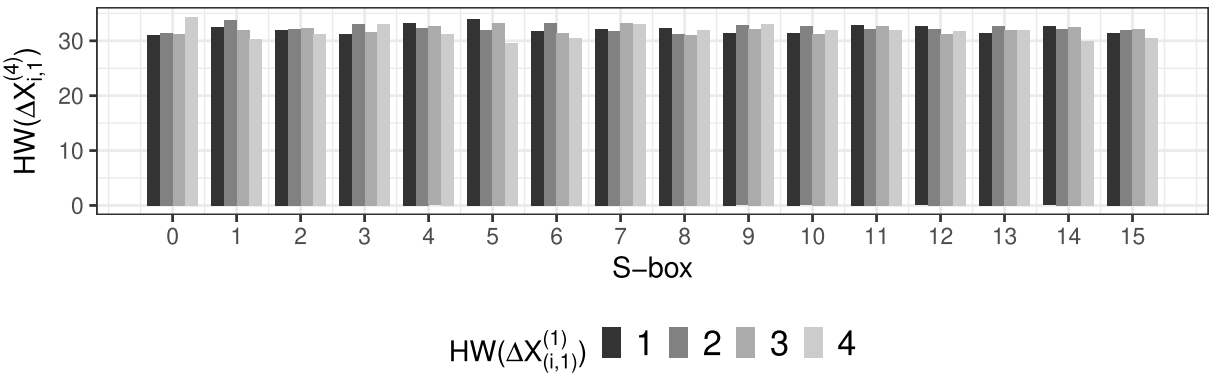


Fig. 8. Switching counts $HW(\Delta X_{i,1}^{(4)})$ for different Hamming distances of input pairs in the first-round S-box.

TABLE III

SUCCESS RATE OF KEY RECOVERY FROM $\mathcal{L}_{i,j}^{(r)}$ FOR THE 3 PARTIALLY PRE-COMPUTED IMPLEMENTATIONS, USING 1 MILLION POWER CONSUMPTION TRACES PER TARGETED PARTIAL KEY

Round number for used leakage	$r=3$	$r=4$	$r=5$
Success rate of key recovery	16/16	1/16	0/16

implementation with the 3-round precomputation. The results are shown in Figure 8, where the vertical bars are colored according to the Hamming weights of the S-box output difference of the first round. The results indicate that the amount of switching in the fourth round does not show particularly strong linearity with respect to the first round S-box output (i.e., Hamming weight $HW(\Delta X_{i,1}^{(1)})$). In addition, Figure 9 shows the Measurements to Disclosure (MTD) results using the switching bias corresponding to that observed in Fig. 8 with j denoting the index of the attacked key nibble. Note that MTD is widely used in this type of evaluation to determine the number of sub-keys that is recovered (or not recovered) when the number of traces is increased. We confirmed that the ranks of the correct key did not show any clear upward trend with the input dataset that was used. These results suggest that it would not be possible to recover additional sub-keys even if more traces were to be used.

D. Discussion

A chosen-plaintext attack on unrolled architectures was also presented in [6]. However, their motivation for choosing plaintexts was to remove the algorithmic noise in the first round in order to enhance the SNR of the measured traces and to reduce the number of traces for successful key recovery. Meanwhile, our attack chooses plaintexts to induce a bias in $HW(\Delta X_{i,j}^{(r)})$ by $\Delta X_{i,j}^{(1)}$, which enables us to exploit the leakage obtained from the inner rounds. To the best of our knowledge, this study is the first to report that the first-round leakage remains in (or propagates to) the inner rounds in unrolled architectures owing to the differential probability.

The first-round information observed during inner-round leakage is unique to unrolled architectures. If we were to implement PRINCE with a round-based or word-serial architecture, it would not be possible to exploit leakage of this nature for CPA attacks. The difference between the architectures is the result of the unique leakage model of the unrolled architecture with $\Delta X_{i,j}^{(r)}$. Such leakage does not appear in round-based and word-serial architectures because the same round function circuit is used repeatedly for all rounds. In this sense, the attack can be enhanced by means of

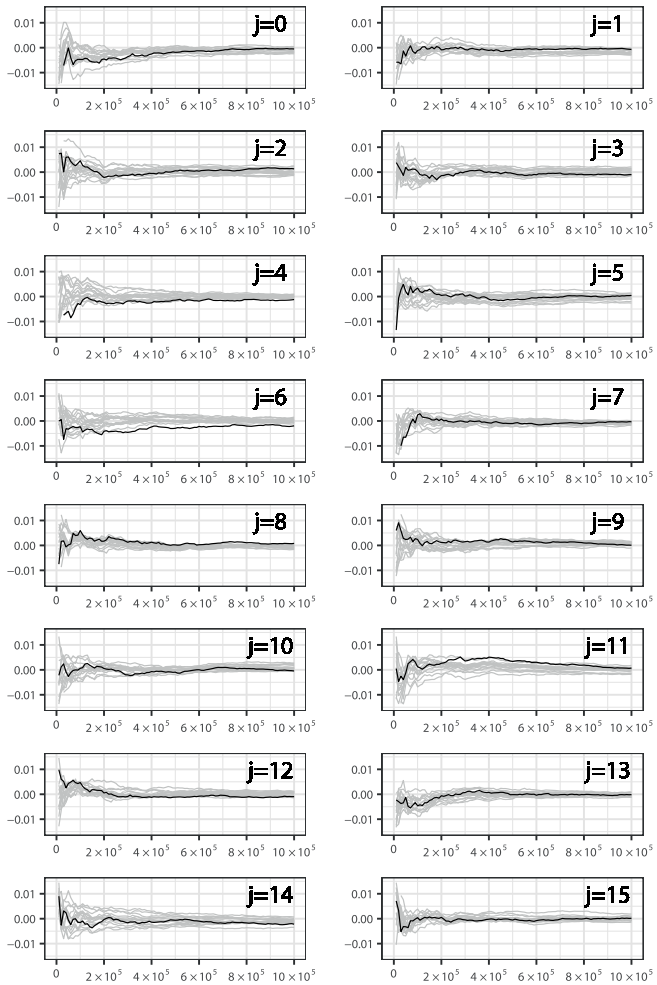


Fig. 9. Measurement to disclosure for 16 S-boxes for the implementation with 3-round precomputation.

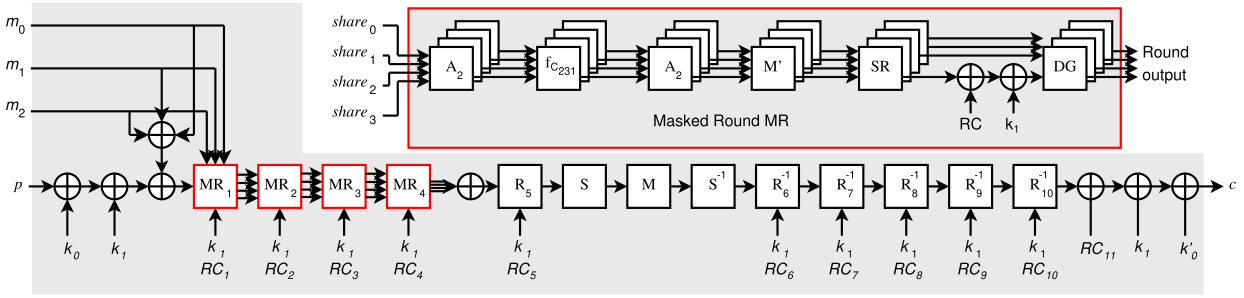


Fig. 10. Overview of unrolled PRINCE hardware with unrolled TI for Rounds 1–4.

sophisticated differential cryptanalyses; however, in this study, we employed a first-order CPA as proof of the concept and did not directly use any other cryptanalysis technique.

The aforementioned side-channel leakage from the inner rounds is also observed in other block ciphers that are implemented with unrolled architectures. Moreover, we can apply the chosen-plaintext attack to a non-fully unrolled architecture when the leakage model of Eq. 1 holds. For example, the AES architecture in [28] with two-round unrolling and four-stage pipelining is a possible target for the attack because the architecture employs block-wise pipelining, where four plaintext blocks are input continuously in four clock cycles.

IV. Countermeasures

This section presents the evaluation of a masking-based countermeasure as applicable to the unrolled architectures of low-latency ciphers. The countermeasure is based on threshold implementation (TI) for PRINCE, proposed by Moradi and Schneider [7], and its subsequent development [29]. We extended the unrolled TI-based countermeasure with register elements, such as pipeline registers and deglitchers. In addition, where the first and last rounds of the cipher were previously implemented with TI [7], we applied their TI-based circuit to the first four rounds in accordance with our findings in Section III, and implemented the remaining rounds without any countermeasure. In this section, we first confirm the validity of such partial-TI-based countermeasures, where the first round(s) are protected by TI, with the same experimental attack as in Section III.C. We then evaluate the hardware overhead of the validated countermeasures. In particular, we show the area that can be saved in the validated partial-TI countermeasure compared to a naive full countermeasure that applies TI to all rounds.

A. Validity Confirmation

First, we show experimental results to confirm the validity of the above-mentioned countermeasure. For evaluating the leakage, we used two methods: (i) attack described above using chosen plaintexts, against the first 1-round, 2-round, 3-round, and 4-round TI implementations, and (ii) a fixed vs. random t-test approach for the 4-round TI implementation.

Figure 10 shows the PRINCE hardware architecture with unrolled TI applied to Rounds 1–4. We implemented the four TI designs individually with an FPGA (Xilinx Kintex-7) on a SAKURA-X board and performed the chosen plaintext attack with the same experimental setup as shown in Section III. The number of nibbles recovered for each design (first 1, 2, 3, and 4-round TI implementations) are shown in Table IV. The MTD results for the first 4-round TI implementation are shown in Fig.11, where j is the index of S-box. These experiment results confirm that the leakage exists in implementations with less than 4 protected rounds, and that the first 4-round TI implementation is enough to suppress the leakage so that key recovery using the shown attack is not possible.

TABLE IV

Success Rate of Key Recovery From $\mathcal{L}_{i,j}^{(r)}$, With Different Number of Rounds Protected With the TI Countermeasure

Protected rounds	1	2	3	4
Success rate of key recovery	16/16	15/16	1/16	0/16

In the context of side-channel security, the t-test is frequently used for leakage assessment [30], [31]. In existing methods, a threshold of $|t| > 4.5$ is defined to distinguish between the existence and the nonexistence of leakage. In other words, if the absolute t-test value at some time index exceeds 4.5, the design is considered to be leaking some side-channel information at that point in the processing. However, note that such information is not guaranteed to be exploitable in a key recovery attack. To evaluate the side-channel leakage, we adopted the non-specific t-test (i.e., fixed vs. random t-test) as described in [30], [32], [33], and we tested for first-order univariate leakage. This test has also been used in previous studies [7], [31].

For the t-test, two sets of data were fed into our designs. The fixed set consists of data pairs with one random 64-bit value and one fixed 64-bit value chosen randomly in advance. The random set is completely composed of random values. A random input is fed into the PRINCE implementation before each encryption so that the circuit is in a random state before encrypting each value from a given set. For all the encryption operations, the three 64-bit masks for TI are generated randomly. To conduct the t-test, we used 10 million power consumption traces observed during the encryptions for each data set. Furthermore, to suppress the effect of changes in the circuit temperature and other environmental factors during

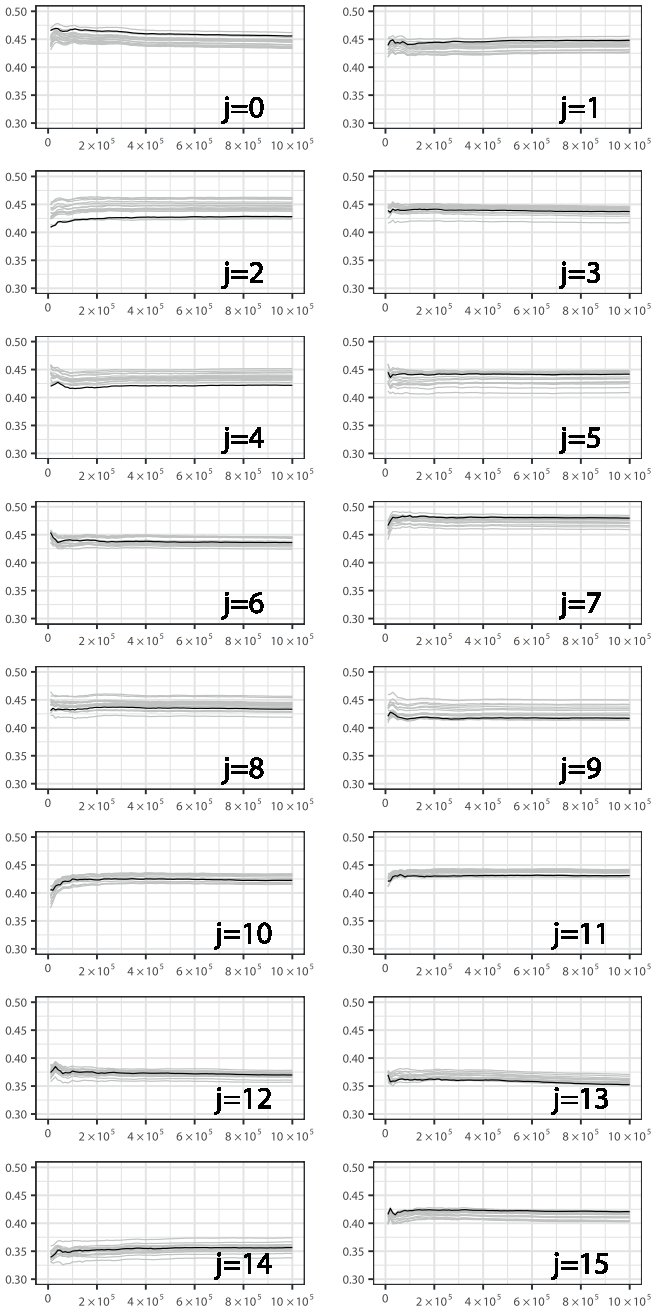


Fig. 11. Measurements to disclosure for 16 S-boxes for the implementation with 4-round TI countermeasure.

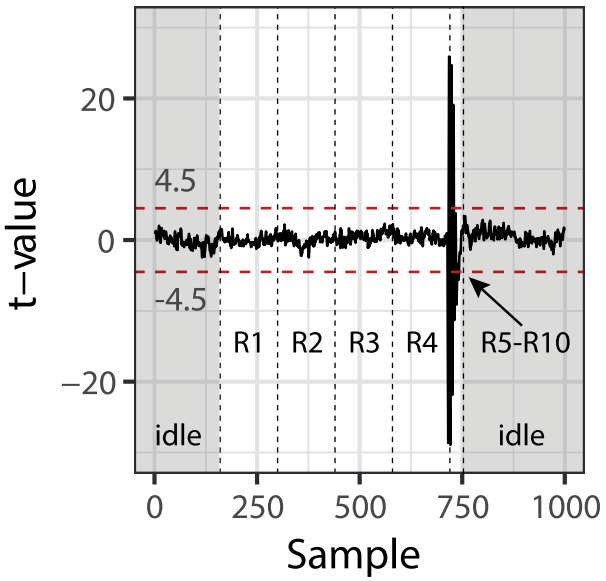


Fig. 12. Result of fixed vs. random t-test for four-round TI-based PRINCE. The bounds $t = 4.5$ and $t = -4.5$ are indicated by red lines. Rounds 5-10 are computed in a completely unrolled way and produce the intensive t-value peak. The areas shaded in grey in the figure show the time before and after the processing, during which the PRINCE hardware is idle.

TABLE V
FPGA IMPLEMENTATION RESULTS OF TI-BASED PRINCE FOR THE EXPERIMENTAL TI IMPLEMENTATION, AND COMPARISON WITH AREA REQUIREMENTS FOR AN IMPLEMENTATION WHERE ALL THE ROUNDS ARE IMPLEMENTED WITH A COUNTERMEASURE

	Area (Slices)	Protected rounds
Without countermeasure	359	Unprotected
TI-based	3,420 (+953%)	1–4
	6,751 (+1,881%)	1–10

the power observations, we fed the two data sets into the circuit in a randomized interleaved manner so that a single observation of the fixed or random sets was made with a probability of 1/2.

The result of the t-test is shown in Fig. 12. The red horizontal lines represent the points where $t = 4.5$ and $t = -4.5$. The result indicates that the t-value during the masked rounds stays within the range of $[-4.5, 4.5]$ and that later rounds result in larger t-values. Here it should be noted that the values for the 5th to the 10th round are computed in a completely unrolled manner. Thus, their power consumption becomes intensive for the sample indices ranging from approximately 700 to 750, which results in higher t-values soon after the termination of the four masked rounds. Also note that idle time occurs before and after the processing, shown in the figure as a shaded area. As mentioned in Section III, the leakage in the later rounds is not exploitable in practice. At the same time, if we require "more" secure countermeasures against unknown attacks, the fully protected implementation would be a possible option.

B. Implementation Overhead

We evaluate the overhead of the above-mentioned TI-based countermeasure implemented on FPGA and ASIC. According to the above results, we consider the unrolled partial-TI implementation where the first four rounds of PRINCE are protected. For reference, we also show the fully protected implementation results in which all 10 rounds of PRINCE were masked. This highlights the amount of logic that can be saved when the leakage is carefully analyzed.

Table V summarizes the implementation results of TI-based PRINCE designs for FPGA (Xilinx Kintex-7), where Area

TABLE VI

ASIC IMPLEMENTATION RESULTS OF TI-BASED PRINCE FOR THE EXPERIMENTAL TI IMPLEMENTATION, AND COMPARISON WITH AREA REQUIREMENTS FOR AN IMPLEMENTATION WHERE ALL THE ROUNDS ARE IMPLEMENTED WITH A COUNTERMEASURE

	Area (GE)	Protected rounds
Without countermeasure	6,144	Unprotected
TI-based	85,127 (1,386%)	1–4
	241,896 (3,937%)	1–10

is given by the number of slices used. The TI-based implementation produced area overheads of 953% and 1881% for the 4-round and the 10-round implementations, respectively. Table VI summarizes the synthesis results for ASIC using Synopsys Design Compiler and NanGate 45nm Open Cell Library with area optimization, where Area is given by two-input NAND gate equivalents (GE).

V. CONCLUSION

We presented a new side-channel leakage for unrolled architectures in chosen-input attack scenarios. The existence and validity of this side-channel leakage were demonstrated through a set of experiments involving PRINCE hardware-implemented on FPGA. In addition, we presented evaluation of masking countermeasures based on the unrolled version of TI. By applying these masking schemes to the first four rounds of PRINCE, we showed that key recovery attacks that exploit the above-mentioned first-order leakage can be thwarted.

Owing to the mechanism by which such leakage occurs, we expect it to also be found in other low-latency block ciphers, such as MANTIS and QARMA. Such leakage may occur in conventional block ciphers if they are implemented with unrolled architectures. On the other hand, the applicability and limitation of attacks exploiting the leakage depend on the diffusive properties of the nonlinear function of the targeted cipher. For example, against ciphers with wide-trail diffusion strategy, such as AES, the strength of the switching bias is much less pronounced. The number of rounds to be protected should be carefully determined depending on the cryptographic algorithm and implementation.

A more detailed security evaluation is required for other types of attacks; although we focused on a straightforward CPA for exploiting the differential probability in this study, other sophisticated attack models would comprise important future works. In addition, other types of attacks, such as collision-based and second-order attacks, should also be studied to defeat the presented countermeasures. Furthermore, a formal approach to analyzing and evaluating the new leakage would be interesting in order to achieve more sophisticated countermeasures.

ACKNOWLEDGMENTS

The authors are grateful for the support.

REFERENCES

[1] J. Borghoff *et al.*, "PRINCE—A low-latency block cipher for pervasive computing applications," in *Proc. ASIACRYPT*, 2012, pp. 208–225.

[2] C. Beierle *et al.*, "The SKINNY family of block ciphers and its low-latency variant MANTIS," in *Proc. Annu. Int. Cryptol. Conf.*, 2016, pp. 123–153.

[3] R. Avanzi, "The QARMA block cipher family. Almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes," *IACR Trans. Symmetric Cryptol.*, vol. 2017, no. 1, pp. 4–44, 2017, doi: 10.13154/tosc.v2017.i1.4-44.

[4] S. Bhasin, S. Guilley, L. Sauvage, and J.-L. Danger, "Unrolling cryptographic circuits: A simple countermeasure against side-channel attacks," in *Proc. CT-RSA*, 2010, pp. 195–207.

[5] V. Yli-Mäyry, N. Homma, and T. Aoki, "Improved power analysis on unrolled architecture and its application to PRINCE block cipher," in *Proc. LightSec*, 2015, pp. 148–163.

[6] V. Yli-Mayry, N. Homma, and T. Aoki, "Chosen-input side-channel analysis on unrolled light-weight cryptographic hardware," in *Proc. 18th Int. Symp. Qual. Electron. Design (ISQED)*, Mar. 2017, pp. 301–306.

[7] A. Moradi and T. Schneider, "Side-channel analysis protection and low-latency in action-case study of PRINCE and Midori," in *Proc. ASIACRYPT*, 2016, pp. 517–547.

[8] S. Nikova, V. Rijmen, and M. Schläffer, "Secure hardware implementation of nonlinear functions in the presence of glitches," *J. Cryptol.*, vol. 24, no. 2, pp. 292–321, Apr. 2011.

[9] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold implementations against side-channel attacks and glitches," in *Proc. ICICS*, 2006, pp. 529–545.

[10] N. Miura *et al.*, "A 2.5ns-latency 0.39pJ/b 289μ^2/Gb/s ultra-light-weight PRINCE cryptographic processor," in *Proc. Symp. VLSI Circuits*, Jun. 2017, pp. C266–C267.

[11] S. Endo *et al.*, "A silicon-level countermeasure against fault sensitivity analysis and its evaluation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 8, pp. 1429–1438, Aug. 2015.

[12] *Data Encryption Standard (Des)*, National Institute of Standards and Technology, FIPS Publication 46-3, Gaithersburg, MD, USA, Oct. 1999.

[13] *Advanced Encryption Standard*, Standard NIST FIPS PUB 197, NIST, Gaithersburg, MD, USA, 2001.

[14] M.-L. Akkar, R. Bévan, and L. Goubin, "Two power analysis attacks against one-mask methods," in *Fast Software Encryption*, B. Roy and W. Meier, Eds. Berlin, Germany: Springer, 2004, pp. 332–347.

[15] J. Lv and Y. Han, "Enhanced des implementation secure against high-order differential power analysis in smartcards," in *Information Security and Privacy*, C. Boyd and J. M. González Nieto, Eds. Berlin, Germany: Springer, 2005, pp. 195–206.

[16] J. Lv, "On two Des.implementations secure against differential power analysis in smart-cards," *Inf. Comput.*, vol. 204, no. 7, pp. 1179–1193, Jul. 2006, doi: 10.1016/j.ic.2006.04.002.

[17] J. Lu, J. Pan, and J. den Hartog, "Principles on the security of aes against first and second-order differential power analysis," in *Proc. ACNS*, J. Zhou and M. Yung, Eds. Berlin, Germany: Springer, 2010, pp. 168–185.

[18] O. Reparaz and B. Gierlichs, "A first-order chosen-plaintext DPA attack on the third round of DES," in *Proc. 16th Int. Conf. Smart Card Res. Adv. Appl. (CARDIS)*, Lugano, Switzerland, Nov. 2017, pp. 42–50, doi: 10.1007/978-3-319-75208-2_3.

[19] M. Renauld and F.-X. Standaert, "Algebraic side-channel attacks," in *Proc. 5th Int. Conf. Inf. Secur. Cryptol.* Berlin, Germany: Springer-Verlag, 2010, pp. 393–410. [Online]. Available: http://dl.acm.org/citation.cfm?id=1950111.1950148

[20] N. Veyrat-Charvillon, B. Gérard, and F.-X. Standaert, "Soft analytical side-channel attacks," in *Proc. ASIACRYPT*. Cham, Switzerland: Springer, 2014, pp. 282–296.

[21] Y. Linge, C. Dumas, and S. Lambert-Lacroix, "Using the joint distributions of a cryptographic function in side channel analysis," in *Proc. 5th Int. Workshop Constructive Side-Channel Anal. Secure Design (COSADE)*, Paris, France, Apr. 2014, pp. 199–213, doi: 10.1007/978-3-319-10175-0_14.

[22] C. Clavier and L. Reynaud, "Improved blind side-channel analysis by exploitation of joint distributions of leakages," in *Proc. 19th Int. Conf. Cryptograph. Hardw. Embedded Syst. (CHES)*, Taipei, Taiwan, Sep. 2017, pp. 24–44, doi: 10.1007/978-3-319-66787-4_2.

[23] C. Clavier, L. Reynaud, and A. Wurcker, "Quadrivariate improved blind side-channel analysis on Boolean masked AES," in *Proc. COSADE*, in Lecture Notes in Computer Science, vol. 10815. Cham, Switzerland: Springer, 2018, pp. 153–167.

[24] A. Moradi and T. Schneider, "Improved side-channel analysis attacks on Xilinx bitstream encryption of 5, 6, and 7 series," in *Constructive Side-Channel Analysis and Secure Design* (Lecture Notes Computer Science), vol. 9689. Cham, Switzerland: Springer, 2016, pp. 71–87.

[25] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*. Berlin, Germany: Springer-Verlag, 1993.

[26] J. Daemen and V. Rijmen, "The block cipher Rijndael," in *Smart Card Research and Applications*, J.-J. Quisquater and B. Schneier, Eds. Berlin, Germany: Springer, 2000, pp. 277–284.

[27] *Side-Channel Attack Standard Evaluation Board (SASEBO)*. Accessed: Oct. 31, 2020. [Online]. Available: https://www.risec.aist.go.jp/project/sasebo/

[28] S. K. Mathew *et al.*, "53 gbps native GF$(2^4)^2$ composite-field AES-encrypt/decrypt accelerator for content-protection in 45 nm high-performance microprocessors," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 767–776, Apr. 2011.

[29] D. Bozilov, M. Knezevic, and V. Nikov, "Optimized threshold implementations: Minimizing the latency of secure cryptographic accelerators," in *Proc. 18th Int. Conf. Smart Card Res. Adv. Appl. (CARDIS)*, Prague, Czech Republic, in Lecture Notes in Computer Science, vol. 11833, S. Belaïd and T. Güneysu, Eds. Cham, Switzerland: Springer, Nov. 2019, pp. 20–39, doi: 10.1007/978-3-030-42068-0_2.

[30] G. Becker, "Test vector leakage assessment (TVLA) methodology in practice," Nat. Inst. Standards Technol., Comput. Secur. Resour. Center, NIST Non-Invasive Attack Test. Workshop, Tech. Rep., 2011.

[31] T. Schneider and A. Moradi, "Leakage assessment methodology," *J. Cryptograph. Eng.*, vol. 6, no. 2, pp. 85–99, Jun. 2016.

[32] A. A. Ding, C. Chen, and T. Eisenbarth, "Simpler, faster, and more robust t-test based leakage detection," in *Proc. COSADE*, 2015, pp. 163–183.

[33] F. Durvaux, F.-X. Standaert, and S. M. Del Pozo, "Towards easy leakage certification," in *Proc. CHES*, 2016, pp. 40–60.

Makoto Nagata (Senior Member, IEEE) received the B.S. and M.S. degrees in physics from Gakushuin University, Tokyo, in 1991 and 1993, respectively, and the Ph.D. degree in electronics engineering from Hiroshima University, Hiroshima, in 2001.

He was a Research Associate with Hiroshima University from 1994 to 2002, and an Associate Professor at Kobe University from 2002 to 2009 and promoted to a Full Professor in 2009. He is currently a Professor with the Graduate School of Science, Technology, and Innovation, Kobe University, Kobe, Japan. His research interests include design techniques targeting high-performance mixed analog, RF and digital VLSI systems with a particular emphasis on power/signal/substrate integrity and electromagnetic compatibility, testing and diagnosis, three-dimensional system integration, as well as their applications for hardware security and safety. He is a Senior Member of IEICE. He was a member of a variety of technical program committees of international conferences such as the Symposium on VLSI Circuits from 2002 to 2009, the Custom Integrated Circuits Conference from 2007 to 2009, and the Asian Solid-State Circuits Conference from 2005 to 2009. He has been a member of a variety of technical program committees of international conferences such as the International Solid-State Circuits Conference since 2014. He is also a member of the European Solid-State Circuits Conference. He has been the Chair of the Technology Directions Subcommittee for International Solid-State Circuits Conference since 2018. He was the Technical Program Chair, the Symposium Chair, and an Executive Committee Member of the Symposium on VLSI Circuits from 2010 to 2011, from 2012 to 2013, and from 2014 to 2015, respectively. He was the Past Chair of the IEEE Solid-State Circuits Society (SSCS) Kansai Chapter from 2017 to 2018 and is currently an AdCom Member of the IEEE SSCS and also serves as a Distinguished Lecturer (DL) of the society. He has been an Associate Editor of IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS since 2015.

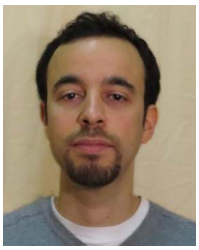
Ville Yli-Mäyry received the B.Sc. degree from the Tampere University of Technology, Finland, and the M.Sc. and Ph.D. degrees from Tohoku University, Japan. He is currently a Researcher with Tohoku University. His research interests include embedded systems security, and the implementation and security evaluation of next generation cryptographic processors.

Rei Ueno (Member, IEEE) is an Assistant Professor with the Research Institute of Electrical Communication, Tohoku University, and is also joining the JST as a Researcher of the PRESTO Project. His research interests include arithmetic circuits, cryptographic implementations, formal verification, and hardware security. He received the Kenneth C. Smith Early Career Award in Microelectronics from ISMVL in 2017.

Noriyuki Miura (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan. He is currently an Associate Professor with Kobe University, Kobe, Japan, and concurrently a JST PRESTO Researcher, working on hardware security and next-generation heterogeneous computing systems. He is currently serving as a TPC Member of A-SSCC and Symposium on VLSI Circuits. He received the Top ISSCC Paper Contributors from 2004 to 2013 and the IACR CHES Best Paper Award in 2014.

Shivam Bhasin (Member, IEEE) received the bachelor's degree from UP Tech, India, in 2007, the master's degree from Mines Saint-Etienne, France, in 2008, and the Ph.D. degree from Telecom Paristech in 2011. He has been a Senior Research Scientist and Principal Investigator with the Physical Analysis and Cryptographic Engineering Laboratory, Temasek Labs, Nanyang Technological University (NTU), Singapore, since 2015. Before joining the NTU, he held the position of a Research Engineer with Institut Mines-Telecom, France. He was also a Visiting Researcher at UCL, Belgium, in 2011, and Kobe University, Japan, in 2013. He regularly publishes at top peer reviewed journals and conferences. Some of his research now also forms a part of ISO/IEC 17825 standard. His research interests include embedded security, trusted computing, and secure designs.

Yves Mathieu is a Full Professor with the Institut Mines-Telecom/TELECOM ParisTech. He is the Vice-Chair of the Education Department of Electronics and Communications. He undertakes research activities inside the "Safe and Secure Hardware" Team with a focus on ASIC design.



Tarik Graba received the master's degree (DEA:Diplôme d'études approfondies) and the Ph.D. degree in electrical engineering from Pierre et Marie Curie University (UPMC), Paris, France, in 2003 and 2006, respectively. He is currently an Associate Professor with the Department of Electronics and Communications, Institut Mines-Telecom/Telecom ParisTech. His research activities include digital ASIC and system on chip design, and hardware security.

Jean-Luc Danger (Member, IEEE) is a Full Professor with Institut Mines-Telecom/TELECOM ParisTech. He is the Head of the Digital Electronic System Research Team whose the main research topics are about security/safety of embedded systems, and the implementation of complex algorithms with physical constraints. He has authored more than 200 scientific publications, holds 20 patents, and co-founded the company Secure-IC in 2010.

Naofumi Homma (Senior Member, IEEE) received the M.S. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 1999 and 2001, respectively. He is currently a Professor with the Research Institute of Electrical Communication, Tohoku University. His research interests include hardware security, computer arithmetic, EDA methodology, and cryptographic implementation. He is a member of Advisory Board for Cryptographic Technology, Japan.