Testing Security Equivalence in the Random Probing Model

Anna Guinet, Carina Graw, Lukas Koletzko, Jan Richter-Brockmann, Holger Dette and Tim Güneysu

Ruhr University Bochum, Bochum, Germany
firstname.lastname@rub.de

Abstract.
The random probing model is a theoretical model that abstracts the physical leakage of an embedded device running a cryptographic scheme with more realistic assumptions compared to the threshold probing model. It assumes that the wires of the target device leak their assigned values with probability p, and the said values may reveal information about secret data, which could lead to a security violation. From that, we can compute the probability ϵ that a side-channel adversary may learn secret data from any random combination of wires as a function of the number of wire combinations that breaches security with rate p. This model is used to evaluate the security of masked cryptographic implementations, or simply named circuits; and the research community has been focusing so far on approximating or estimating the probability ϵ for one circuit. Yet, no proposition has been made to quickly compare the probability ϵ of different circuits, e.g., a circuit and its optimized version. In this context, we present two statistical tests to make decisions about the level of security in the random probing model: the equivalence test compares the security of two circuits in terms of ϵ's and the superiority test decides whether the undetermined ϵ of one circuit falls below a security threshold ϵ_0, both with quantified uncertainty about the computations of the probabilities ϵ's. The validity of these tests is proven mathematically sound and verified via empirical studies on small masked S-boxes.

Keywords: random probing model · equivalence testing · superiority testing · masking

1 Introduction

Physical implementation attacks are a well-known threat to cryptographic algorithms implemented on embedded devices [Koc96, KJJ99, GMO01]. Among these threats, power side-channel analysis is a powerful and popular attack to extract secret data from a target device [RD20]. Since its first introduction in 1999 [KJJ99], researchers have presented several approaches to design and implement efficient countermeasures. Presently, *masking* [CJRR99, GP99] is a popular technique to hamper the extraction of sensitive data by exploiting dependencies in the power consumption of the target device. However, due to its complexity, the application of masking to cryptographic algorithms can be challenging. Consequently, the community has proposed various security models to abstract the physical behavior of an Integrated Circuit (IC), thereby facilitating the design process of efficient and secure countermeasures.

In that regard, in 2003, Ishai, Sahai, and Wagner presented the d-probing model [ISW03] as a high-level abstraction of the said physical behavior. However, it does not account for critical effects like glitches, transitions, and couplings occurring in hardware implementations. Consequently, Faust et *al.* refined it and presented the robust d-probing

model in 2018 [FGM+18]. Shortly thereafter, Belaïd et *al.* [BCP+20] explicitly argued that the d-probing model and its extension does not capture repeated manipulation of variables exploited by horizontal side-channel attacks [BCPZ16]. As a result, they proposed a framework to verify security and formalized properties in the *random probing model* [ISW03, Ajt11, ADF16, AIS18]. The authors of [DDF14] proved this model to be an intermediate step in the reduction between the abstract d-probing model and the more realistic noisy leakage model [CJRR99, PR13]. In short, the random probing model assumes a leaking probability p for each individual wire in a target circuit and calculates the overall probability ϵ that an adversary may learn information about the secret data from the values leaked by the wires.

However, the application of the ϵ-random probing model can be extremely tedious. Researchers from industry and academia presented several computer-aided verification tools, e.g., VRAPS [BCP+20], STRAPS [CFOS21], IronMask [BMRT22], IronMask+ [BFG+24a], VERICA+ [BFG+24a], INDIANA [BFG+25], to compute the resulting probability ϵ. These tools are able to approximate or estimate the probability ϵ of one circuit with deterministic or probabilistic methods (as done for STRAPS and INDIANA in the latter case), usually by providing its lower and/or upper bound due to practical limitations, often determined under significant computational efforts. As for now, the research community has been focusing on assessing the random probing security for one circuit, while improving the accuracy of verification tools and reducing computational complexity. However, a given circuit is rarely left unchanged, it is frequently modified to reach a targeted security level. From a practical perspective, every modification of the original circuit requires re-running the tools over the full circuit to re-calculate the probability ϵ, repeatedly incurring the same computational costs. To avoid high computational costs in this context, it would be highly beneficial of having additional efficient methods that can subsequently check and compare the invariance (or enhancement) of a circuit modification, transformation or optimization regarding side-channel protection. We use *equivalence testing* to bridge this gap and assess the similarity of security between two circuits through the probability ϵ.

Motivation. To further motivate the use of equivalence tests, let C be a secure circuit in the probing model; in the random probing model, we either know its probability ϵ or we can check whether a (random) combination of wires of size i does not lead to recover secret data. Moreover, we assume that each individual wire in C leaks information at a rate p. Now, we frequently modify the circuit in some way in practice, e.g., we optimize it by reducing its gate count or use of randomness, to become C'. Afterwards, we would like to assess whether C' has a similar level of security in the random probing model as the original circuit C, i.e., no security assumptions are accidentally violated. To this effect, one would need to check whether ϵ' is somewhat equivalent to ϵ, raising the question of how to perform this assessment both accurately and efficiently.

A straightforward approach would be to compare the *numerical approximations* of the probabilities ϵ and ϵ' of the said circuits obtained by the aforementioned verification tools. While this approach yields an accurate decision about similarity, it requires a lot of computational power to have precise approximations, which are usually given as lower and/or upper bounds, especially for tools that perform exhaustive search like VERICA+. Another approach is to compare *statistical approximations* of the probabilities ϵ and ϵ', provided by tools like STRAPS with Monte-Carlo sampling. Although such methods are more efficient, the resulting decision about similarity is subject to error since these methods involve random sampling procedures, which result in some variability in the obtained probabilities. In other words, a decision based on probabilistic estimations of ϵ and ϵ' always comes with a statistical uncertainty, i.e., some (random) inaccuracy due to the coin flips in the sampling procedure. Understanding this variability and quantifying its uncertainty is essential when working with such statistical decision procedures. While

STRAPS computes an upper bound for ϵ with a certain probability, there is no discussion on how this bound can be used for validating a given security level nor how uncertain (in terms of how likely being wrong) a decision based on that bound would be. More recently, Beierle et *al.* presented INDIANA which is a novel verification tool to efficiently compute the upper and lower bounds of the probability ϵ for large circuits, beyond the capabilities of VERICA$^+$ and IronMask$^+$. It computes the bounds of the probabilities ϵ by randomly selecting the wires of a circuit up to a limited number of samples. However, the random selection of wires can lead to slightly different results for each invocation as explained above and, like STRAPS, no discussion is held on how to quantify the unpredictability of the computed bounds of ϵ and on how to make decisions about the validity of the security level through ϵ. Even more important, both tools—STRAPS and INDIANA—do not consider at all a comparison between the probabilities ϵ and ϵ' of two different circuits. For the naive approach of comparing the given upper bounds of those probabilities, it does not provide any statistical guarantees on the variability of the outcome of such an evaluation.

Contributions. To address an unexplored aspect of the literature, our focus is to *make decisions* about the level of security of a given circuit based on its probability ϵ rather than to just numerically approximate or probabilistically estimate it. Specifically, in this paper, we consider two decision-problems in the random probing model:

> *Equivalence of security:* Let ϵ_1 denote the (unknown) probability that an adversary may learn secret values assigned to the wires of a circuit C_1. The latter undergoes an optimization process to become C_2 and we want to know whether its level of security is preserved. In order words, we want to decide whether $|\epsilon_1 - \epsilon_2| < \Delta$ holds true or not for a fixed threshold $\Delta > 0$.
>
> *Superiority of security:* Let ϵ denote the unknown probability of a given circuit. For a fixed threshold $\epsilon_0 > 0$, we want to decide whether $\epsilon < \epsilon_0$ holds true or not, that is, whether the security of the circuit exceeds a targeted level ϵ_0 to improve the overall side-channel protection of the circuit.

We present *statistical tests* based on a random sample of wires to decide for equivalence or superiority. Using such statistical methodology (significantly) reduces the computational time in comparison to an exhaustive evaluation by allowing some slight—but measured—variability in the decision about the probabilities ϵ's. The uncertainty of these decisions is controlled by guaranteeing bounds on the probability of a false negative and positive. Concerning the test of superiority, we show that our proposed approach provides more certainty in the final test decision compared to the approach based on statistical bounds provided by STRAPS.

In addition, our statistical equivalence test is applicable to compare circuits with different numbers of wire combinations and distinct overall leakage rate p. The latter corresponds to a particular form of the general random probing model [BFG+24a], where a circuit C_1 has a wire leakage rate p_1 and another circuit C_2 has p_2. It could correspond to performing measurements under slightly different setups.

Furthermore, we illustrate the capabilities of our two tests with the first- and second-order masked hardware implementations of two 4-bit S-boxes of the lightweight symmetric schemes SKINNY [BJK+16] and PRESENT [BKL+07]. For both S-boxes, we use the DOM [GMK16], HPC1 [CGLS21], HPC2 [CGLS21] and HPC3 [KM22] gadgets to mask their inner multiplications, resulting in different but functionally equivalent masked circuits.

To the best of our knowledge, the only work using equivalence testing in the context of Side-Channel Analysis (SCA) is ROSITA^{++} [SCS+21], which is an automated tool for detecting higher-order leakage from a masked software cryptographic implementation, through the use of equivalence testing, and then mitigating it. It tests whether the means of probability distributions of emulated side-channel measurements are equivalent up to

a threshold. However, we apply for the first time equivalence testing to compare the *probabilities* that two circuits are secure in the random probing model, while providing a far more detailed analysis of the statistical test used.

Overview. The rest of the paper is organized as follows: In Section 2, we present our notation and recall basic definitions. We revisit the foundations of the random probing model in Section 3. Section 4 constitutes the main body of the paper and introduces a test for superiority of security and a test for equivalence of security. We prove the validity of these procedures mathematically in Section A and conclude by empirically verifying our Theorems using extensive case studies on two S-boxes (4-bit SKINNY [BJK+16] and PRESENT [BKL+07] S-boxes) in Sections 5 and 6, followed by a discussion in Section 7.

2 Preliminaries

2.1 Notation

In this paper, we use an upper-case calligraphic font for sets (e.g., $\mathcal{S}$). We denote by $|\mathcal{S}|$ the cardinality of a set $\mathcal{S}$. We define $\mathcal{S}_k$ as a set of cardinality k. We write $\mathcal{P}^+(\mathcal{S})$ the power set of $\mathcal{S}$ without the empty set $\emptyset$, i.e., the set of all subsets of $\mathcal{S}$ that includes the set $\mathcal{S}$ itself, but excludes the empty set $\emptyset$. The set containing all subsets of cardinality k of $\mathcal{S}$ is $\mathcal{P}^+_k(\mathcal{S}) = \{s \subseteq \mathcal{S} \mid |s| = k,\ k > 0\}$. The sans-serif font f indicates an automated tool or a function. We denote the logic gates operating on Boolean values XOR, AND and NOT by the symbols $\oplus, \wedge, \neg$, respectively. We write $\mathbb{P}[A]$ for the probability of an event A. We differentiate between a probabilistic and a deterministic function using, respectively, the symbols $\leftarrow$ and $=$. An estimator of a parameter r is denoted by the hat symbol $\hat{r}$. Closed intervals are indicated by square brackets [] and open ones by round brackets ().

2.2 Circuit model

In this paper, we consider a circuit C as a Direct Acyclic Graph (DAG). More precisely, $C = \{\mathcal{G}, \mathcal{W}\}$ where the vertices of the graph represent the gates $g \in \mathcal{G}$ of the circuit and the edges represent the wires $w \in \mathcal{W}$ that carry Boolean values. The considered gates in $\mathcal{G}$ are logical gates to perform the AND, NOT and XOR operations on Boolean values, memory gates (registers), probabilistic gates to output a random uniform Boolean value, and constant gates to provide a deterministic Boolean value. The wires connect the individual gates.

2.3 Masking

The most popular countermeasure against SCA is *masking*, which was first introduced by Chari et *al.* [CJRR99] and by Goubin and Patarin [GP99]. The idea is to split a secret variable $x \in \mathbb{F}_2$ into a vector $(x_0, \ldots, x_d) \in (\mathbb{F}_2)^{d+1}$, $d \in \mathbb{N}$, such that $x = \bigoplus_{i=0}^{d} x_i$. The variables x_i are called *shares* of x. It guarantees that an adversary cannot recover the original secret x with the knowledge of any subset of at most d shares. The number d is the *order* of the masking and it represents the number of allowed adversarial observations, or *probes*, about the intermediate computations of a circuit in the d-threshold probing model [ISW03]. A *masked circuit* C[1] operates and manipulates the shares of the inputs while keeping intact the functionality of the unmasked circuit. Since we only consider masked circuits throughout this paper, we might simply refer to C as a circuit. To preserve the sharing, all operations that were carried on secret variables x are transformed to manipulate a set, or subset, of their shares and produce shared output values. We call

[1] A masked circuit is also formally known as a randomized circuit, see [ISW03][AIS18, Circuit Compilers].

the above individual or set of transformed operations *gadgets*, i.e., masked computations. For additional security guarantees, those gadgets might require the use of extra random values to protect inner computations carried on the shares that are called *refresh* values. The generation of the shares from a secret variable is referred to as an *encoding* procedure Enc, and reversely, the reconstitution of the original secret from the shares is known as the *decoding* function Dec. Both procedures are considered to not be part of the masked implementations.

2.4 Statistics

An *estimator* of an unknown parameter θ is a function which assigns to each sample of collected data an estimate $\hat{\theta}$ of the unknown parameter. We speak of a *consistent* estimator, if the probability that it deviates by more than an arbitrarily small margin from the true parameter approaches 0 as the sample size goes to infinity. Consistency is considered as a minimal requirement for a reasonable estimator.

A *hypothesis test* is an algorithm that outputs a decision between two competing hypotheses: a null hypothesis H_0 and its alternative H_1, about an unknown parameter θ of interest based on a sample of collected data. If the sample provides strong evidence against H_0, the test rejects the null hypothesis in favor of the alternative H_1. If there is not enough evidence to support H_1, the test does not reject the null. Since usually the sample comes from a larger population, the decision provided by the test might change for another sample: it comes with *uncertainty*. The latter is quantified by two probabilities: the probability α_I of a *false positive* (also called *type I error*), that is, of rejecting the null hypothesis H_0 even though it is true and the probability β_{II} of a *false negative* (also called *type II error*), that is, of not rejecting H_0 even though it is false. Since there is an unavoidable trade-off between false positives and false negatives, it is impossible to construct tests which minimize the probabilities of a type I error and type II error simultaneously. Therefore, it is customary to choose a so-called significance level $\alpha \in (0, 1)$ and to construct tests which satisfy two properties:

> **Level α:** If the null hypothesis H_0 is true, the probability of a false positive α_I is asymptotically bounded above by α.
>
> **Consistency:** If the null hypothesis H_0 is false, the probability of a false negative β_{II} decreases with increasing sample size and vanishes in the limit.

A test that satisfies both of these properties is called a *consistent level α test* and is considered fit for use in practice. We emphasize that there are no general bounds available for β_{II} (often, it does not only depend on the sample size but on the underlying truth). Thus, the hypothesis should always be chosen in a way such that a false negative has the less severe consequences in practice.

In addition, we use the *standard normal distribution*, or *z-distribution*, that is a well-known statistical distribution with density $\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-x^2/2\right)$. Its cumulative distribution function is denoted by $\Phi(x) = \int_{-\infty}^{x} \phi(z)dz$. The inverse, denoted by $\Phi^{-1}(x)$, is also called the *quantile function*. For $\alpha \in (0, 1)$, we denote by z_α the α quantile of the standard normal distribution: z_α is the value for which $\Phi(z_\alpha) = \alpha$, or equivalently $\Phi^{-1}(\alpha) = z_\alpha$, holds.

3 Security Model

To model the reality of physical leakage of a device, the *noisy leakage model* was originally proposed by Chari et *al.* in [CJRR99]. The overall idea is to presume that the value carried by a wire of a circuit leaks in presence of additive white Gaussian noise (assumed

to be independent). The noisy leakage model was thereafter extended and generalized by Prouff and Rivain in [PR13] where the value of each intermediate variable leaks and is perturbed by some general noise, and the difference between the probability distributions of the said value with and without noise is bounded. One drawback of this model is its low level of abstraction and the difficulty to formulate sound security arguments in it. However, it was proven possible to reduce the noisy leakage model of Prouff and Rivain to the more abstract *random probing model* in [DDF14, PGMP19]. In the latter, the adversary learns the value carried by a wire with some probability or leakage rate p, which represents the uncertainty of recovering the underlying value in presence of noise. This idea was introduced in [ISW03], along with the even more abstract d-probing model, and later refined in [Ajt11, DDF14, ADF16, AIS18, BCP+20]. The work of [DDF14, PGMP19] also prove that the random probing model is reducible to the d-probing model, where a subset of up to d wires leak with probability $p = 1$ and the remaining wires with probability $1-p = 0$. While the d-probing model is popular to design and verify masked implementations due to its simplicity to elaborate security proofs because of its high level of abstraction, it fails to capture the repeated manipulation of identical variables to retrieve secret values as shown to be a threat by *horizontal side-channel attacks* [BCPZ16]. For this reason, the research community has found a recent interest in more realistic models: The choice naturally falls on the intermediate random probing model that has a tighter reduction to the noisy leakage model than the probing model and captures horizontal attacks.

3.1 Random Probing Security

We consider that a masked circuit C has $|\mathcal{W}| \in \mathbb{N}$ number of wires. In the random probing model, an adversary can invoke a masked circuit C multiple times and probes a limited amount of those wires. The latter forms a subset that we name *wire combination* $\widetilde{\mathcal{W}}$, $\widetilde{\mathcal{W}} \subseteq \mathcal{W}$.

Definition 1 (Leakage rate)**.** Each wire w of $\widetilde{\mathcal{W}}$ is assumed to *leak*, i.e., to reveal the value carried by each of them, with probability $p \in [0, 1]$. The leakage events are mutually independent.

As a result, the probability of an adversary learning exactly and only the assigned values of a single wire combination $\widetilde{\mathcal{W}}$ with $|\widetilde{\mathcal{W}}| = i$ wires is given by $p^i\,(1-p)^{|\mathcal{W}|-i}$. Then, the question remains whether those learned values lead to a security violation or not. A security breach means that the adversary can recover (part of) the non-encoded secret inputs of C. This verification is captured by the concept of *simulation*: A simulator Sim is an algorithm reproducing the real-world view that an adversary would get by attacking C.

More formally, we have $n \in \mathbb{N}$ secret variables $x_1, \ldots, x_n$, which are treated as deterministic values. From those inputs, a sharing is created with the procedure Enc and fed as input to the masked circuit C.

Definition 2 (Simulation in the Random Probing Model [AIS18, BCP+20])**.** If given the joint probability distribution of the values leaked by the wire combination $\widetilde{\mathcal{W}}$, the description of the masked circuit C and the leakage rate p, a simulator Sim is not able to recreate the joint probability distribution of the values leaked by $\widetilde{\mathcal{W}}$ without knowing (part of) the n original inputs x_i, then $\widetilde{\mathcal{W}}$ leads to a security violation in the random probing model. This event is called *a simulation failure* for the wire combination $\widetilde{\mathcal{W}}$.

From the previous definition, we write Sim as a procedure taking the wire combination that leaks $\widetilde{\mathcal{W}}$, the description of a circuit C, and the leakage rate p as inputs. Like in [AIS18, BCP+20], we consider a simulator that returns the joint probability distribution of the values assigned to the wires of $\widetilde{\mathcal{W}}$ if the simulation is successful, and aborts if the

simulation fails, which comes from the reduction of the noisy leakage model to the random probing model [DDF14]. Then, it outputs

$$\mathsf{Sim}(\widetilde{\mathcal{W}}, C, p) = \begin{cases} \mathbb{P}[\widetilde{\mathcal{W}}] & \text{simulation success for } \widetilde{\mathcal{W}}\,, \\ \perp & \text{aborts, simulation failure for } \widetilde{\mathcal{W}}\,. \end{cases}$$

The probability that a simulation failure occurs over all the possible wire combinations of C, is represented by ϵ, which is ideally small. In other words, for all $\widetilde{\mathcal{W}}$, $\mathbb{P}[\mathsf{Sim}(\widetilde{\mathcal{W}}, C, p) = \perp] = \epsilon$. We refer to ϵ as the *simulation failure probability* for a given circuit C and leakage rate p. Complementarily, a masked circuit C is secure in the random probing model if an adversary observing the values assigned to arbitrary probed subsets of wires does not recover any original secret inputs with high probability $1 - \epsilon$.

3.2 Simulation Failure Ratios

We formulate the simulation failure probability of a masked circuit C according to the ratio of leaking wire combinations of size i ($|\widetilde{\mathcal{W}}| = i$) that lead to a simulation failure. Let $\mathcal{P}^+(\mathcal{W})$ be the power set of $\mathcal{W}$ for a given C without the empty set $\emptyset$. And let $\mathcal{P}_i^+(\mathcal{W})$ be all wire combinations of cardinality i with $i > 0$.

Definition 3 (Simulation failure ratios r_i)**.** The ratio, or probability, of leaking wire combinations of size i failing simulation is defined by

$$r_i = \frac{|\{\widetilde{\mathcal{W}} \in \mathcal{P}_i^+(\mathcal{W}) \mid \mathsf{Sim}(\widetilde{\mathcal{W}}, C, p) = \perp\}|}{|\mathcal{P}_i^+(\mathcal{W})|}\,,$$

where $|\mathcal{P}_i^+(\mathcal{W})| = \binom{|\mathcal{W}|}{i}$.

From this definition, we explicitly expressed the simulation failure probability ϵ for a given masked circuit C and leakage rate p by

$$\epsilon = \sum_{i=1}^{|\mathcal{W}|} r_i \binom{|\mathcal{W}|}{i} p^i \, (1-p)^{|\mathcal{W}|-i}\,. \tag{1}$$

In other terms, ϵ is a weighted sum of individual binomial probabilities by the ratios r_i with a truncation at the zero point as $i > 0$. Indeed, whereas an adversary can learn about no wire combination (i.e., the empty set $\emptyset$), the simulator Sim will always return a failure in this case, thus it is explicitly excluded in the model.

3.3 Estimation of the Simulation Failure Ratios

The difficulty lies in estimating the simulation failure ratios $r_1, \ldots, r_{|\mathcal{W}|}$ (Definition 3) for all possible sizes of leaking wire combinations for a given circuit. Towards that end, we propose to perform a probabilistic estimation of these ratios as opposed to a systematic computation, as done in VRAPS, IronMask, IronMask$^+$ and VERICA$^+$.

To be precise, let $\widetilde{\mathcal{W}}_i^1, \ldots, \widetilde{\mathcal{W}}_i^n$ denote n leaking wire combinations drawn uniformly at random and with replacement from the set $\mathcal{P}_i^+(\mathcal{W})$ of all subsets of $\mathcal{W}$ with size $i > 0$. Whether each of the n drawn wire combination $\widetilde{\mathcal{W}}_i$ leads to a simulation failure is represented by n independent and identically distributed Bernoulli random variables with success probability r_i. We denote a collection of $n|\mathcal{W}|$ random samples by the set $\mathcal{S}_{\widetilde{\mathcal{W}}} = \{\widetilde{\mathcal{W}}_i^1, \ldots, \widetilde{\mathcal{W}}_i^n \mid i = 1, \ldots, |\mathcal{W}|\}$.

Let us define an indicator function

$$\delta(\widetilde{\mathcal{W}}_i^j) = \begin{cases} 1 & \text{if } \mathsf{Sim}(\widetilde{\mathcal{W}}_i^j, C, p) = \perp\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{2}$$

Based on each random sample $\widetilde{\mathcal{W}}_i^1, \ldots, \widetilde{\mathcal{W}}_i^n$, a consistent estimator for the unknown ratio r_i is given by

$$\hat{r}_i = \frac{1}{n} \sum_{j=1}^{n} \delta(\widetilde{\mathcal{W}}_i^j), \tag{3}$$

i.e., the proportion of leaking wire combinations in the sample. The described method is summarized in Algorithm 1.

Building on the estimators $\hat{r}_1, ..., \hat{r}_{|\mathcal{W}|}$ obtained in this fashion, the unknown simulation failure probability ϵ can then be consistently estimated by replacing all (unknown) fractions r_i in (1) by their corresponding estimates $\hat{r}_i$ yielding

$$\hat{\epsilon} = \sum_{i=1}^{|\mathcal{W}|} \hat{r}_i \binom{|\mathcal{W}|}{i} p^i (1-p)^{|\mathcal{W}|-i}. \tag{4}$$

Contrary to the approach in STRAPS, we do not choose the sample size $\mathcal{S}_{\widetilde{\mathcal{W}}}$ in an adaptive way, resulting in a simpler method (see Section 4.1 for further details). Concerning the method adopted by INDIANA, it is a combination of a deterministic and probabilistic approach and as such, the assumption of wire combinations drawn uniformly at random does not hold in this case. Indeed, in INDIANA, the user first chooses a size of wire combinations t, with $t = 1, \ldots, |\mathcal{W}|$. Then, the tool randomly selects m wire combinations of size t where m is also defined by the user. For each of those m wire combinations $\widetilde{\mathcal{W}}$, INDIANA checks whether sub-combinations of wires in $\widetilde{\mathcal{W}}$ lead to a simulation failure or not, with the Fast Fourier-Hadamard Transform. From that, it derives the simulation failure ratios r_i with $i \leq t$. Note that if $t = |\mathcal{W}|$, there is no random sampling to perform. Because the ratios r_i are not estimated based on independent draws in this case, one would need to be careful when applying Algorithm 1 to output estimations of the said ratios. The obtained estimators are not independent anymore and therefore the following test procedures need to be adapted carefully.

Algorithm 1 *Estimation*(C, n) - Estimation of unknown simulation failure ratios

Input: Circuit C, numbers of observations n
Output: Vector of estimated ratios $\hat{r} = (\hat{r}_1, \ldots, \hat{r}_{|\mathcal{W}|})$
for $i = 1, \ldots, |\mathcal{W}|$ **do**
  Draw $\widetilde{\mathcal{W}}_i^1, \ldots, \widetilde{\mathcal{W}}_i^n$ uniformly at random and with replacement from the set $\mathcal{P}_i^+(\mathcal{W})$ of all subsets of $\mathcal{W}$ with size i.
  Compute $\hat{r}_i$ as in (3).
end for
return $\hat{r} = (\hat{r}_1, \ldots, \hat{r}_{|\mathcal{W}|})$

4 Comparing Circuit Security

This section constitutes the main results of our paper. We first introduce in Section 4.1 a test procedure that checks whether a circuit is superior than a benchmark, in the sense that its simulation failure probability is smaller than a threshold ϵ_0. Then, we present a test procedure in Section 4.2 that checks whether two circuits are equivalent to each other, meaning that their simulation failure probabilities differ at most by a threshold Δ.

4.1 Testing Superiority of Simulation Failure Probabilities

In this section, we are interested in deciding whether the *unknown* simulation failure probability ϵ of a circuit C of interest is smaller than a fixed security level ϵ_0, i.e., we aim

to decide whether the hypothesis $\epsilon < \epsilon_0$ holds true or not.

This question arises naturally when optimizing or modifying the implementation of a masked circuit (e.g., to reduce the area or the latency or simply the probability ϵ), and we want to compare its resulting unknown simulation failing probability ϵ against a known one ϵ_0. The latter can either represent a security threshold to not exceed or a simple reference probability to compare several circuits to, in the random probing model.

We can formalize this decision problem in terms of hypotheses testing by the following one-sided hypotheses about the unknown security parameter ϵ of the modified circuit C:

$$H_0 : \epsilon \geq \epsilon_0 \quad \text{vs.} \quad H_1 : \epsilon < \epsilon_0. \tag{5}$$

Note that here only ϵ is assumed unknown and ϵ_0 is a fixed and known number. Such decision problems are well studied, see, for example, [DH04, CB24].

Let $\hat{\epsilon}$ be the estimator of the unknown simulation failure probability ϵ of C defined in (4). We define an estimator $\hat{\sigma}^2$ for the variance of $\hat{\epsilon}$ by

$$\hat{\sigma}^2 = \sum_{i=1}^{|\mathcal{W}|} \hat{r}_i(1-\hat{r}_i)\binom{|\mathcal{W}|}{i}^2 p^{2i}(1-p)^{2(|\mathcal{W}|-i)}. \tag{6}$$

Employing the estimators $\hat{\epsilon}$ and $\hat{\sigma}^2$, we use the following test for the hypotheses (5).

Test 1 (Superiority Test)**.** *Based on the collected sample $\mathcal{S}_{\widetilde{\mathcal{W}}}$ drawn uniformly at random and with replacement from the set $\mathcal{P}_i^+(\mathcal{W})$ of all subsets of wires $\widetilde{\mathcal{W}}$ of size $i > 0$, we reject the null hypothesis in (5) whenever*

$$\hat{\epsilon} < \epsilon_0 + \frac{\hat{\sigma} z_\alpha}{\sqrt{n}},$$

where z_α denotes the α-quantile of the standard normal distribution, $\alpha \in (0,1)$ is the pre-specified significance level of the test and n denotes the number of random wire combinations drawn for the estimation of each simulation failure ratio r_i.

The term $\hat{\sigma} z_\alpha / \sqrt{n}$ represents the margin of error that takes into account sampling variability, i.e., how sufficiently below the threshold ϵ_0 the estimated $\hat{\epsilon}$ should be to be called statistically significant. The described procedure is summarized in Algorithm 2.

Algorithm 2 Superiority Test for Hypotheses (5)

Input: Circuit C with leakage rate p, threshold ϵ_0, significance level α and number of observations n.
Output: Test decision: 'reject $H_0 : \epsilon \geq \epsilon_0$' or 'not reject $H_0 : \epsilon \geq \epsilon_0$'.

Estimate $\hat{r} \leftarrow \mathit{Estimation}(C, n)$ from Algorithm 1.
Compute $\hat{\epsilon}$ as in (4).
Compute $\hat{\sigma}^2$ as in (6).
if $\hat{\epsilon} < \epsilon_0 + \frac{\hat{\sigma} z_\alpha}{\sqrt{n}}$ **then**
  return reject $H_0 : \epsilon \geq \epsilon_0$
else
  return not reject $H_0 : \epsilon \geq \epsilon_0$
end if

Definition A.1 in Section A shows that Test 1 is a consistent, asymptotic level α test for the hypotheses (5), meaning that the probability of a false positive is bounded by α

and that the probability of a type II error converges to zero with increasing sample size. This theoretically justifies the use of this test in practice.

The computational time of Algorithm 2 is $O(n|\mathcal{W}|)$ while the computational time of a complete exhaustive search is $O\left(\sum_{i=1}^{|\mathcal{W}|}\binom{|\mathcal{W}|}{i} = 2^{|\mathcal{W}|}\right)$.

Choice of n. For a rule of thumb on how to choose the number of sampled wires n, we take a look at the probability of a false negative, i.e., to not reject the null hypothesis where we do not claim the desired security level despite the circuit is considered sufficiently secure according to the reference ϵ_0. The probability of a type II error, for a fixed level α, is given by $\beta_{II} = \mathbb{P}(\hat{\epsilon} \geq \epsilon_0 + \hat{\sigma} z_\alpha/\sqrt{n}) = 1 - \mathbb{P}(\sqrt{n}(\hat{\epsilon} - \epsilon)/\hat{\sigma} < \sqrt{n}(\epsilon_0 - \epsilon)/\hat{\sigma} + z_\alpha)$, where ϵ denotes the true (unknown) simulation failure probability. For sufficiently large n, $\sqrt{n}(\hat{\epsilon} - \epsilon)/\hat{\sigma}$ follows a standard normal distribution by the Central Limit Theorem. As a result, the probability of a type II error is bounded by $\beta_{II} \leq 1 - \Phi(z_\alpha + 2\sqrt{n}(\epsilon_0 - \epsilon))$, since the variance σ^2 is bounded by $1/4$.[2] Let us now have β an upper bound such that $\beta_{II} \leq \beta$. It implies that, if we want to correctly classify circuits as secure with respect to some threshold ϵ_0 with probability at least $1 - \beta$, while having a type I error of at most α at the same time, it is sufficient to ensure that $1 - \Phi(z_\alpha + 2\sqrt{n}(\epsilon_0 - \epsilon)) \leq \beta$. We now write the $1 - \beta$ quantile of the standard normal distribution by $z_{1-\beta} = \Phi^{-1}(1 - \beta)$. By rearranging the terms in the latter inequality, we see that it holds true if $n \geq (z_{1-\beta} - z_\alpha/2(\epsilon_0 - \epsilon))^2$. In other words, if we draw at least that many wire combinations of each size $i = 1, \ldots, |\mathcal{W}|$, the test correctly classifies secure circuits as secure with a probability of at least $1 - \beta$. Thus, we obtain the following proposition concerning the minimum number of samples n_0 for n to be sufficiently large to achieve the chosen error rates α, β.

Proposition 1. *Let $n \geq n_0$. The value of n_0 depends on the difference $\epsilon_0 - \epsilon$ such that $n_0 = \left(\frac{z_{1-\beta} - z_\alpha}{2(\epsilon_0 - \epsilon)}\right)^2$. Then, the probability of a false negative for Algorithm 2 is bounded by β. In other words, for $n \geq n_0$, it holds that $\beta_{II} \leq \beta$.*

Fig. 1 illustrates how the required number of samples n_0 varies as a function of $\epsilon_0 - \epsilon$ and β, for $\alpha = 5\%$. For clarification, this figure does not depend on p since it only influences the value of ϵ and the plot varies over different values of $\epsilon_0 - \epsilon$, where ϵ_0 is fixed. Furthermore, the difference $\epsilon_0 - \epsilon$ is unknown in practice. It can be replaced by a granularity η such that Proposition 1 holds for $n \geq n_0$ whenever $\epsilon_0 \geq \eta + \epsilon$.

As a result, we obtain a rule of thumb—depending on the desired accuracy in terms of the error probabilities α, β and the granularity η—for when an exhaustive search is to be preferred over the statistical approach, stated below as a guideline.

Rule of thumb 1. *Whenever the required number of samples n_0 for the desired accuracy is larger than the number of wire combinations of size i, the exhaustive computation of the ratio r_i is faster than their estimations. In other words, we propose to compute the corresponding ratio r_i for wire combinations of size i via an exhaustive search whenever $n_0 \geq \binom{|\mathcal{W}|}{i}$.*

Moreover, Jahandideh et *al.* point out in [JMB24] that the Monte Carlo approach (or any randomt reliably estimate simulation failure ratios r_i that are smaller than $1/n$. Aar behavior is observed when the true simulation failure probability ϵ is close to the threshold ϵ_0: Circuits can not reliably be labeled as secure if $\epsilon_0 - \epsilon$ is smaller than $1/\sqrt{n}$. Note that, if $\epsilon_0 - \epsilon = 1/\sqrt{n}$, then from Proposition 1, we have $n_0 = n\,(z_{1-\beta} - z_\alpha)^2/4$.

[2] Note that $r_i(1 - r_i)$ in (6) is bounded by $1/4$ because $r_i \in [0, 1]$. Since the other terms in the sum are bounded by a probability $\binom{|\mathcal{W}|}{i}^2 p^{2i}(1-p)^{2(|\mathcal{W}|-i)} \leq \binom{|\mathcal{W}|}{i} p^i (1-p)^{(|\mathcal{W}|-i)}$, the remaining sum is bounded by one.

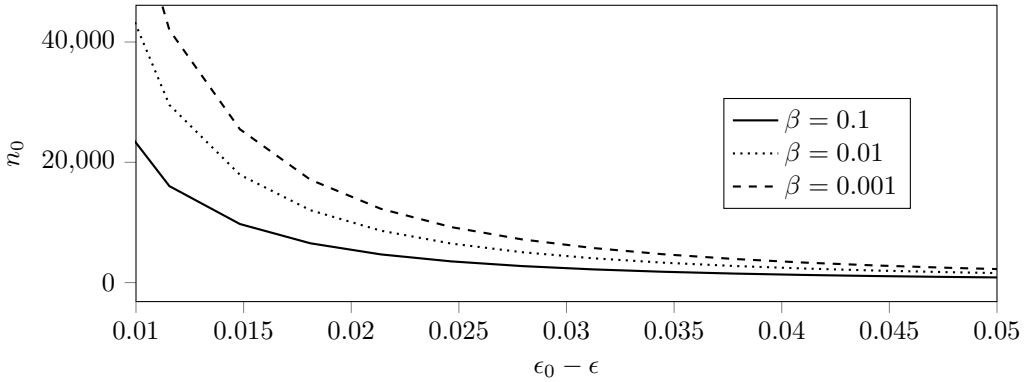

Figure 1: Required number of samples n_0 such that the type II error is bounded by β according to the unknown difference $\epsilon_0 - \epsilon$ for $\alpha = 5\%$.

Therefore, $n \geq n_0$ does not hold true for arbitrary choices of β and α since it now requires $(z_{1-\beta} - z_\alpha) \leq 2$. For higher accuracy, a larger number of samples is required and our provided guideline indicates for which scenarios a random sampling approach is to be preferred over an exhaustive search.

Remark 1. *We choose the quantiles z_α of a z-distribution in Test 1, leading to a so-called z-test. When theses quantiles are replaced by the ones of a t-distribution with $n-1$ degrees of freedom, we obtain a t-test. Note ththe latter is different from Welch's t-test in the Test Vector Leakage Assessment (TVLA) methodology [GGJR+11] since it checks whether mean values of probability distributions are equal or not, whereas we compare probability distributions (simulation failure probabilities). Moreover, for large n, the t-distribution with $n-1$ degrees of freedom is (approximately) the same as the standard normal distribution. As this approximation is rather precise for $n \geq 25$, we only consider the z-test, but all results can easily be extended to the t-test.*

Comparison to STRAPS. While the estimation of the ratios r_i (see Algorithm 1) is rather similar to STRAPS [CFOS21], our test is based on different confidence intervals than theirs. Indeed, we give a probabilistic upper bound for ϵ at level $1-\alpha$ using the variance $\hat{\sigma}^2$ of the estimator $\hat{\epsilon}$ in (6). More precisely, we have $\epsilon < \hat{\epsilon} + \frac{\hat{\sigma}\, z_{1-\alpha}}{\sqrt{n}}$ with probability $1-\alpha$. In contrast, the authors of [CFOS21] compute a probabilistic upper bound for ϵ at level $1-\alpha$ based on corresponding bounds for the ratios r_i. Each probabilistic upper bound that is called r_i^U, is based on $\hat{r}_i$ and satisfies $r_i < r_i^U$ with probability $1 - \frac{\alpha}{|\mathcal{W}|+1}$. The probabilistic upper bound for ϵ is then given by $\epsilon^U = \sum_{i=1}^{|\mathcal{W}|} r_i^U \binom{|\mathcal{W}|}{i} p^i (1-p)^{|\mathcal{W}|-i}$. The division by $|\mathcal{W}|+1$ is called a Bonferroni correction [Dun61, BA95]. While this ensures that the obtained bound is an upper bound for ϵ with probability $1-\alpha$, such Bonferroni corrections do not yield tight confidence intervals. This is illustrated in Fig. 5 in Section 6.2.

Although such loose upper bounds can also be used for testing (reject, if $\epsilon^U < \epsilon_0$), it is well-known that the corresponding decision often suffers from a high Type II error (see, for example, [Mor03]). In other words, STRAPS test will not label secure circuits as secure, which is not desirable in practice. This is demonstrated in Fig. 6 in Section 6.2.

In addition, Proposition 1 gives a bound on the Type II error of our procedure and therefore a rule of thumb on how to choose the sample size. To the best of our knowledge, such bounds are not available for STRAPS and might be hard to compute since r_i^U depends implicitly on $\hat{r}_i$ through the regularized incomplete beta function.

Additionally, [CFOS21] proposed an hybrid algorithm, where—if suitable—some ratios r_i are exhaustively computed. The corresponding lower bound is then set to $r_i^U = r_i$. Our

procedure can be adapted to this scenario as well, by adapting the variance estimator $\hat{\sigma}^2$ in (6) accordingly (note that the variance of a deterministic value is zero).

4.2 Testing Equivalence of Simulation Failure Probabilities

In this section, we are interested in deciding whether the *unknown* simulation failing probabilities ϵ_1 and ϵ_2 corresponding to two circuits C_1 and C_2, respectively, are similar. This decision process is easily adaptable for one known and one unknown simulation failing probability.

The main motivation of this study is to assess which design changes do not significantly impact the security level of a given circuit in the random probing model, thereby guiding designers in their choices. In other words, we assess how close two security levels are, which cannot be accomplished using superiority tests. For example, consider a reference circuit C_1 that we know secure in the probing model but we may not know its corresponding simulation failing probability ϵ_1. If we optimize C_1 with respect to some metric (e.g., area, latency, randomness in hardware) to obtain a new circuit C_2, a natural question is whether we obtain a similar level of security in the random probing model compared to the original version C_1. A positive answer would indicate that the performed modifications on the original circuit do not violate its security. Therefore, our objective is to quickly decide whether those two circuits offer an equivalence sense of security in the random probing model. However, this test is not restricted to this use case; the two circuits to be compared do not have to be transformations of each other. We could also be interested in quickly deciding whether two unrelated circuits provide a similar level of security in the random probing model, for example, to compare two different gadgets masking the same operation. In addition, the reference circuit does not have to be secure for having meaningful similarity assessments. One could check whether design changes have preserved the insecurity of the circuit in order to showcase their ineffectiveness.

To formalize the decision problem, we denote by ϵ_1 and ϵ_2 the corresponding (unknown) simulation failing probabilities defined by

$$\epsilon_1 = \sum_{i=1}^{|\mathcal{W}_1|} r_i \binom{|\mathcal{W}_1|}{i} p_1^i (1-p_1)^{|\mathcal{W}_1|-i} \quad \text{and} \quad \epsilon_2 = \sum_{i=1}^{|\mathcal{W}_2|} s_i \binom{|\mathcal{W}_2|}{i} p_2^i (1-p_2)^{|\mathcal{W}_2|-i}$$

with unknown simulation failing ratios $r_1, ..., r_{|\mathcal{W}_1|}$ and $s_1, ..., s_{|\mathcal{W}_2|}$ and possibly distinct leakage rates p_1 and p_2 of the circuits C_1 and C_2, respectively. Indeed, for this decision problem, we are able to compare circuits with different leakage rates; it could correspond to the assessment of the wire leakages for each circuit under different noise variations. Or, it can be viewed as a particular form of the general random probing model [BFG+24a] that allows different leakage rates, but the leakage rate of each circuit is kept constant (note that a circuit could simply be a gadget).

Definition 4 (Similarity/Equivalence)**.** Let $\Delta > 0$ denote an equivalence (or similarity) threshold that is pre-specified in practice. Two probabilities ϵ_1 and ϵ_2 are *similar* or *equivalent* if and only if $|\epsilon_1 - \epsilon_2| < \Delta$ holds true.

Our goal is to decide whether the two probabilities ϵ_1 and ϵ_2 are similar as per this definition, that is, we want to assess whether the two circuits have comparable security in the random probing model. Formally, we can describe this problem as aiming to test the following hypotheses:

$$H_0 : |\epsilon_1 - \epsilon_2| \geq \Delta \quad \text{vs.} \quad H_1 : |\epsilon_1 - \epsilon_2| < \Delta. \tag{7}$$

This problem is well-studied; notable works include, but are not limited to, [Wel02, LRY+05, RF04, BDK24]. We propose to test the hypotheses (7) by using the popular Two

One Side t-test (TOST), which is based on the intersection-union principle (see [Sch87] for further details). Noting that H_0 in (7) holds when either $\epsilon_1 - \epsilon_2 \geq \Delta$ or $\epsilon_1 - \epsilon_2 \leq -\Delta$ holds, TOST consists in testing the following two hypothesis pairs simultaneously (each at significance level α):

$$\begin{aligned} H_{a,0} : \epsilon_1 - \epsilon_2 \geq \Delta \quad &\text{vs.} \quad H_{a,1} : \epsilon_1 - \epsilon_2 < \Delta \\ H_{b,0} : \epsilon_1 - \epsilon_2 \leq -\Delta \quad &\text{vs.} \quad H_{b,1} : \epsilon_1 - \epsilon_2 > -\Delta \end{aligned}$$

and the null hypothesis H_0 in (7) is rejected at significance level α if both $H_{a,0}$ and $H_{b,0}$ are rejected. The two null hypotheses $H_{a,0}$ and $H_{b,0}$ can be tested analogous to (1).

Let $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ be the variance estimators as in (6) for circuit C_1 and C_2, respectively.

Test 2 (Test for Equivalence)**.** *Based on the collected samples $\mathcal{S}_{\widetilde{\mathcal{W}}_1}$ and $\mathcal{S}_{\widetilde{\mathcal{W}}_2}$ drawn uniformly at random and with replacement from the set of all subsets of wires $\mathcal{W}_1$ of the circuit C_1 and of all subsets of wires $\mathcal{W}_2$ of C_2, respectively, of size $i > 0$, we reject the null hypothesis in (7), whenever*

$$\hat{\epsilon}_1 - \hat{\epsilon}_2 \leq \Delta + \frac{\hat{\sigma}_{dif} z_\alpha}{\sqrt{n}} \text{ and } \hat{\epsilon}_1 - \hat{\epsilon}_2 \geq -\Delta + \frac{\hat{\sigma}_{dif} z_{1-\alpha}}{\sqrt{n}}$$

with $\hat{\sigma}_{dif}^2 = \hat{\sigma}_1^2 + \hat{\sigma}_2^2$.

The pseudo-code for this test is given in Algorithm 3.

Algorithm 3 TOST - Equivalence Test for Hypotheses (7)

Input: Circuit C_1 with leakage rate p_1 and circuit C_2 with leakage rate p_2, tolerated difference Δ, significance level α, number of observations n
Output: Test decision 'reject $H_0 : |\epsilon_1 - \epsilon_2| \geq \Delta$' or 'not reject $H_0 : |\epsilon_1 - \epsilon_2| \geq \Delta$'

Estimate $\hat{r} \leftarrow \mathit{Estimation}(C_1, n)$, $\hat{s} \leftarrow \mathit{Estimation}(C_2, n)$ from Algorithm 1.
Compute $\hat{\epsilon}_1$ and $\hat{\epsilon}_2$ based on $\hat{r}, p_1$ and $\hat{s}, p_2$, resp., as in (4).
Compute $\hat{\Delta} = \hat{\epsilon}_1 - \hat{\epsilon}_2$.
Compute $\hat{\sigma}_{dif}^2 = \hat{\sigma}_1^2 + \hat{\sigma}_2^2$, with $\hat{\sigma}_1$ and $\hat{\sigma}_2$ based on $\hat{r}, p_1$ and $\hat{s}, p_2$, resp., as in (6).
if $\hat{\Delta} \leq \Delta + \hat{\sigma}_{dif} z_\alpha n^{-1/2}$ **and** $\hat{\Delta} \geq -\Delta + \hat{\sigma}_{dif} z_{1-\alpha} n^{-1/2}$ **then**
  return reject $H_0 : |\epsilon_1 - \epsilon_2| \geq \Delta$
else
  return not reject $H_0 : |\epsilon_1 - \epsilon_2| \geq \Delta$
end if

Theorem A.2 in the Appendix shows that Test 2 is a consistent, asymptotic level α test for the equivalence hypotheses (7), meaning that the probability of a false positive is bounded by α and that the probability of a type II error converges to zero with increasing sample size. This theoretically justifies the use of this test in practice.

Note that the confidence intervals from STRAPS are unsuitable for equivalence testing, since they are given for a single simulation failure probability ϵ_1 or ϵ_2, whereas a confidence interval for $|\epsilon_1 - \epsilon_2|$ is needed.

Remark 2. *We have presented a test for investigating the equivalence of two simulation failure probabilities in the (more complicated) case where both values are unknown. If, however, one of the two simulation failure probabilities is known, it is more reasonable to instead test the simpler hypothesis pair $H_0 : |\epsilon_0 - \epsilon| \geq \Delta$ vs. $H_1 : |\epsilon_0 - \epsilon| < \Delta$, where ϵ_0 denotes the known and ϵ the unknown simulation failure probability. We remark that a simple adaption of Algorithm 3 yields a valid test for those hypotheses. We omit this discussion for the sake of brevity.*

Equivalence vs. Superiority Testing. We emphasize that the superiority test $H_0 : \epsilon \geq \epsilon_0$ vs. $H_1 : \epsilon < \epsilon_0$ from (5) cannot be applied to the equivalence problem in (7). Firstly, the superiority problem compares a single unknown probability ϵ against a fixed reference value ϵ_0, whereas the equivalence problem compares two unknown probabilities ϵ_1 and ϵ_2. For the latter, even if one probability is fixed and the other is left unknown as shown in Remark 2, the hypotheses for superiority in (5) remains different than the ones for equivalence, which lead to a different conclusion regarding the outcome of the test. Therefore, Test 1 is not directly applicable to the current setting (some adjustments are first needed). Indeed, a superiority test can only detect whether a modified circuit is strictly superior to the original, i.e., its simulation failure probability is strictly lower than the targeted one. If this is the case, security is certainly preserved by this conservative approach. However, if we optimize a circuit to achieve greatly improved performance on a non-security metric, and its simulation failure probability is slightly higher or lower than the original one (say $\epsilon_2 = \epsilon_1 \pm \Delta_\epsilon$), the superiority test would only say the optimized circuit has not improved on the benchmark ϵ_1 if the probability ϵ_2 is slightly higher, and does not provide any information at all about how close the new probability is to the original one. In contrast, an equivalence test would detect them as similar given a large enough sample size, because the security difference is deemed marginal ($\Delta > \Delta_\epsilon$), allowing us to proceed with the new and significantly improved circuit.

5 Evaluated Designs

To demonstrate our proposed test procedures, we choose two S-boxes, which are symmetric cryptographic nonlinear components, for our test cases: the 4-bit SKINNY [BJK+16] and PRESENT [BKL+07] S-boxes. While operating on the same number of bits, the hardware implementations and possible optimizations of those two S-boxes are very different, allowing to have various values of simulation failure probabilities ϵ. The latter are used to prove that our tests provide reliable results under their underlying assumptions, which is the aim of this work. We do not seek to compare the circuits (or gadgets) themselves.

We use first-order ($d = 1$) and second-order ($d = 2$) masked hardware implementations for both S-boxes. Following the principles of Private Circuits from [ISW03], we consider the three following operations in $\mathbb{F}_2$ for our circuits: the addition, the multiplication and the negation, which are respectively encoded by the XOR, the AND and the NOT gate. We only consider 2 fan-in XOR and AND gates (i.e., manipulating two input variables). Those operations are then replaced by gadgets; the secure masking of the addition and the negation are a trivial exercise. Let x, y be two secret variables in $\mathbb{F}_2$, their d-th order sharing are $(x_0, \ldots, x_d)$ and $(y_0, \ldots, y_d)$, respectively. Then, we perform securely a share-wise masked addition in $\mathbb{F}_2$: $x_i \oplus y_i$ for $i \in \{0, \ldots, d\}$, such that $\sum_i x_i \oplus y_i = x \oplus y$. In a similar fashion, we mask the negation of a secret variable x by assigning the value $x_0 \oplus 1$ to the first share x_0 and keeping the other shares unchanged, such that the sum of the new shares equals $x \oplus 1$. Multiplication gadgets are less trivial to design. To the best of our knowledge, we do not know of any hardware implementation of multiplication gadgets tailored for the random probing model. Therefore, we choose four implementations from the literature that are secure in the more abstract threshold probing model: DOM [GMK16], HPC1 [CGLS21], HPC2 [CGLS21] and HPC3 [KM22]. Thus, for each S-box, we provide masked circuits with different multiplication gadgets, where the latter are briefly compared in Table 1. For a security comparison of those gadgets, see [KSM20, KM22] or the supplementary material of this work. Concerning other (hardware design) metrics, such as clock cycle latency or critical path delay, we refer to [KM22, CSV24, BLH+25]. The same implementation masked with any of those gadgets results in functionally equivalent designs for one S-box and one masking order, but with different number of wires, and as such, is a ideal test case to show the validity of our equivalence test (see Test 2).

Table 1: Description of the number of gates, wires (SCA positions) and random bits for each considered multiplication gadget in HADES according to VERICA$^+$. We give the true value of a lower bound $\epsilon_{\min}$ from (8) as reported by VERICA$^+$ without the binomial tree optimization and an upper bound $\epsilon_{\max}$, both for $t = 4$ (see Section 6.1).

	Order	**Circuit**		**Random.**	**Model**	
Gadget (only)	d	gates $\lvert\mathcal{G}\rvert$	wires $\lvert\mathcal{W}\rvert$	bits	p	$\epsilon_{\min}/\epsilon_{\max}$
DOM [GMK16]	1	12	15	1	0.01	3.016e-3/3.016e-3
	2	30	36	3	0.01	3.55e-4/3.84e-4
HPC1 [CGLS21]	1	20	24	2	0.01	4.820e-3/4.823e-3
	2	43	51	5	0.01	5.32e-4/6.92e-4
HPC2 [CGLS21]	1	27	30	1	0.01	1.310e-2/1.312e-2
	2	69	75	3	0.01	4.397e-3/5.363e-3
HPC3 [KM22]	1	24	28	2	0.01	6.792e-3/6.800e-3
	2	60	69	6	0.01	1.084e-3/1.744e-3

All designs were automatically generated with HADES from Buschkowski et *al.* [BLH$^+$25], which is an automated tool for generating d-order masked hardware cryptographic implementations using the NanGate 45nm library.

5.1 4-bit PRESENT S-box

PRESENT [BKL$^+$07] is a popular lightweight block cipher introduced by Bogdanov et *al.* with a substitution-permutation network that minimizes the required area of its hardware implementation, while providing strong security guarantees against cryptanalysis attacks.

Let (x_3, x_2, x_1, x_0) and (y_3, y_2, y_1, y_0) denote, respectively, the 4-bit input and 4-bit output, using the most significant bit first ordering, of the PRESENT S-box whose lookup table is given in Table 6 in Section B.1. From that, the Algebraic Normal Forms (ANFs) to describe the S-box are

$$
\begin{aligned}
y_3 &= x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \oplus x_0 \oplus x_1x_2 \oplus x_1 \oplus x_3 \oplus 1\,,\\
y_2 &= x_0x_1x_3 \oplus x_0x_1 \oplus x_0x_2x_3 \oplus x_0x_3 \oplus x_1x_3 \oplus x_2 \oplus x_3 \oplus 1\,,\\
y_1 &= x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \oplus x_1x_3 \oplus x_1 \oplus x_2x_3 \oplus x_3\,,\\
y_0 &= x_0 \oplus x_1x_2 \oplus x_2 \oplus x_3\,.
\end{aligned}
$$

We consider a first PRESENT S-box hardware implementation PD-I from [RV23] where the logic has been adapted to use only NOT, XOR and AND gates. The adapted sequential circuit is described in Fig. 2 and is composed of 10 XOR, 12 AND and 6 NOT gates. In the following, we simply refer to this implementation as the PRESENT S-box circuit.

For comparison, we choose an optimized hardware implementation for masking that minimizes the number of multiplications, which are the costlier gadgets since they require to draw uniformly random Boolean values (refreshing). We use the hardware implementation with latency optimization from Cassiers et *al.* in [CGLS21]. The circuit is detailed in Fig. 3 and contains 16 XOR, 4 AND and 5 NOT gates. In the remaining of this paper, we refer to this implementation as the PRESENT S-box *optimized* circuit. The number of performed multiplications between this optimized circuit compared to the previous one has been reduced by a third.

5.2 4-bit SKINNY S-box

The SKINNY family was presented by Beierle et *al.* in [BJK$^+$16] as a family of lightweight tweakable block ciphers with a very compact S-box, offering as well strong security

$$
\begin{array}{llll}
n_0 = \neg x_0 & n_3 = \neg m_1 & a_1 = n_0 \oplus m_4 & m_9 = n_0 \wedge a_6 \\
n_1 = \neg x_1 & m_4 = x_3 \wedge n_3 & a_2 = m_6 \oplus a_1 & n_5 = \neg m_9 \\
n_2 = \neg x_2 & m_5 = x_3 \wedge x_0 & a_3 = m_1 \oplus m_7 & m_{10} = x_3 \wedge n_5 \\
m_0 = x_1 \wedge x_0 & n_4 = \neg m_5 & m_8 = x_3 \wedge a_3 & m_{11} = x_1 \wedge n_3 \\
m_1 = x_2 \wedge x_0 & a_0 = m_3 \oplus n_4 & a_4 = n_2 \oplus m_0 & a_7 = m_{10} \oplus m_{11} \\
m_2 = x_2 \wedge n_1 & m_6 = x_1 \wedge a_0 & a_5 = m_8 \oplus a_4 & a_8 = x_3 \oplus x_0 \\
m_3 = x_2 \wedge n_0 & m_7 = n_1 \wedge n_0 & a_6 = x_1 \oplus x_2 & a_9 = a_8 \oplus m_2
\end{array}
\qquad
\begin{array}{l}
y_3 = a_2 \\
y_2 = a_5 \\
y_1 = a_7 \\
y_0 = a_9
\end{array}
$$

Figure 2: Circuit description of 4-bit PRESENT S-box adapted from PD-I in [RV23, Figure 2] to substitute logical OR operations, with most significant bit first ordering, referred to as the non-optimized implementation.

$$
\begin{array}{llll}
a_0 = x_3 \oplus x_0 & a_3 = x_1 \oplus x_0 & a_7 = m_0 \oplus a_4 & m_3 = a_6 \wedge a_9 \\
a_1 = x_2 \oplus x_1 & a_4 = x_3 \oplus a_3 & a_8 = m_0 \oplus m_1 & a_{10} = a_8 \oplus m_3 \\
a_2 = a_0 \oplus a_1 & a_5 = x_3 \oplus x_1 & n_3 = \neg a_5 & a_{11} = x_0 \oplus a_{10} \\
n_0 = \neg a_1 & m_0 = n_0 \wedge n_1 & n_4 = \neg a_7 & a_{12} = a_8 \oplus a_3 \\
n_1 = \neg x_3 & m_1 = x_2 \wedge n_2 & a_9 = a_8 \oplus n_3 & a_{13} = m_2 \oplus a_{12} \\
n_2 = \neg x_1 & a_6 = a_3 \oplus x_2 & m_2 = a_2 \wedge n_4 & a_{14} = m_3 \oplus a_5
\end{array}
\qquad
\begin{array}{l}
a_{15} = m_1 \oplus a_0 \\
y_3 = a_{11} \\
y_2 = a_{13} \\
y_1 = a_{14} \\
y_0 = a_{15}
\end{array}
$$

Figure 3: Circuit description of 4-bit PRESENT S-box from [CGLS21, Figure 6b] with most significant bit first ordering, referred to as the optimized implementation.

guarantees against cryptanalysis attacks. For our application, we consider the smallest version of the S-box working over 4 bits.

Like the previous section, we keep the most significant bit first ordering to represent the 4-bit input and output of the S-box, as indicated in the lookup table in Table 7 in Section B.1. From that, the following Boolean functions describe the S-box

$$
\begin{aligned}
y_3 &= x_0 \oplus (x_3 \oplus 1) \wedge (x_2 \oplus 1)\,, \\
y_2 &= x_3 \oplus (x_2 \oplus 1) \wedge (x_1 \oplus 1)\,, \\
y_1 &= x_2 \oplus (x_1 \oplus 1) \wedge (y_3 \oplus 1)\,, \\
y_0 &= x_1 \oplus (y_3 \oplus 1) \wedge (y_2 \oplus 1)\,.
\end{aligned}
$$

The description of the SKINNY S-box is itself very compact and its implementation is straightforward from the above component functions with AND, XOR and NOT gates; no optimization is found in that regard. First, we generate with HADES a pipelined circuit that naively computes the component functions separately and therefore have three redundant (unmasked) negations: $x_2 \oplus 1$, $x_2 \oplus 1$ and $y_3 \oplus 1$. We refer to this circuit as the SKINNY S-box one. To compare two functionally and very similar circuits, we generate an optimized version that eliminates the repetitive $x_2 \oplus 1$ and $y_3 \oplus 1$ that are computed at the same stage of the pipeline,[3] resulting in a slightly more compact masked implementations. It results in a little variation of the amount of wires to be probed between the two versions, which changes the true value of the simulation failure probability ϵ. Consequently, we can observe the effect that this optimization has with our test procedures and name this circuit the SKINNY S-box *optimized* one.

[3]We experimented with HADES by forcing the pipelined circuit to compute only once the negation $x_1 \oplus 1$, but it resulted in the same number of gates/wires—or even higher depending on the multiplication gadget—than the non-optimized version.

5.3 Secure vs. Insecure Implementations

We generate secure masked hardware implementation of the SKINNY and the PRESENT S-box, with four multiplication gadgets: DOM, HPC1, HPC2 and HPC3, in the d-threshold probing model. It provides us somewhat equivalent masked designs to run our tests from Section 4. For further comparison, we introduce security flaws in each of those secure designs to break d-threshold probing security, i.e., information about an original secret x can be recovered with at most d probes; thus we can examine the resulting probability ϵ between their secure and insecure version with our test procedures.

For the insecure versions, we put the constant bit value 0 instead of random ones in the first multiplication gadget in (1) the SKINNY S-box, for the non-optimized and optimized versions, this bit reset affects $(x_3 \oplus 1) \wedge (x_2 \oplus 1)$ in the output y_3 and thus, y_1 and y_0; in (2) the non-optimized implementation of the PRESENT S-box, it changes the operation $m_0 = x_1 \wedge x_0$ affecting the output y_2; and in (3) the optimized version of the PRESENT S-box, it impacts $m_0 = n_0 \wedge n_1$, and thus the output values y_3, y_2 and y_1. The number of gates, wires, random bits for each (in)secure masked S-box implementations are reported in Table 8 for PRESENT and in Table 9 for SKINNY, in Section B.2.

All the designs were formally verified by VERICA$^+$ [BFG+24a], an extension of VERICA [RBFSG22], which is a tool to automatically formally verify SCA security for hardware circuits in the threshold probing and random probing model. The circuits generated by HADES are labeled as secure in the d-threshold probing model by VERICA$^+$ as expected; similarly, the variants with injected security flaws are reported as insecure.

6 Test Evaluation

In this section, we explicit how we created the masked implementations of the SKINNY and PRESENT S-boxes from Section 5. Then, we show that, for those considered designs, our proposed test procedure indeed distinguishes correctly between

$$H_0 : \epsilon \geq \epsilon_0 \text{ vs. } H_1 : \epsilon < \epsilon_0,$$

and

$$H_0 : |\epsilon_1 - \epsilon_2| \geq \Delta \text{ vs. } H_1 : |\epsilon_1 - \epsilon_2| < \Delta.$$

6.1 Verification Setup

We choose VERICA$^+$ as formal verification tool since it conducts exhaustive checking of the values assigned to the wires of a circuit to be correct up to a certain threshold of t wires. Hence, it provides us with the exact numbers of simulation-failing wire combinations of size $i = 1, \ldots, t$, and thus the true values of the ratios $r_1, \ldots, r_t$, for a validation of Algorithms 2 and 3. From that, VERICA$^+$ computes the true value of a lower bound of ϵ (see (1)) by the sum

$$\epsilon_{\min} = \sum_{i=1}^{t} r_i \binom{|\mathcal{W}|}{i} p^i (1-p)^{|\mathcal{W}|-i} \tag{8}$$

where we consider that any combination of more than t wires do not lead to a security violation. Likewise, we can consider that all of the said combinations lead to a security violation to compute an upper bound $\epsilon_{\max}$. Those assumptions follow the method presented by Belaïd et *al.* in [BCP+20] to derive bounds for ϵ. For practical reasons, we choose $t = 4$ since VERICA$^+$ exhaustively verifies all wire combinations up to this size, which requires time. To demonstrate that our tests performs as expected given our assumptions, we empirically check them by applying them on the lower bound $\epsilon_{\min}$. For that, we first need the exact number of simulation-failing wire combinations up to a threshold t. Hence, we

deactivate the binomial tree optimization of VERICA$^+$ presented in the extended paper version [BFG+24b]. Indeed, this deterministic optimization method provides a tighter lower bound for ϵ by deriving the simulation-failing wire combinations (potentially larger than t) containing one of size $i = 1, \ldots, t$ with the support of binomial trees, to perform a depth-first search. However, to run our random sampling procedure in Algorithm 1, we need VERICA$^+$ to perform a breath-first search up to wire combinations of t wires.

The tightness of the lower bound in (8) depends on the number of wires $|\mathcal{W}|$ as well as the leaking probability p. Note that $\forall i$, $r_i \leq 1$; therefore, the difference between ϵ in (1) and its lower bound in (8) is at most $\epsilon - \epsilon_{\min} \leq \sum_{i=t+1}^{|\mathcal{W}|} \binom{|\mathcal{W}|}{i} p^i (1-p)^{|\mathcal{W}|-i}$. Furthermore, $\binom{|\mathcal{W}|}{i} p^i (1-p)^{|\mathcal{W}|-i}$ corresponds to the probability of i successes from $|\mathcal{W}|$ Bernoulli experiments with success probability p. Therefore, $\epsilon - \epsilon_{\min}$ is bounded by the probability of having $t+1$ or more successes. This probability can be approximated by the probability that a normal distribution with mean $p|\mathcal{W}|$ and variance $|\mathcal{W}|p(1-p)$ is at least $t+1$, i.e., by $1 - \Phi((t - p|\mathcal{W}|)/\sqrt{|\mathcal{W}|p(1-p)})$. Bounding this quantity by a small constant, say 0.001, yields $\epsilon - \epsilon_{\min} \leq 0.001$. By solving $1 - \Phi((t - p|\mathcal{W}|)/\sqrt{|\mathcal{W}|p(1-p)}) = 0.001$ for t, this observation results in the following guideline.

Rule of thumb 2. *To obtain an informative lower bound, we propose to choose* $t = p|\mathcal{W}| + 3 \cdot \sqrt{(1-p) \cdot p|\mathcal{W}|}$ *to cover the relevant wire combinations The constant* 3 *approximately corresponds to the* 99.9% *quantile of the normal distribution. Choosing a higher constant value leads to an even tighter bound while choosing a lower constant loosens the bound but fastens the computation.*

From Rule of thumb 2, we can write $x^2 = p|\mathcal{W}|$, $b = 3\sqrt{1-p}$ and $c = -t$ and solve the resulting quadratic equation $x^2 + bx + c = 0$. The solutions are $x = (-b \pm \sqrt{b^2 - 4c})/2$ with $\sqrt{b^2 - 4c} \geq 0$. We now replace b, c in the solution $(-b + \sqrt{b^2 - 4c})/2$ and choose $x = \sqrt{p|\mathcal{W}|}$ to have $p|\mathcal{W}| = (1-p)(\sqrt{2.25 + t/(1-p)} - 1.5)^2$. In the latter expression, it yields that, for $p = 0.01$, $\epsilon_{\min}$ with $t = 4$ is a good approximation for circuits with less than $|\mathcal{W}| = 100$ wires. And since the number of wires $|\mathcal{W}|$ of our considered circuits is often larger than 100 (see Tables 8 and 9) and an exhaustive search takes quite some time, we recommend to use a statistical approach in this scenarios, since the latter yields accurate approximations more quickly (see Rule of thumb 1). Similarly, if we choose $p = 0.001$ in the latter expression, we found that a lower bound $\epsilon_{\min}$ with $t = 4$ is a good approximation for circuits with less than 1000 wires. Even if the lower bound $\epsilon_{\min}$ is tight, computing it by an exhaustive search still takes some time, especially for circuits with a large number of wires. As a result, a statistical approach is to be preferred in this settings as well.

In parallel, we proceed to a validation of Algorithms 2 and 3 based on the obtained simulation failing ratios $\hat{r}_1, \ldots, \hat{r}_t$ from Algorithm 1 with the numbers of simulation-failing wire combinations of size $i = 1, \ldots, t$ provided by VERICA$^+$. Indeed, with the latter, we simulate drawing wire combinations $\widetilde{\mathcal{W}}_i^j$ independent and uniformly at random over all subsets of wires in $\mathcal{W}$ and computing $\delta(\widetilde{\mathcal{W}}_i^j)$ as in (2) by drawing independently from a Bernoulli distribution with success probability r_i and estimate $\hat{r}_i$ (see (3)).

For each scenario, we simulate the considered algorithm $N_{sim} = 1000$ times and count how often the null hypothesis is rejected. Since, by Definition A.1 and Definition A.2, both procedures lead a consistent level α test as described in Section 2.4, we expect the following result: If H_0 holds true, then the test should reject at most αN_{sim} times, i.e., the rejection rate is at most α. On the other hand, if H_1 is true, the rejection rate should be close to one and increasing with the sample size n.

6.2 Testing for Superiority

We demonstrate the validity of Algorithm 2 by studying its empirical rejection rates for different values of ϵ_0 and numbers of sampled wire combinations n, and various circuits.

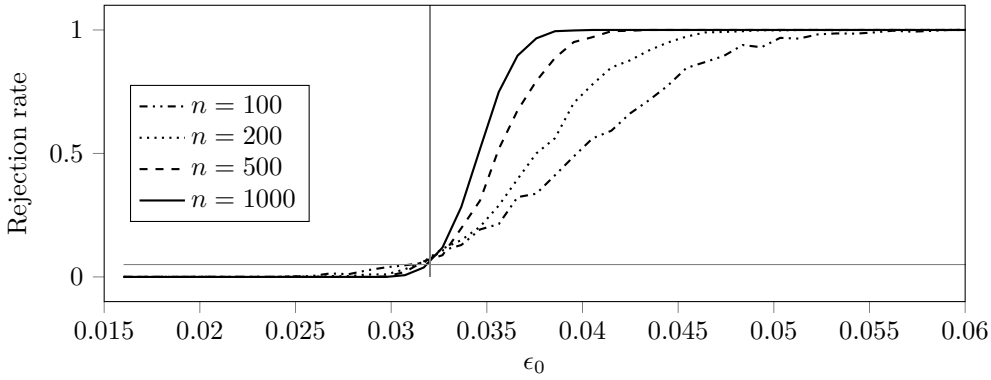

Figure 4: Rejection rate of Algorithm 2 for first-order secure SKINNY S-box with DOM gadget for $p = 0.01$ for different numbers of drawn wire combinations n. The horizontal line corresponds to the significance level of $\alpha = 5\%$ and the vertical line to $\epsilon_{\min} = 0.032$.

As an example, in Fig. 4, we display the empirical results of a first-order secure implementation of the SKINNY S-box with the DOM gadget. The horizontal line corresponds to the chosen α-level (here $\alpha = 5\%$) and the vertical line corresponds to the true lower bound $\epsilon_{\min}$ of the considered circuit. The rejection rate for $\epsilon_0 \leq \epsilon_{\min}$ corresponds to the type I error α_I, while the rejection rate for $\epsilon_0 > \epsilon_{\min}$ corresponds to a correct rejection, i.e., to $1 - \beta_{II}$. The true value of the lower bound $\epsilon_{\min}$ for $t = 4$ (see (8)) is computed by VERICA$^+$. We see that the level of $\alpha = 5\%$ is guaranteed, even for small sample sizes. Furthermore, the type II error decreases faster for larger values of n, i.e., the test correctly claims security for smaller thresholds for larger n while for smaller ones false negatives are more likely. Note that n directly corresponds to the computational time, meaning that there is a trade-off between the type II error and the required computational time.

In Tables 2 and 3, respectively, we display the rejection rates for different first-order secure designs for SKINNY S-box and PRESENT S-box, respectively, for $\alpha = 1\%$. In both tables, 'Gad.' stands for Gadget and 'Opt.?' for whether we refer to the non-optimized or optimized version of the circuits (see Section 5). For each design, we chose two values $\epsilon_{0,I}$ and $\epsilon_{0,II}$, respectively, that are smaller and larger than the true lower bound of the simulation failure probability $\epsilon_{\min}$ ($t = 4$) obtained from VERICA$^+$, which is again used to show the statistical validity of the proposed procedure. More precisely, we choose $\epsilon_{0,I} = \epsilon_{\min}/2$ and $\epsilon_{0,II} = 2\epsilon_{\min}$. The rejection rate for $\epsilon_{0,I}$ is the rate of false positives α_I while the rejection rate for $\epsilon_{0,II}$ is the rate of true positives $1 - \beta_{II}$. We see that the type I error is zero for the considered thresholds $\epsilon_{0,I}$. Furthermore, the type II error is equal to zero for the considered thresholds $\epsilon_{0,II}$. It means that the test correctly labels circuits as secure with respect to $\epsilon_{0,II}$: Our tests provide reliable results given the assumptions. Note that the error rates might be slightly larger for different choices of $\epsilon_{0,I}$ and $\epsilon_{0,II}$ as demonstrated in Fig. 4. However, the type I error is bounded by α and the type II error will converge to zero for large enough n, see Definition A.1.

Additional simulation results for our superiority test procedure that further support the validity of our Algorithm 2, for insecure first-order, and secure and insecure second-order circuits are included in Section C.1.

Comparison to STRAPS. Fig. 5 illustrates the difference between $1 - \alpha = 90\%$ confidence intervals of STRAPS and ours, both for a non-optimized first-order SKINNY S-box masked implementation with the DOM gadget with $p = 0.01$(see Section 5.2). The diamond shapes represent the true value of the upper bound of the simulation failure probability $\epsilon_{\max}$ obtained from VERICA$^+$ (see Section 6.1). We observe that the confidence intervals obtained

Table 2: Rejection rate of Algorithm 2 for first-order secure SKINNY S-box designs for different numbers of drawn wire combinations n and for $\alpha = 1\%$.

	Circuit		**Model**	**Thresh.**	**Rejection rate for n, $\alpha_I/1-\beta_{II}$**			
Gadget	Opt.?	$\lvert\mathcal{W}\rvert$	p	$\epsilon_{0,I}/\epsilon_{0,II}$	100	200	500	1000
DOM	No	84	0.01	$0.0152/0.0608$	$0/0.998$	$0/1$	$0/1$	$0/1$
DOM	Yes	82	0.01	$0.0146/0.0584$	$0/0.996$	$0/1$	$0/1$	$0/1$
HPC1	No	136	0.01	$0.0182/0.0728$	$0/0.992$	$0/1$	$0/1$	$0/1$
HPC1	Yes	134	0.01	$0.0177/0.0709$	$0/0.992$	$0/1$	$0/1$	$0/1$
HPC2	No	160	0.01	$0.0455/0.182$	$0/1$	$0/1$	$0/1$	$0/1$
HPC2	Yes	158	0.01	$0.0447/0.179$	$0/1$	$0/1$	$0/1$	$0/1$
HPC3	No	136	0.01	$0.0222/0.0888$	$0/0.999$	$0/1$	$0/1$	$0/1$
HPC3	Yes	134	0.01	$0.0216/0.0863$	$0/0.999$	$0/1$	$0/1$	$0/1$

Table 3: Rejection rate of Algorithm 2 for first-order secure PRESENT S-box designs for different numbers of drawn wire combinations n and for $\alpha = 1\%$.

	Circuit		**Model**	**Thresh.**	**Rejection rate for n, $\alpha_I/1-\beta_{II}$**			
Gadget	Opt.?	$\lvert\mathcal{W}\rvert$	p	$\epsilon_{0,I}/\epsilon_{0,II}$	100	200	500	1000
DOM	No	182	0.01	$0.0398/0.159$	$0/1$	$0/1$	$0/1$	$0/1$
DOM	Yes	89	0.01	$0.0106/0.0425$	$0/0.974$	$0/1$	$0/1$	$0/1$
HPC1	No	290	0.01	$0.0257/0.103$	$0/0.995$	$0/1$	$0/1$	$0/1$
HPC1	Yes	125	0.01	$0.0109/0.0437$	$0/0.928$	$0/0.998$	$0/1$	$0/1$
HPC2	No	362	0.01	$0.0743/0.297$	$0/1$	$0/1$	$0/1$	$0/1$
HPC2	Yes	149	0.01	$0.0308/0.123$	$0/1$	$0/1$	$0/1$	$0/1$
HPC3	No	338	0.01	$0.0384/0.154$	$0/1$	$0/1$	$0/1$	$0/1$
HPC3	Yes	141	0.01	$0.0159/0.0636$	$0/0.989$	$0/1$	$0/1$	$0/1$

from STRAPS are much larger than the ones of our Algorithm 2. Whereas both confidence intervals guarantee (asymptotically) the same level $1-\alpha$, our proposed confidence intervals are more informative than the ones obtained by STRAPS. Indeed, they give a smaller interval with the same level of confidence. We highlight that the presented difference in confidence intervals occurs for all our considered designs (see Section 5) and for different levels α, therefore we omit the presentation of these results for the sake of brevity.

We consider the same circuit design and we further illustrate the difference between the rejection rates of Algorithm 2 and a rejection rule based on the confidence intervals from STRAPS for $\alpha = 5\%$ by displaying them in Fig. 6 (see also Fig. 4). We see that, not only our test has the desired level even for small sample sizes n, it also correctly classifies secure circuits as secure much more reliably than STRAPS. The latter fails to detect secure circuits when they are moderately close to desired threshold ϵ_0.

6.3 Testing for Equivalence

We demonstrate the validity of Algorithm 3 by studying the empirical rejection rate of the null hypothesis for various values of Δ and numbers of sampled wire combinations n, for several circuits.

As an example, we present the rejection rate of Algorithm 3 for first-order secure SKINNY S-box non-optimized with a DOM and a HPC1 gadget in Fig. 8, and non-optimized and optimized version with the same DOM gadget in Fig. 7. The horizontal line corresponds to the chosen α level (here $\alpha = 5\%$) and the vertical line corresponds the true difference between the lower bounds of the simulation failure probabilities $\epsilon_{\min}$ (for $t = 4$) of the corresponding circuits given by VERICA$^+$. The rejection rate for values smaller than the true difference of $\epsilon_{\min}$ of both circuits corresponds to the type I error

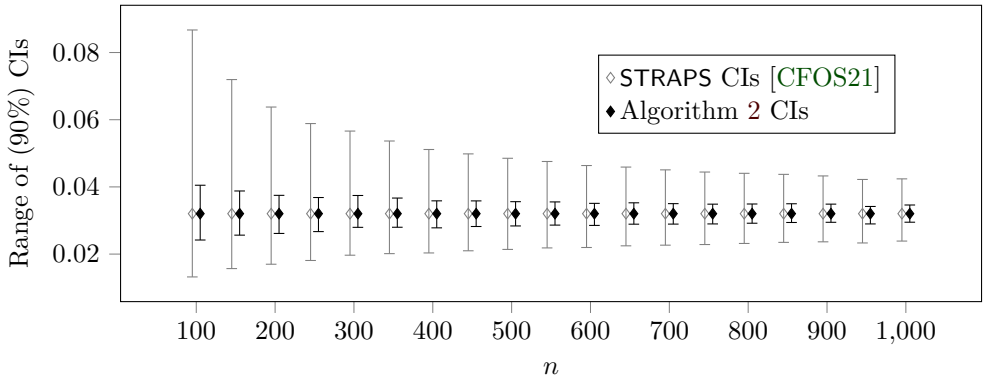

Figure 5: Comparison between the 90% confidence intervals (CIs) computed by STRAPS and ours used in Algorithm 2 for a first-order non-optimized secure implementation of the SKINNY S-box with the DOM gadget. The diamond shapes are the true value of $\epsilon_{\max}$.

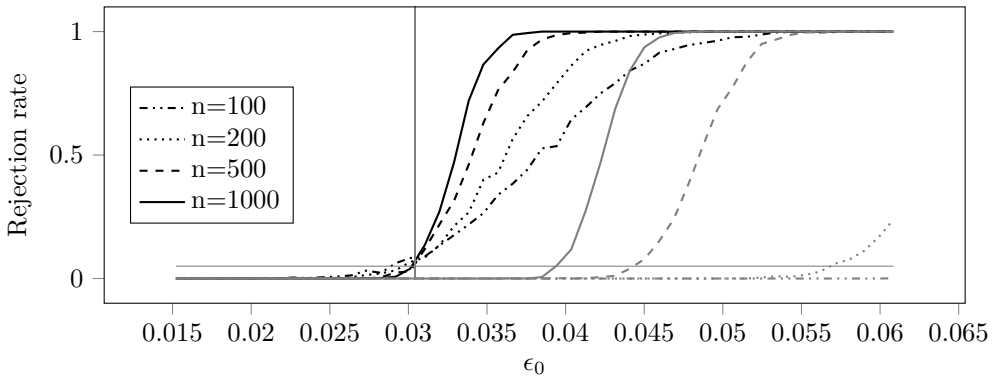

Figure 6: Rejection rate of Algorithm 2 (black) and using the confidence intervals from STRAPS (gray) for first-order secure non-optimized SKINNY S-box with DOM gadget for $p = 0.01$ and various numbers of drawn wire combinations n. The horizontal line corresponds to the significance level of $\alpha = 5\%$ and the vertical line to $\epsilon_{\min} = 0.032$.

α_I, while the rejection rate for values larger than this difference corresponds to a correct rejection, i.e., to $1 - \beta_{II}$.

We see that the level of $\alpha = 5\%$ is guaranteed, even for small sample sizes. Moreover, the type II error decreases for larger sample sizes n and similarity thresholds Δ, meaning that the test correctly labels the two circuits as similar (for $\Delta \geq 0.02$). For smaller Δ, false negatives, or labeling the circuits wrongly as not equivalent, are more likely.

In Tables 4 and 5, we display the rejection rates for different masked implementations of first-order SKINNY S-box and PRESENT S-box, respectively, with both using the DOM gadget and for $\alpha = 1\%$. In both tables, 'Sec.?' stands for whether we consider the secure or insecure circuit version and 'Opt.?' for whether we refer to the non-optimized or optimized circuits (see Section 5). For each design, we choose two values Δ_I and Δ_{II}, respectively, that are smaller and larger than the true difference of the lower bound of simulation failure probabilities obtained from VERICA$^+$. More precisely, we choose Δ_I as half of the true difference, and Δ_{II} as twice the true difference but at least 0.05. The rejection rate for Δ_I is the rate of false positives α_I, while the rejection rate for Δ_{II} is the rate of true positives $1 - \beta_{II}$. We observe that the type I error is 0 for all considered values for Δ_I. Furthermore, the type II error is close to zero for $n \geq 500$, while it is larger

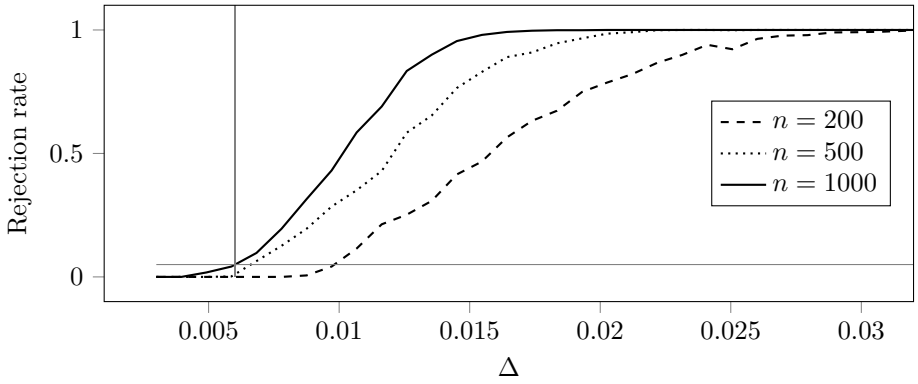

Figure 7: Rejection rate of Algorithm 3 for 1st-order non-optimized SKINNY S-box with DOM and HPC1 gadget ($p_1 = p_2 = 0.01$) for distinct number of samples n. The horizontal line is the significance level $\alpha = 5\%$ and the vertical one the true difference of the lower bounds of ϵ_1, ϵ_2 ($= 0.006$).

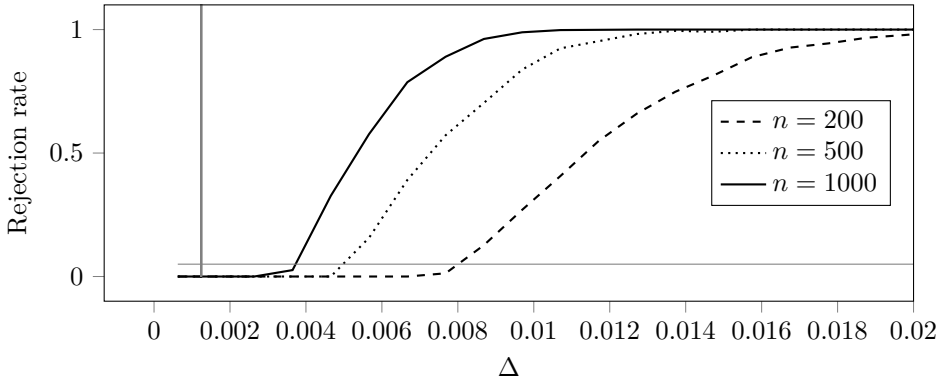

Figure 8: Rejection rate of Algorithm 3 for first-order optimized and non-optimized SKINNY S-box with DOM gadget ($p_1 = p_2 = 0.01$) for distinct number of samples n. The horizontal line is the significance level $\alpha = 5\%$ and the vertical one the true difference of the lower bounds of ϵ_1, ϵ_2 ($= 0.0012$).

for $n = 200$, which indicates that larger samples should be considered when testing for equivalence. Note that the error rates might be slightly larger for different choices of Δ_I and Δ_{II} as demonstrated in Fig. 7. However, the type I error is bounded by α and the type II error converges to zero for large enough n, which makes our tests fit for practical use (see Section 2.4), see details in Definition A.2.

We provide further simulation results for our equivalence testing that continue to support the validity of Algorithm 3, for our first- and second-order S-boxes in Section C.2.

7 Discussion

In this paper, we present a statistical framework to compare the security level of two circuits in the random probing model, utilizing equivalence testing. In addition, we present a statistical procedure to rapidly determine whether the security of a given circuit exceeds a fixed threshold with superiority testing. For both methods, we provide proofs of concepts to show their validity with two masked S-boxes using variant implementations and various multiplication gadgets by giving numerous empirical rejection rates for various random

Table 4: Rejection rate of Algorithm 3 for different implementation first-order SKINNY S-box with the DOM gadget for distinct numbers of samples n and for $\alpha = 1\%$.

Circuit 1		**Circuit 2**		**Model**		**Threshold**	**Rejection rate**		
Opt. ?	Sec. ?	Opt. ?	Sec. ?	p_1	p_2	Δ_I/Δ_{II}	$\alpha_I/1-\beta_{II}$ 200	$\alpha_I/1-\beta_{II}$ 500	$\alpha_I/1-\beta_{II}$ 1000
No	Yes	No	No	0.01	0.01	$^{0.014}/_{0.0559}$	$^{0}/_{0.937}$	$^{0}/_{1}$	$^{0}/_{1}$
No	Yes	Yes	No	0.01	0.01	$^{0.0132}/_{0.0528}$	$^{0}/_{0.923}$	$^{0}/_{0.999}$	$^{0}/_{1}$
Yes	Yes	Yes	No	0.01	0.01	$^{0.0138}/_{0.0553}$	$^{0}/_{0.941}$	$^{0}/_{1}$	$^{0}/_{1}$
No	Yes	Yes	Yes	0.01	0.01	$^{6.25e\text{-}4}/_{0.05}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$

Table 5: Rejection rate of Algorithm 3 for different implementation first-order PRESENT S-box with the DOM gadget for distinct numbers of samples n and for $\alpha = 1\%$.

Circuit 1		**Circuit 2**		**Model**		**Threshold**	**Rejection rate**		
Opt. ?	Sec. ?	Opt. ?	Sec. ?	p_1	p_2	Δ_I/Δ_{II}	$\alpha_I/1-\beta_{II}$ 200	$\alpha_I/1-\beta_{II}$ 500	$\alpha_I/1-\beta_{II}$ 1000
No	Yes	No	No	0.01	0.01	$^{0.0125}/_{0.05}$	$^{0}/_{0.485}$	$^{0}/_{0.892}$	$^{0}/_{0.998}$
No	Yes	Yes	No	0.01	0.01	$^{0.0158}/_{0.0632}$	$^{0}/_{0.882}$	$^{0}/_{0.999}$	$^{0}/_{1}$
Yes	Yes	Yes	No	0.01	0.01	$^{0.0133}/_{0.0533}$	$^{0}/_{0.972}$	$^{0}/_{1}$	$^{0}/_{1}$
No	Yes	Yes	Yes	0.01	0.01	$^{0.0291}/_{0.117}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$

sample size. Importantly, our methodology can compare the level of security of circuits by randomly sampling their wires while providing bounds on the probability of a false negative and false positive, and thus, reduce the amount of computational resources needed to produce a precise approximation for each ϵ. Our tests provide a trade-off between the chosen number of random samples (computational time) and accuracy.

The procedures introduced in this paper can be applied to more finely tune implementations of cryptographic designs in the random probing model and assessing promptly the resulting level of security with a quantified confidence level: One could try to purposely reduce the simulation failure probability ϵ of a circuit according to the Random Probing Expandability (RPE) strategy [BCP+20], which replaces a wire by n^k ones that carry the same information, where $k \in \mathbb{N}$ is the expansion factor and $n \in \mathbb{N}$ is the number of wire shares. The logical gates have to be consequently replaced by gadgets handling this duplication of variables. Then, a designer could use the equivalence test to compare the resulting simulation failure probabilities, or the superiority test to decide whether the targeted security threshold is not exceeded. For that, an important future direction is to find a good choice for the parameters Δ and ϵ_0 in practice in order to balance between security against a SCA adversary and the number of samples to draw.

Another example is to estimate ϵ for a gadget with respect to the Random Probing Composability (RPC) security notion [BCP+20]. Indeed, for the latter, simulation failure probabilities are computed for the individual gadgets composing a circuit, with some additional constraints on the number of input and output shares, and the highest probability value defines the security level for the overall circuit. It equals the maximum ϵ of all the gadgets multiplied by the number of gadgets, for a given p. In this case, one could use our superiority test to decide whether a gadget surpasses a targeted threshold.

An open question is how the presented methods can be extended to verification tools applying a mix of deterministic and probabilistic approach, like INDIANA. In addition, one could generalize the methodology of this paper to compare the security of more than two circuits at once. Our testing methodology is not restricted to the measure of the simulation failure probability, but can be applied to estimate any other security parameters.

Acknowledgments

We would like to express our gratitude to Fabian Buschkowski for his generous support with HADES, to Sonia Belaïd for her valuable clarifications regarding the random probing model, and to Gaëtan Cassiers for his insightful input on STRAPS. We would also like to thank the TCHES reviewers for their constructive comments and suggestions. Anna Guinet, Carina Graw, Holger Dette and Tim Güneysu have been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972. Tim Güneysu has also been supported by the German Federal Ministry of Education and Research BMBF through the projects DIOWAS (16ME0967) and POST (16ME2157), and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under EMBOSOM (535695900).

References

[ADF16] Marcin Andrychowicz, Stefan Dziembowski, and Sebastian Faust. Circuit compilers with $O(1/\log(n))$ leakage rate. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 586–615, May 2016.

[AIS18] Prabhanjan Ananth, Yuval Ishai, and Amit Sahai. Private circuits: A modular approach. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 427–455, August 2018.

[Ajt11] Miklós Ajtai. Secure computation with information leaking to an adversary. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 715–724. ACM Press, June 2011.

[BA95] J. Martin Bland and Douglas G. Altman. Multiple significance tests: the bonferroni method. *BMJ*, 310(6973):170, 1995.

[BCP+20] Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Abdul Rahman Taleb. Random probing security: Verification, composition, expansion and new constructions. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 339–368, August 2020.

[BCPZ16] Alberto Battistello, Jean-Sébastien Coron, Emmanuel Prouff, and Rina Zeitoun. Horizontal side-channel attacks and countermeasures on the ISW masking scheme. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *CHES 2016*, volume 9813 of *LNCS*, pages 23–39, August 2016.

[BDK24] Patrick Bastian, Holger Dette, and Lukas Koletzko. Testing equivalence of multinomial distributions — A constrained bootstrap approach. *Statistics & Probability Letters*, 206:109999, 2024.

[BFG+24a] Sonia Belaïd, Jakob Feldtkeller, Tim Güneysu, Anna Guinet, Jan Richter-Brockmann, Matthieu Rivain, Pascal Sasdrich, and Abdul Rahman Taleb. Formal definition and verification for combined random fault and random probing security. LNCS, pages 167–200, December 2024.

[BFG+24b] Sonia Belaid, Jakob Feldtkeller, Tim Güneysu, Anna Guinet, Jan Richter-Brockmann, Matthieu Rivain, Pascal Sasdrich, and Abdul Rahman Taleb. Formal definition and verification for combined random fault and random probing security. Cryptology ePrint Archive, Report 2024/757, 2024.

[BFG+25] Christof Beierle, Jakob Feldtkeller, Anna Guinet, Tim Güneysu, Gregor Leander, Jan Richter-Brockmann, and Pascal Sasdrich. INDIANA - verifying (random) probing security through indistinguishability analysis. LNCS, pages 33–63, June 2025.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 123–153, August 2016.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES 2007*, volume 4727 of *LNCS*, pages 450–466, September 2007.

[BLH+25] Fabian Buschkowski, Georg Land, Niklas Höher, Jan Richter-Brockmann, Pascal Sasdrich, and Tim Güneysu. HADES: Automated hardware design exploration for cryptographic primitives. *IACR TCHES*, 2025(4):1–45, 2025.

[BMRT22] Sonia Belaïd, Darius Mercadier, Matthieu Rivain, and Abdul Rahman Taleb. IronMask: Versatile verification of masking security. In *2022 IEEE Symposium on Security and Privacy*, pages 142–160. IEEE Computer Society Press, May 2022.

[CB24] George Casella and Roger Berger. *Statistical inference*. CRC press, 2nd edition, 2024.

[CFOS21] Gaëtan Cassiers, Sebastian Faust, Maximilian Orlt, and François-Xavier Standaert. Towards tight random probing security. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part III*, volume 12827 of *LNCS*, pages 185–214, Virtual Event, August 2021.

[CGLS21] Gaëtan Cassiers, Benjamin Grégoire, Itamar Levi, and François-Xavier Standaert. Hardware private circuits: From trivial composition to full verification. *IEEE Trans. Computers*, 70(10):1677–1690, 2021.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 398–412, August 1999.

[CSV24] Gaëtan Cassiers, François-Xavier Standaert, and Corentin Verhamme. Low-latency masked gadgets robust against physical defaults with application to ascon. *IACR TCHES*, 2024(3):603–633, 2024.

[DDF14] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 423–440, May 2014.

[DH04] Herold Dehling and Beate Haupt. *Grundbegriffe der Testtheorie*, pages 249–272. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[Dun61] Olive Jean Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, 1961.

[FGM+18] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and François-Xavier Standaert. Composable masking schemes in the presence of physical defaults & the robust probing model. *IACR TCHES*, 2018(3):89–120, 2018.

[GGJR+11] Benjamin Jun Gilbert Goodwill, Josh Jaffe, Pankaj Rohatgi, et al. A testing methodology for side-channel resistance validation. In *NIST non-invasive attack testing workshop*, volume 7, pages 115–136, 2011.

[GMK16] Hannes Groß, Stefan Mangard, and Thomas Korak. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. In *TIS@CCS*, page 3. ACM, 2016.

[GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES 2001*, volume 2162 of *LNCS*, pages 251–261, May 2001.

[GP99] Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *CHES'99*, volume 1717 of *LNCS*, pages 158–172, August 1999.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481, August 2003.

[JMB24] Vahid Jahandideh, Bart Mennink, and Lejla Batina. An algebraic approach for evaluating random probing security with application to AES. *IACR TCHES*, 2024(4):657–689, 2024.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397, August 1999.

[KM22] David Knichel and Amir Moradi. Low-latency hardware private circuits. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 1799–1812. ACM Press, November 2022.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 104–113, August 1996.

[KSM20] David Knichel, Pascal Sasdrich, and Amir Moradi. SILVER - statistical independence and leakage verification. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 787–816, December 2020.

[LRY+05] Giselle B Limentani, Moira C Ringo, Feng Ye, Mandy L Bergquist, and Ellen O McSorley. Beyond the t-test: statistical equivalence testing, 2005.

[Mor03] Matthew D Moran. Arguments for rejecting the sequential bonferroni in ecological studies. *Oikos*, 100(2):403–405, 2003.

[PGMP19] Thomas Prest, Dahmun Goudarzi, Ange Martinelli, and Alain Passelègue. Unifying leakage models on a Rényi day. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 683–712, August 2019.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 142–159, May 2013.

[RBFSG22] Jan Richter-Brockmann, Jakob Feldtkeller, Pascal Sasdrich, and Tim Güneysu. VERICA - verification of combined attacks automated formal verification of security against simultaneous information leakage and tampering. *IACR TCHES*, 2022(4):255–284, 2022.

[RD20] Mark Randolph and William Diehl. Power side-channel attack analysis: A review of 20 years of study for the layman. *Cryptogr.*, 4(2):15, 2020.

[RF04] Andrew P Robinson and Robert E Froese. Model validation using equivalence tests. *Ecological Modelling*, 176(3-4):349–358, 2004.

[RV23] Mohanapriya R and Nithish Kumar V. Optimized implementation of S-box and inverse S-box for PRESENT lightweight block cipher. In *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, page 1–5. IEEE, May 2023.

[Sch87] Donald J. Schuirmann. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics*, 15(6):657–680, Dec 1987.

[SCS+21] Madura A. Shelton, Lukasz Chmielewski, Niels Samwel, Markus Wagner, Lejla Batina, and Yuval Yarom. Rosita++: Automatic higher-order leakage elimination from cryptographic code. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 685–699. ACM Press, November 2021.

[Wel02] Stefan Wellek. *Testing statistical hypotheses of equivalence*. Chapman and Hall/CRC, 2002.

A Statistics

In this appendix, we state the mathematical justification for the tests presented in Section 4. Let $Var(X)$ denote the variance of a random variable X, $min(a,b)$ the minimum of two real numbers a, b and $\lim_{n\to\infty} a_n$ the limiting behavior of a sequence a_n when n converges to infinity. If a limit a exists, we write $a_n \to a$ as well. Furthermore, $N(0,1)$ represents the standard normal distribution. By *weak convergence* of a random variable, we refer to the convergence of its distribution function.

Theorem A.1. *The test defined in Test 1 is a consistent, asymptotic α-test. More precisely, we have*

$$\lim_{n\to\infty} \mathbb{P}\Big(\hat{\epsilon} < \epsilon_0 + \frac{\hat{\sigma} z_\alpha}{\sqrt{n}}\Big) = \begin{cases} 0, & \epsilon > \epsilon_0 \\ \alpha, & \epsilon = \epsilon_0 \\ 1, & \epsilon < \epsilon_0. \end{cases}$$

Proof. Note that for $i = 1, ..., |\mathcal{W}|$, we have $Var(\hat{r}_i) = r_i(1-r_i)/n$ and the weak convergence

$$\sqrt{n}(\hat{r}_i - r_i) \to N(0, r_i(1-r_i))$$

by the central limit theorem. Therefore, since all $\hat{r}_i$ are independent,

$$Var(\hat{\epsilon}) = \frac{1}{n}\sum_{i=1}^{|\mathcal{W}|} r_i(1-r_i)\binom{|\mathcal{W}|}{i}^2 p^{2i}(1-p)^{2(|\mathcal{W}|-i)} = \frac{\sigma^2}{n}$$

and (note again independence of $\hat{r}_i$)

$$\sqrt{n}(\hat{\epsilon} - \epsilon) \to N(0, \sigma^2),$$

weakly, where $\sigma^2 = \sum_{i=1}^{|\mathcal{W}|} r_i(1-r_i)\binom{|\mathcal{W}|}{i}^2 p^{2i}(1-p)^{2(|\mathcal{W}|-i)}$ since the limit is a sum of independent Gaussians. Since $\hat{r}_i \to r_i$ in probability by the law of large numbers, $\hat{\sigma} \to \sigma$ in probability and therefore

$$\sqrt{n}\left(\frac{\hat{\epsilon} - \epsilon}{\hat{\sigma}}\right) \to N(0,1) \tag{9}$$

weakly, where $\hat{\sigma}^2$ is the plug-in estimator of σ^2 defined in equation (6). From (9) the assertion of Theorem A.1 follows. □

Theorem A.2. *The test defined in Test 2 is a consistent, asymptotic α-test. More precisely, we have*

$$\lim_{n\to\infty} \mathbb{P}\Big(\hat{\epsilon}_1 - \hat{\epsilon}_2 \le \Delta + \frac{\hat{\sigma}_{dif} z_\alpha}{\sqrt{n}} \text{ and } \hat{\epsilon}_1 - \hat{\epsilon}_2 \ge -\Delta + \frac{\hat{\sigma}_{dif} z_{1-\alpha}}{\sqrt{n}}\Big) = \begin{cases} 0, & |\epsilon_1 - \epsilon_2| > \Delta \\ \alpha, & |\epsilon_1 - \epsilon_2| = \Delta \\ 1, & |\epsilon_1 - \epsilon_2| < \Delta. \end{cases}$$

Proof. As a first step, note that

$$\sqrt{n}\frac{\hat{\epsilon}_1 - \epsilon_2 - \hat{\epsilon}_1 + \epsilon_2}{\hat{\sigma}_{dif}} \to N(0,1) \tag{10}$$

weakly by the same arguments used for the proof of (9) and since $Var(\hat{\epsilon}_1 - \hat{\epsilon}_2) = Var(\hat{\epsilon}_1) + Var(\hat{\epsilon}_2) = \sigma_{dif}^2$ by the independence of $\hat{\epsilon}_1$ and $\hat{\epsilon}_2$. We now consider each of the three cases separately.

For $|\epsilon_1 - \epsilon_2| > \Delta$: It is sufficient to show

$$\min\left\{\mathbb{P}\Big(\hat{\epsilon}_1 - \hat{\epsilon}_2 \leq \Delta + \frac{\hat{\sigma}_{dif} z_\alpha}{\sqrt{n}}\Big), \mathbb{P}\Big(\hat{\epsilon}_1 - \hat{\epsilon}_2 \geq -\Delta + \frac{\hat{\sigma}_{dif} z_{1-\alpha}}{\sqrt{n}}\Big)\right\} \to 0,$$

since $\mathbb{P}(A \cap B) \leq \min\{\mathbb{P}(A), \mathbb{P}(B)\}$ for two events A and B. This assertion follows directly from (10).

For $|\epsilon_1 - \epsilon_2| = \Delta$: We only show the convergence for the case $\epsilon_1 - \epsilon_2 = \Delta$, since the case $\epsilon_1 - \epsilon_2 = -\Delta$ follows by similar arguments. We remind that $\mathbb{P}(A \cap B) = \mathbb{P}(A) \cdot \mathbb{P}(B|A)$ and therefore

$$\begin{aligned}
&\mathbb{P}\left(\hat{\epsilon}_1 - \hat{\epsilon}_2 \leq \Delta + \frac{\hat{\sigma}_{dif} z_\alpha}{\sqrt{n}} \text{ and } \hat{\epsilon}_1 - \hat{\epsilon}_2 \geq -\Delta + \frac{\hat{\sigma}_{dif} z_{1-\alpha}}{\sqrt{n}}\right) \\
&= \mathbb{P}\left(\hat{\epsilon}_1 - \hat{\epsilon}_2 \leq \Delta + \frac{\hat{\sigma}_{dif} z_\alpha}{\sqrt{n}} | \hat{\epsilon}_1 - \hat{\epsilon}_2 \geq -\Delta + \frac{\hat{\sigma}_{dif} z_{1-\alpha}}{\sqrt{n}}\right) \cdot \mathbb{P}\left(\hat{\epsilon}_1 - \hat{\epsilon}_2 \geq -\Delta + \frac{\hat{\sigma}_{dif} z_{1-\alpha}}{\sqrt{n}}\right)
\end{aligned}$$

Since $\hat{\epsilon}_1 - \hat{\epsilon}_2$ converges in probability to Δ it holds that

$$\mathbb{P}\left(\hat{\epsilon}_1 - \hat{\epsilon}_2 \geq -\Delta + \frac{\hat{\sigma}_{dif} z_{1-\alpha}}{\sqrt{n}}\right) \to 1.$$

Therefore, and by (10), it follows that

$$\begin{aligned}
&\lim_{n\to\infty} \mathbb{P}\left(\hat{\epsilon}_1 - \hat{\epsilon}_2 \leq \Delta + \frac{\hat{\sigma}_{dif} z_\alpha}{\sqrt{n}} \mid \hat{\epsilon}_1 - \hat{\epsilon}_2 \geq -\Delta + \frac{\hat{\sigma}_{dif} z_{1-\alpha}}{\sqrt{n}}\right) \\
&= \lim_{n\to\infty} \mathbb{P}\left(\hat{\epsilon}_1 - \hat{\epsilon}_2 \leq \Delta + \frac{\hat{\sigma}_{dif} z_\alpha}{\sqrt{n}}\right) = \alpha.
\end{aligned}$$

For $|\epsilon_1 - \epsilon_2| < \Delta$: Note that $|\hat{\epsilon}_1 - \hat{\epsilon}_2|$ converges in probability to $|\epsilon_1 - \epsilon_2|$, hence the assertion follows. □

B Hardware implementations

B.1 Lookup Tables

Table 6: 4-bit PRESENT S-box Lookup Table (hexadecimal notation).

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
y	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Table 7: 4-bit SKINNY S-box Lookup Table (hexadecimal notation).

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
y	C	6	9	0	1	A	2	B	3	8	5	D	4	E	7	F

B.2 Secure and Insecure Masked S-boxes Designs

We report the number of gates, wires (SCA locations) and random bits for each PRESENT S-box masked circuits in Table 8, and for SKINNY ones in Table 9, for $d \in \{1, 2\}$ and (non-) optimized implementations, as reported by VERICA$^+$. In addition, for each implementation, we provide an upper and a lower bound of the simulation failure probability for combinations

up to $t = 4$ wires (see Section 6.1). The lower bound $\epsilon_{\min}$ is computed without the binomial tree optimization of VERICA$^+$ for a leakage rate p (if this optimization is activated, $\epsilon_{\min}$ is closer to its true value ϵ, see the extended paper version [BFG+24b]).

For the non-optimized PRESENT S-box implementation masked with the HPC3 gadget at the second order, VERICA$^+$ could only run the estimation of the lower and upper bounds of ϵ up to combinations of $t = 3$ wires due to a fatal error with the internal CUDD library with $t = 4$,[4] which we were not able to fix even after contacting the developers of VERICA$^+$. The affected probabilities values are marked by a star $\star$.

Table 8: Description of considered PRESENT S-box masked circuits.

Masked Design	**Gadget** $\wedge$	**Order** d	**Circuit** $\lvert\mathcal{G}\rvert$	$\lvert\mathcal{W}\rvert$	**Random.** # bits	**Model** p	$\epsilon_{\min}/\epsilon_{\max}$
PRESENT S-box Secure	DOM	1	170	182	12	0.01	7.9510e-2/1.1656e-1
		2	396	432	36	0.01	2.942e-3/4.0754e-1
	HPC1	1	266	290	24	0.01	5.1383e-2/2.1687e-1
		2	552	612	60	0.01	4.78e-4/5.4794e-1
	HPC2	1	350	362	12	0.01	1.4856e-1/4.3593e-1
		2	864	900	36	0.01	1.014e-3/3.1924e-1
	HPC3	1	314	338	24	0.01	7.6877e-2/3.2242e-1
		2	756	828	72	0.01	3.9e-5*/2.8867e-1*
PRESENT S-box Insecure	DOM	1	170	181	11	0.01	1.0609e-1/1.4240e-1
		2	396	429	33	0.01	1.8085e-2/4.1802e-1
	HPC1	1	266	288	22	0.01	7.3335e-2/2.3568e-1
		2	552	607	55	0.01	5.608e-3/5.5323e-1
	HPC2	1	350	361	11	0.01	1.6630e-1/4.5193e-1
		2	864	897	33	0.01	2.402e-3/3.2373e-1
	HPC3	1	314	336	22	0.01	9.3885e-2/3.3597e-1
		2	756	822	66	0.01	4.29e-4*/2.9577e-1*
PRESENT S-box Optimized Secure	DOM	1	85	89	4	0.01	2.1286e-2/2.3356e-2
		2	173	185	12	0.01	1.161e-3/4.0448e-2
	HPC1	1	117	125	8	0.01	2.1860e-2/3.0585e-2
		2	225	245	20	0.01	7.88e-4/1.0162e-1
	HPC2	1	145	149	4	0.01	6.1569e-2/7.9096e-2
		2	329	341	12	0.01	4.274e-3/2.5502e-1
	HPC3	1	133	141	8	0.01	3.1825e-2/4.5956e-2
		2	293	317	24	0.01	1.159e-3/2.1076e-1
PRESENT S-box Optimized Insecure	DOM	1	85	88	3	0.01	5.2067e-2/5.4037e-2
		2	173	182	9	0.01	3.5763e-2/7.2809e-2
	HPC1	1	117	123	6	0.01	4.8464e-2/5.6633e-2
		2	225	240	15	0.01	2.7330e-2/1.2187e-1
	HPC2	1	145	148	3	0.01	1.0742e-2/1.2450e-1
		2	329	338	9	0.01	4.5168e-2/2.9071e-1
	HPC3	1	133	139	6	0.01	5.9879e-2/7.3236e-2
		2	293	311	18	0.01	2.4679e-2/2.2423e-1

[4] `cuddGarbageCollect: problem in constants. dead count != deleted`

Table 9: Description of considered SKINNY S-box masked circuits.

Masked Design	Gadget	Order	Circuit		Random.	Model	
	$\wedge$	d	$\lvert\mathcal{G}\rvert$	$\lvert\mathcal{W}\rvert$	# bits	p	$\epsilon_{\min}/\epsilon_{\max}$
SKINNY S-box Secure	DOM	1	80	84	4	0.01	3.0425e-2/3.2029e-2
		2	164	176	12	0.01	2.715e-3/3.5517e-2
	HPC1	1	128	136	8	0.01	3.6419e-2/4.8669e-2
		2	240	260	20	0.01	1.440e-3/1.2232e-1
	HPC2	1	156	160	4	0.01	9.1005e-2/1.1402e-1
		2	344	356	12	0.01	7.350e-3/2.8425e-1
	HPC3	1	128	136	8	0.01	4.4407e-2/5.6657e-2
		2	284	308	24	0.01	2.280e-3/1.9685e-1
SKINNY S-box Insecure	DOM	1	80	83	3	0.01	5.8432e-2/5.9953e-2
		2	164	173	9	0.01	3.9006e-2/6.9806e-2
	HPC1	1	128	134	6	0.01	7.3110e-2/8.4658e-2
		2	240	255	15	0.01	3.3090e-2/1.4710e-1
	HPC2	1	156	159	3	0.01	1.3631e-1/1.5879e-1
		2	344	353	9	0.01	4.9767e-2/3.2142e-1
	HPC3	1	128	134	6	0.01	7.3621e-2/8.5169e-2
		2	284	302	18	0.01	2.8953e-2/2.1367e-1
SKINNY S-box Optimized Secure	DOM	1	78	82	4	0.01	2.9176e-2/3.0617e-2
		2	162	174	12	0.01	2.639e-3/3.4097e-2
	HPC1	1	126	134	8	0.01	3.5428e-2/4.6976e-2
		2	238	258	20	0.01	1.416e-3/1.1952e-1
	HPC2	1	154	158	4	0.01	8.9420e-2/1.1137e-1
		2	342	354	12	0.01	7.323e-3/2.8073e-1
	HPC3	1	126	134	8	0.01	4.3143e-2/5.4691e-2
		2	282	306	24	0.01	2.245e-3/1.93514e-1
SKINNY S-box Optimized Insecure	DOM	1	78	81	3	0.01	5.6820e-2/5.8185e-2
		2	162	171	9	0.01	3.8698e-2/6.8206e-2
	HPC1	1	126	132	6	0.01	7.1435e-2/8.2309e-2
		2	238	253	15	0.01	3.2913e-2/1.4423e-1
	HPC2	1	154	157	3	0.01	1.3444e-1/1.5586e-1
		2	342	351	9	0.01	4.9926e-2/3.1809e-1
	HPC3	1	126	132	6	0.01	7.1966e-2/8.2839e-2
		2	282	300	18	0.01	2.8963e-2/2.1043e-1

C Additional Simulation Results

C.1 Superiority Test - Algorithm 2

In the following tables, the rejection rates are reported for distinct numbers of drawn wire combinations n for $\alpha = 1\%$. Moreover, $\epsilon_{0,I}$ and $\epsilon_{0,II}$ are chosen smaller and larger than the true lower bound $\epsilon_{\min}$ for each circuit. In all tables, 'Gad.' means Gadget and 'Opt.?' refers to whether we study the non-optimized or optimized version of the circuits (see Section 5).

We provide empirical results for SKINNY S-box circuits that are first-order insecure in Table 10, second-order secure in Table 11, and second-order insecure in Table 12.

Similarly, we give the results for PRESENT S-box circuits that are first-order insecure

in Table 13, second-order secure in Table 14, and second-order insecure in Table 15.

Table 10: Rejection rate of Algorithm 2 for first-order insecure SKINNY S-box designs.

	Circuit		**Model**	**Thresh.**	**Rejection rate** n, $\alpha_I/1-\beta_{II}$			
Gad.	Opt.?	$\lvert\mathcal{W}\rvert$	p	$\epsilon_{0,I}/\epsilon_{0,II}$	100	200	500	1000
DOM	No	83	0.01	$0.0292/0.117$	$0/0.999$	$0/1$	$0/1$	$0/1$
	Yes	81	0.01	$0.0284/0.114$	$0/1$	$0/1$	$0/1$	$0/1$
HPC1	No	134	0.01	$0.0352/0.141$	$0/1$	$0/1$	$0/1$	$0/1$
	Yes	132	0.01	$0.0344/0.137$	$0/1$	$0/1$	$0/1$	$0/1$
HPC2	No	159	0.01	$0.0671/0.268$	$0/1$	$0/1$	$0/1$	$0/1$
	Yes	157	0.01	$0.0641/0.256$	$0/1$	$0/1$	$0/1$	$0/1$
HPC3	No	134	0.01	$0.0355/0.142$	$0/1$	$0/1$	$0/1$	$0/1$
	Yes	132	0.01	$0.036/0.144$	$0/1$	$0/1$	$0/1$	$0/1$
DOM	No	83	0.001	$1.22e\text{-}3/4.89e\text{-}3$	$0.093/0.547$	$0.005/0.654$	$0/0.958$	$0/1$
	Yes	81	0.001	$1.21e\text{-}3/4.85e\text{-}3$	$0.072/0.55$	$0.004/0.685$	$0/0.962$	$0/1$
HPC1	No	134	0.001	$9.08e\text{-}4/3.63e\text{-}3$	$0.012/0.479$	$0.002/0.565$	$0/0.849$	$0/0.981$
	Yes	132	0.001	$8.94e\text{-}4/3.57e\text{-}3$	$0.021/0.464$	$0.004/0.549$	$0/0.843$	$0/0.977$
HPC2	No	159	0.001	$2.31e\text{-}3/9.23e\text{-}3$	$0.01/0.557$	$0/0.778$	$0/0.99$	$0/1$
	Yes	157	0.001	$1.43e\text{-}3/5.72e\text{-}3$	$0.001/0.547$	$0/0.771$	$0/0.981$	$0/1$
HPC3	No	134	0.001	$9.14e\text{-}4/3.66e\text{-}3$	$0.013/0.482$	$0.003/0.553$	$0/0.854$	$0/0.984$
	Yes	132	0.001	$1.34e\text{-}3/5.35e\text{-}3$	$0.057/0.521$	$0.03/0.629$	$0/0.898$	$0/0.996$

Table 11: Rejection rate of Algorithm 2 for second-order secure SKINNY S-box designs.

	Circuit		**Model**	**Thresh.**	**Rejection rate** n, $\alpha_I/1-\beta_{II}$			
Gad.	Opt.?	$\lvert\mathcal{W}\rvert$	p	$\epsilon_{0,I}/\epsilon_{0,II}$	100	200	500	1000
DOM	No	176	0.01	$1.36e\text{-}3/5.43e\text{-}3$	$0.036/0.421$	$0.011/0.537$	$0.001/0.851$	$0/0.98$
	Yes	174	0.01	$1.32e\text{-}3/5.28e\text{-}3$	$0.041/0.434$	$0.013/0.523$	$0/0.836$	$0/0.977$
HPC1	No	260	0.01	$7.2e\text{-}4/2.88e\text{-}3$	$0.402/0.377$	$0.163/0.388$	$0.009/0.474$	$0.003/0.683$
	Yes	258	0.01	$7.08e\text{-}4/2.83e\text{-}3$	$0.398/0.387$	$0.17/0.416$	$0.004/0.481$	$0.001/0.675$
HPC2	No	356	0.01	$3.68e\text{-}3/1.47e\text{-}2$	$0.023/0.47$	$0.004/0.642$	$0/0.953$	$0/1$
	Yes	354	0.01	$3.66e\text{-}3/1.46e\text{-}2$	$0.039/0.474$	$0.004/0.655$	$0/0.943$	$0/1$
HPC3	No	308	0.01	$1.14e\text{-}3/4.56e\text{-}3$	$0.288/0.279$	$0.085/0.298$	$0/0.585$	$0/0.824$
	Yes	306	0.01	$1.12e\text{-}3/4.49e\text{-}3$	$0.289/0.281$	$0.072/0.302$	$0.002/0.563$	$0/0.813$
DOM	No	176	0.001	$2.86e\text{-}6/1.14e\text{-}5$	$0.447/0.522$	$0.263/0.276$	$0.037/0.511$	$0.002/0.648$
	Yes	174	0.001	$2.74e\text{-}6/1.1e\text{-}5$	$0.463/0.523$	$0.281/0.265$	$0.052/0.484$	$0.002/0.676$
HPC1	No	260	0.001	$2.6e\text{-}6/1.04e\text{-}5$	$0.415/0.824$	$0.405/0.676$	$0.272/0.407$	$0.132/0.42$
	Yes	258	0.001	$2.52e\text{-}6/1.01e\text{-}5$	$0.383/0.83$	$0.393/0.715$	$0.274/0.386$	$0.146/0.427$
HPC2	No	356	0.001	$2.77e\text{-}5/1.11e\text{-}4$	$0.183/0.482$	$0.11/0.53$	$0.012/0.622$	$0/0.864$
	Yes	354	0.001	$2.72e\text{-}5/1.09e\text{-}4$	$0.204/0.412$	$0.122/0.526$	$0.012/0.632$	$0/0.861$
HPC3	No	308	0.001	$5.88e\text{-}6/2.35e\text{-}5$	$0.27/0.743$	$0.264/0.616$	$0.184/0.286$	$0.081/0.543$
	Yes	306	0.001	$5.7e\text{-}6/2.28e\text{-}5$	$0.272/0.778$	$0.262/0.616$	$0.172/0.251$	$0.067/0.53$

Table 12: Rejection rate of Algorithm 2 for second-order insecure SKINNY S-box designs.

	Circuit		**Model**	**Thresh.**	**Rejection rate** n, $\alpha_I/1-\beta_{II}$			
Gad.	Opt.?	$\lvert\mathcal{W}\rvert$	p	$\epsilon_{0,I}/\epsilon_{0,II}$	100	200	500	1000
DOM	No	173	0.01	$^{0.0195}/_{0.078}$	$^{0}/_{0.96}$	$^{0}/_{0.999}$	$^{0}/_{1}$	$^{0}/_{1}$
	Yes	171	0.01	$^{0.0184}/_{0.0738}$	$^{0}/_{0.953}$	$^{0}/_{0.999}$	$^{0}/_{1}$	$^{0}/_{1}$
HPC1	No	255	0.01	$^{0.0165}/_{0.0662}$	$^{0}/_{0.934}$	$^{0}/_{0.997}$	$^{0}/_{1}$	$^{0}/_{1}$
	Yes	253	0.01	$^{0.0165}/_{0.0658}$	$^{0}/_{0.926}$	$^{0}/_{0.998}$	$^{0}/_{1}$	$^{0}/_{1}$
HPC2	No	353	0.01	$^{0.0249}/_{0.0995}$	$^{0}/_{0.994}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$
	Yes	351	0.01	$^{0.025}/_{0.0999}$	$^{0}/_{0.995}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$
HPC3	No	302	0.01	$^{0.0145}/_{0.0579}$	$^{0}/_{0.877}$	$^{0}/_{0.996}$	$^{0}/_{1}$	$^{0}/_{1}$
	Yes	300	0.01	$^{0.0142}/_{0.0569}$	$^{0}/_{0.889}$	$^{0}/_{0.997}$	$^{0}/_{1}$	$^{0}/_{1}$
DOM	No	173	0.001	$^{1.56e-3}/_{6.23e-3}$	$^{0.136}/_{0.482}$	$^{0.02}/_{0.544}$	$^{0.001}/_{0.837}$	$^{0}/_{0.986}$
	Yes	171	0.001	$^{1.13e-3}/_{4.53e-3}$	$^{0.173}/_{0.296}$	$^{0.062}/_{0.505}$	$^{0.001}/_{0.754}$	$^{0}/_{0.954}$
HPC1	No	255	0.001	$^{1.57e-3}/_{6.29e-3}$	$^{0.153}/_{0.314}$	$^{0.05}/_{0.502}$	$^{0}/_{0.749}$	$^{0}/_{0.941}$
	Yes	253	0.001	$^{1.57e-3}/_{6.28e-3}$	$^{0.153}/_{0.29}$	$^{0.052}/_{0.528}$	$^{0.001}/_{0.754}$	$^{0}/_{0.958}$
HPC2	No	353	0.001	$^{3.18e-3}/_{1.27e-2}$	$^{0.042}/_{0.477}$	$^{0.012}/_{0.611}$	$^{0}/_{0.929}$	$^{0}/_{0.998}$
	Yes	351	0.001	$^{3.17e-3}/_{1.27e-2}$	$^{0.035}/_{0.462}$	$^{0.009}/_{0.623}$	$^{0}/_{0.936}$	$^{0}/_{0.999}$
HPC3	No	302	0.001	$^{1.58e-3}/_{6.32e-3}$	$^{0.098}/_{0.36}$	$^{0.058}/_{0.402}$	$^{0.002}/_{0.742}$	$^{0}/_{0.944}$
	Yes	300	0.001	$^{1.21e-3}/_{4.83e-3}$	$^{0.067}/_{0.512}$	$^{0.041}/_{0.488}$	$^{0.01}/_{0.67}$	$^{0.001}/_{0.9}$

Table 13: Rejection rate of Algorithm 2 for first-order insecure PRESENT S-box designs.

	Circuit		**Model**	**Thresh.**	**Rejection rate** n, $\alpha_I/1-\beta_{II}$			
Gad.	Opt.?	$\lvert\mathcal{W}\rvert$	p	$\epsilon_{0,I}/\epsilon_{0,II}$	100	200	500	1000
DOM	No	181	0.01	$^{0.0522}/_{0.209}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$
	Yes	88	0.01	$^{0.0239}/_{0.0958}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$
HPC1	No	288	0.01	$^{0.0364}/_{0.146}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$
	Yes	123	0.01	$^{0.0242}/_{0.0969}$	$^{0}/_{0.99}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$
HPC2	No	361	0.01	$^{0.0832}/_{0.333}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$
	Yes	148	0.01	$^{0.0537}/_{0.215}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$
HPC3	No	336	0.01	$^{0.0469}/_{0.188}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$
	Yes	139	0.01	$^{0.0299}/_{0.12}$	$^{0}/_{0.999}$	$^{0}/_{1}$	$^{0}/_{1}$	$^{0}/_{1}$
DOM	No	181	0.001	$^{1.26e-3}/_{5.03e-3}$	$^{0.001}/_{0.587}$	$^{0}/_{0.714}$	$^{0}/_{0.935}$	$^{0}/_{0.999}$
	Yes	88	0.001	$^{7.13e-4}/_{2.85e-3}$	$^{0.103}/_{0.304}$	$^{0.042}/_{0.623}$	$^{0.003}/_{0.834}$	$^{0}/_{0.982}$
HPC1	No	288	0.001	$^{1.32e-3}/_{5.28e-3}$	$^{0.004}/_{0.681}$	$^{0}/_{0.598}$	$^{0.001}/_{0.868}$	$^{0}/_{0.988}$
	Yes	123	0.001	$^{1.16e-3}/_{4.64e-3}$	$^{0.149}/_{0.518}$	$^{0.039}/_{0.599}$	$^{0}/_{0.856}$	$^{0}/_{0.991}$
HPC2	No	361	0.001	$^{4.78e-3}/_{0.0191}$	$^{0.002}/_{0.735}$	$^{0}/_{0.907}$	$^{0}/_{1}$	$^{0}/_{1}$
	Yes	148	0.001	$^{2.42e-3}/_{9.67e-3}$	$^{0.05}/_{0.46}$	$^{0.003}/_{0.746}$	$^{0}/_{0.98}$	$^{0}/_{1}$
HPC3	No	336	0.001	$^{2.28e-3}/_{9.12e-3}$	$^{0.003}/_{0.54}$	$^{0}/_{0.687}$	$^{0}/_{0.953}$	$^{0}/_{0.995}$
	Yes	139	0.001	$^{1.24e-3}/_{4.95e-3}$	$^{0.094}/_{0.499}$	$^{0.042}/_{0.522}$	$^{0.001}/_{0.848}$	$^{0}/_{0.988}$

Table 14: Rejection rate of Algorithm 2 for second-order secure PRESENT S-box designs.

	Circuit		**Model**	**Thresh.**	**Rejection rate** n, $\alpha_I/1-\beta_{II}$			
Gad.	Opt.?	$\lvert\mathcal{W}\rvert$	p	$\epsilon_{0,I}/\epsilon_{0,II}$	100	200	500	1000
DOM	No	432	0.01	1.47e-3/5.88e-3	0.221/0.204	0.046/0.409	0.003/0.68	0/0.904
	Yes	185	0.01	5.81e-4/2.32e-3	0.287/0.292	0.068/0.424	0.015/0.532	0/0.766
HPC1	No	612	0.01	2.39e-4/9.57e-4	0.678/0.678	0.46/0.449	0.155/0.406	0.015/0.462
	Yes	245	0.01	3.94e-4/1.58e-3	0.582/0.553	0.325/0.338	0.059/0.384	0.003/0.52
HPC2	Yes	341	0.01	0.00214/8.55e-3	0.103/0.332	0.017/0.492	0.001/0.78	0/0.975
	No	900	0.01	5.07e-4/2.03e-3	0.052/0.453	0.005/0.647	0/0.923	0/0.998
HPC3	Yes	317	0.01	5.79e-4/2.32e-3	0.525/0.546	0.293/0.302	0.036/0.385	0.005/0.577
	No	828	0.01	1.96e-5/7.84e-5	0.848/0.867	0.722/0.732	0.423/0.445	0.189/0.191
DOM	No	432	0.001	1.96e-5/7.85e-5	0.213/0.678	0.153/0.517	0.09/0.499	0.02/0.579
	Yes	185	0.001	1.24e-6/4.94e-6	0.269/0.786	0.416/0.644	0.263/0.29	0.079/0.361
HPC1	No	612	0.001	1.35e-5/5.41e-5	0.66/0.68	0.447/0.739	0.131/0.642	0.073/0.431
	Yes	245	0.001	1.27e-6/5.06e-6	0.586/0.801	0.351/0.808	0.372/0.577	0.263/0.345
HPC2	Yes	341	0.001	1.39e-5/5.58e-5	0.285/0.629	0.233/0.396	0.079/0.434	0.013/0.656
	No	900	0.001	3.35e-4/1.34e-3	0.032/0.44	0.019/0.527	0.001/0.74	0/0.939
HPC3	Yes	317	0.001	3.11e-6/1.24e-5	0.536/0.773	0.307/0.767	0.163/0.546	0.176/0.32
	No	828	0.001	3.43e-5/1.37e-4	0.828/0.872	0.715/0.722	0.425/0.432	0.195/0.193

Table 15: Rejection rate of Algorithm 2 for second-order insecure PRESENT S-box designs.

	Circuit		**Model**	**Thresh.**	**Rejection rate** n, $\alpha_I/1-\beta_{II}$			
Gad.	Opt.?	$\lvert\mathcal{W}\rvert$	p	$\epsilon_{0,I}/\epsilon_{0,II}$	100	200	500	1000
DOM	No	429	0.01	9.04e-3/0.0362	0.002/0.757	0/0.95	0/1	0/1
	Yes	182	0.01	0.0179/0.0715	0/0.948	0/0.999	0/1	0/1
HPC1	No	607	0.01	2.8e-3/0.0112	0.009/0.536	0.002/0.74	0/0.985	0/1
	Yes	240	0.01	0.0128/0.051	0/0.846	0/0.986	0/1	0/1
HPC2	No	897	0.01	1.2e-3/4.8e-3	0.002/0.673	0/0.918	0/1	0/1
	Yes	338	0.01	0.0226/0.0903	0/0.985	0/1	0/1	0/1
HPC3	Yes	311	0.01	0.0123/0.0494	0/0.844	0/0.983	0/1	0/1
	No	822	0.01	2.14e-4/8.58e-4	0.122/0.382	0.034/0.416	0.002/0.757	0/0.942
DOM	No	429	0.001	1.59e-3/6.37e-3	0.092/0.493	0.038/0.482	0.006/0.665	0/0.896
	Yes	182	0.001	1.55e-3/6.18e-3	0.155/0.491	0.045/0.559	0/0.809	0/0.968
HPC1	No	607	0.001	1.56e-3/6.25e-3	0.19/0.528	0.053/0.412	0.005/0.598	0/0.873
	Yes	240	0.001	7.35e-4/2.94e-3	0.039/0.635	0.043/0.451	0.005/0.625	0.001/0.806
HPC2	No	897	0.001	3.39e-3/0.0136	0.044/0.455	0.007/0.613	0/0.909	0/0.996
	Yes	338	0.001	3.11e-3/0.0124	0.055/0.453	0.012/0.615	0/0.933	0/0.998
HPC3	Yes	311	0.001	1.56e-3/6.22e-3	0.12/0.416	0.041/0.419	0/0.69	0/0.915
	No	822	0.001	1.53e-3/6.12e-3	0.182/0.502	0.078/0.404	0.005/0.65	0/0.869

C.2 Equivalence Test - Algorithm 3

For each table, Δ_I and Δ_{II} are chosen smaller and larger than the true difference of the lower bounds of the simulation failure probabilities of the considered circuits 1 and 2. We report the rejection for different numbers of drawn wire combinations n for $\alpha = 1\%$. In our tables, 'Sec.?' refers to whether we consider the secure or insecure circuit version and 'Opt.?' to whether we consider the non-optimized or optimized circuits (see Section 5).

We consider distinct circuits for the first-order SKINNY S-box with the HPC1 gadget in Table 16, HPC2 in Table 17 and HPC3 in Table 18; and all secure first-order circuits with different gadgets are compared in Table 19.

Likewise, we consider first-order PRESENT S-box with the HPC1 gadget in Table 20, HPC2 in Table 21 and HPC3 in Table 22; and the secure first-order circuits with the considered gadgets are analyzed in Table 23.

For the sake of brevity, we omit the the analysis for the second-order implementations.

Table 16: Rejection rate of Algorithm 3 for different implementations of the first-order SKINNY S-box with the HPC1 gadget.

Circuit 1		**Circuit 2**		**Model**		**Threshold**	**Rejection rate**		
Opt.?	Sec.?	Opt.?	Sec.?	p_1	p_2	Δ_I/Δ_{II}	$\alpha_I/1-\beta_{II}$ 200	$\alpha_I/1-\beta_{II}$ 500	$\alpha_I/1-\beta_{II}$ 1000
No	Yes	No	No	0.01	0.01	$0.017/0.068$	$0/0.958$	$0/1$	$0/1$
No	Yes	Yes	No	0.01	0.01	$0.0161/0.0646$	$0/0.95$	$0/1$	$0/1$
Yes	Yes	Yes	No	0.01	0.01	$0.0166/0.0665$	$0/0.96$	$0/1$	$0/1$
No	Yes	Yes	Yes	0.01	0.01	$4.95e\text{-}4/0.05$	$0/1$	$0/1$	$0/1$
No	Yes	No	No	0.001	0.001	$6.88e\text{-}4/0.05$	$0.019/1$	$0.004/1$	$0/1$
No	Yes	Yes	No	0.001	0.001	$6.74e\text{-}4/0.05$	$0.02/1$	$0.005/1$	$0/1$
Yes	Yes	Yes	No	0.001	0.001	$6.81e\text{-}4/0.05$	$0.026/1$	$0.004/1$	$0/1$
No	Yes	Yes	Yes	0.001	0.001	$7.43e\text{-}6/0.05$	$0/1$	$0/1$	$0/1$

Table 17: Rejection rate of Algorithm 3 for different implementations of the first-order SKINNY S-box with the HPC2 gadget.

Circuit 1		**Circuit 2**		**Model**		**Threshold**	**Rejection rate**		
Opt.?	Sec.?	Opt.?	Sec.?	p_1	p_2	Δ_I/Δ_{II}	$\alpha_I/1-\beta_{II}$ 200	$\alpha_I/1-\beta_{II}$ 500	$\alpha_I/1-\beta_{II}$ 1000
No	Yes	No	No	0.01	0.01	$0.0216/0.0864$	$0/0.903$	$0/1$	$0/1$
No	Yes	Yes	No	0.01	0.01	$0.0186/0.0744$	$0/0.809$	$0/0.999$	$0/1$
Yes	Yes	Yes	No	0.01	0.01	$0.0194/0.0775$	$0/0.868$	$0/0.998$	$0/1$
No	Yes	Yes	Yes	0.01	0.01	$7.92e\text{-}4/0.05$	$0/0.987$	$0/1$	$0/1$
No	Yes	No	No	0.001	0.001	$1.68e\text{-}3/0.05$	$0.019/1$	$0.001/1$	$0/1$
No	Yes	Yes	No	0.001	0.001	$8.01e\text{-}4/0.05$	$0/1$	$0/1$	$0/1$
Yes	Yes	Yes	No	0.001	0.001	$8.17e\text{-}4/0.05$	$0/1$	$0.001/1$	$0/1$
No	Yes	Yes	Yes	0.001	0.001	$1.6e\text{-}5/0.05$	$0/1$	$0/1$	$0/1$

Table 18: Rejection rate of Algorithm 3 for different implementations of the first-order SKINNY S-box with the HPC3 gadget.

Circuit 1		**Circuit 2**		**Model**		**Threshold**	**Rejection rate**		
Opt.?	Sec.?	Opt.?	Sec.?	p_1	p_2	Δ_I/Δ_{II}	$\alpha_I/1-\beta_{II}$ 200	$\alpha_I/1-\beta_{II}$ 500	$\alpha_I/1-\beta_{II}$ 1000
No	Yes	No	No	0.01	0.01	$0.0133/0.0532$	$0/0.786$	$0/0.993$	$0/1$
No	Yes	Yes	No	0.01	0.01	$0.0138/0.0551$	$0/0.802$	$0/0.994$	$0/1$
Yes	Yes	Yes	No	0.01	0.01	$0.0144/0.0576$	$0/0.83$	$0/0.999$	$0/1$
No	Yes	Yes	Yes	0.01	0.01	$6.32e\text{-}4/0.05$	$0/1$	$0/1$	$0/1$
No	Yes	No	No	0.001	0.001	$6.4e\text{-}4/0.05$	$0.021/1$	$0.005/1$	$0.001/1$
No	Yes	Yes	No	0.001	0.001	$1.06e\text{-}3/0.05$	$0.03/1$	$0/1$	$0/1$
Yes	Yes	Yes	No	0.001	0.001	$1.07e\text{-}3/0.05$	$0.032/1$	$0.002/1$	$0/1$
No	Yes	Yes	Yes	0.001	0.001	$9.83e\text{-}6/0.05$	$0/1$	$0/1$	$0/1$

Table 19: Rejection rate of Algorithm 3 for different implementations of secure first-order SKINNY S-box with various gadgets with $p_1 = p_2$.

Circuit 1		**Circuit 2**		**Model**		**Threshold**	**Rejection rate**		
Gad.?	Opt.?	Gad.?	Opt.?	p_1	p_2	Δ_I/Δ_{II}	$\alpha_I/1-\beta_{II}$ 200	$\alpha_I/1-\beta_{II}$ 500	$\alpha_I/1-\beta_{II}$ 1000
DOM	No	HPC1	No	0.01	0.01	$0.003/0.05$	$0/1$	$0/1$	$0/1$
DOM	No	HPC2	No	0.01	0.01	$0.0303/0.121$	$0/1$	$0/1$	$0/1$
DOM	No	HPC3	No	0.01	0.01	$6.99e\text{-}3/0.05$	$0/0.999$	$0/1$	$0/1$
HPC1	No	HPC2	No	0.01	0.01	$0.0273/0.109$	$0/1$	$0/1$	$0/1$
HPC1	No	HPC3	No	0.01	0.01	$3.99e\text{-}3/0.05$	$0/1$	$0/1$	$0/1$
HPC2	No	HPC3	No	0.01	0.01	$0.0233/0.0932$	$0/0.999$	$0/1$	$0/1$
DOM	Yes	HPC1	Yes	0.01	0.01	$3.13e\text{-}3/0.05$	$0/1$	$0/1$	$0/1$
DOM	Yes	HPC2	Yes	0.01	0.01	$0.0301/0.12$	$0/1$	$0/1$	$0/1$
DOM	Yes	HPC3	Yes	0.01	0.01	$6.98e\text{-}3/0.05$	$0/1$	$0/1$	$0/1$
HPC1	Yes	HPC2	Yes	0.01	0.01	$0.027/0.108$	$0/1$	$0/1$	$0/1$
HPC1	Yes	HPC3	Yes	0.01	0.01	$3.86e\text{-}3/0.05$	$0/1$	$0/1$	$0/1$
HPC2	Yes	HPC3	Yes	0.01	0.01	$0.0231/0.0926$	$0/0.999$	$0/1$	$0/1$

Table 20: Rejection rate of Algorithm 3 for different implementations of the first-order PRESENT S-box with the HPC1 gadget.

Circuit 1		**Circuit 2**		**Model**		**Threshold**	**Rejection rate**		
Opt.?	Sec.?	Opt.?	Sec.?	p_1	p_2	Δ_I/Δ_{II}	$\alpha_I/1-\beta_{II}$ 200	$\alpha_I/1-\beta_{II}$ 500	$\alpha_I/1-\beta_{II}$ 1000
No	Yes	No	No	0.01	0.01	$0.0107/0.05$	$0/0.658$	$0/0.971$	$0/1$
No	Yes	Yes	No	0.01	0.01	$1.47e\text{-}3/0.05$	$0/0.997$	$0/1$	$0/1$
Yes	Yes	Yes	No	0.01	0.01	$0.0133/0.0531$	$0/0.926$	$0/1$	$0/1$
No	Yes	Yes	Yes	0.01	0.01	$0.0148/0.059$	$0/0.927$	$0/1$	$0/1$
No	Yes	No	No	0.001	0.001	$7.27e\text{-}4/0.05$	$0/1$	$0/1$	$0.001/1$
No	Yes	Yes	No	0.001	0.001	$5.67e\text{-}4/0.05$	$0/1$	$0/1$	$0/1$
Yes	Yes	Yes	No	0.001	0.001	$1.04e\text{-}3/0.05$	$0.049/1$	$0/1$	$0/1$
No	Yes	Yes	Yes	0.001	0.001	$4.69e\text{-}4/0.05$	$0.003/1$	$0/1$	$0/1$

Table 21: Rejection rate of Algorithm 3 for different implementations of the first-order PRESENT S-box with the HPC2 gadget.

Circuit 1		**Circuit 2**		**Model**		**Threshold**	**Rejection rate**		
Opt.?	Sec.?	Opt.?	Sec.?	p_1	p_2	Δ_I/Δ_{II}	$\alpha_I/1-\beta_{II}$ 200	$\alpha_I/1-\beta_{II}$ 500	$\alpha_I/1-\beta_{II}$ 1000
No	Yes	No	No	0.01	0.01	8.87e-3/0.05	0/0.438	0/0.883	0/0.997
No	Yes	Yes	No	0.01	0.01	0.0206/0.0823	0/0.816	0/0.996	0/1
Yes	Yes	Yes	No	0.01	0.01	0.0229/0.0917	0/0.983	0/1	0/1
No	Yes	Yes	Yes	0.01	0.01	0.0435/0.174	0/1	0/1	0/1
No	Yes	No	No	0.001	0.001	1.98e-3/0.05	0/1	0/1	0/1
No	Yes	Yes	No	0.001	0.001	3.87e-4/0.05	0/1	0/1	0/1
Yes	Yes	Yes	No	0.001	0.001	2.05e-3/0.05	0.002/1	0/1	0/1
No	Yes	Yes	Yes	0.001	0.001	2.44e-3/0.05	0/1	0/1	0/1

Table 22: Rejection rate of Algorithm 3 for different implementations of the first-order PRESENT S-box with the HPC3 gadget.

Circuit 1		**Circuit 2**		**Model**		**Threshold**	**Rejection rate**		
Opt.?	Sec.?	Opt.?	Sec.?	p_1	p_2	Δ_I/Δ_{II}	$\alpha_I/1-\beta_{II}$ 200	$\alpha_I/1-\beta_{II}$ 500	$\alpha_I/1-\beta_{II}$ 1000
No	Yes	No	No	0.01	0.01	8.5e-3/0.05	0/0.669	0/0.987	0/1
No	Yes	Yes	No	0.01	0.01	8.5e-3/0.05	0/0.811	0/0.996	0/1
Yes	Yes	Yes	No	0.01	0.01	0.014/0.0561	0/0.881	0/1	0/1
No	Yes	Yes	Yes	0.01	0.01	0.0225/0.0901	0/0.998	0/1	0/1
No	Yes	No	No	0.001	0.001	1.09e-3/0.05	0/1	0/1	0/1
No	Yes	Yes	No	0.001	0.001	5.15e-5/0.05	0/1	0/1	0/1
Yes	Yes	Yes	No	0.001	0.001	1.05e-3/0.05	0.048/1	0/1	0/1
No	Yes	Yes	Yes	0.001	0.001	0.001/0.05	0.001/1	0/1	0/1

Table 23: Rejection rate of Algorithm 3 for different implementations of secure first-order PRESENT S-box with various gadgets with $p_1 = p_2$.

Circuit 1		**Circuit 2**		**Model**		**Threshold**	**Rejection rate**		
Gad.?	Opt.?	Gad.?	Opt.?	p_1	p_2	Δ_I/Δ_{II}	$\alpha_I/1-\beta_{II}$ 200	$\alpha_I/1-\beta_{II}$ 500	$\alpha_I/1-\beta_{II}$ 1000
DOM	No	HPC1	No	0.01	0.01	0.0141/0.0563	0/0.683	0/0.984	0/1
DOM	No	HPC2	No	0.01	0.01	0.0345/0.138	0/1	0/1	0/1
DOM	No	HPC3	No	0.01	0.01	0.00132/0.05	0/0.978	0/1	0/1
HPC1	No	HPC2	No	0.01	0.01	0.0486/0.194	0/1	0/1	0/1
HPC1	No	HPC3	No	0.01	0.01	0.0127/0.051	0/0.542	0/0.933	0/0.999
HPC2	No	HPC3	No	0.01	0.01	0.0358/0.143	0/0.999	0/1	0/1
DOM	Yes	HPC1	Yes	0.01	0.01	3.08e-4/0.05	0/1	0/1	0/1
DOM	Yes	HPC2	Yes	0.01	0.01	0.0202/0.0807	0/1	0/1	0/1
DOM	Yes	HPC3	Yes	0.01	0.01	5.29e-3/0.05	0/1	0/1	0/1
HPC1	Yes	HPC2	Yes	0.01	0.01	0.0199/0.0794	0/0.999	0/1	0/1
HPC1	Yes	HPC3	Yes	0.01	0.01	4.98e-3/0.05	0/1	0/1	0/1
HPC2	Yes	HPC3	Yes	0.01	0.01	0.0149/0.0595	0/0.952	0/1	0/1