IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2024, No. 4, pp. 657–689. DOI:10.46586/tches.v2024.i4.657-689

An Algebraic Approach for Evaluating Random Probing Security With Application to AES

Vahid Jahandideh, Bart Mennink and Lejla Batina

Radboud University, Nijmegen, The Netherlands
{v.jahandideh,b.mennink,lejla}@cs.ru.nl

Abstract. We employ an algebraic approach to estimate the success rate of a side-channel adversary attacking secrets of a masked circuit within the Random Probing Model (RPM), where intermediate variables of the implementation leak with a probability p. Our method efficiently handles masked linear circuits, enabling security bound estimation for practically large masking orders. For non-linear circuits, we employ a linearization technique. To reason about the security of complex structures like an S-box, we introduce a composition theorem, reducing the RPM security of a circuit to that of its constituent gadgets. Moreover, we lower the complexity of the multiplication gadget of CHES 2016 from $\mathcal{O}(n^2 \log(n))$ to $\mathcal{O}(n^2)$ while demonstrating its conjectured RPM security. Collectively, these novel methods enable the development of a practical masking scheme with $\mathcal{O}(n^2)$ complexity for AES, maintaining security for a considerably high leakage rate $p \leq 0.02 \approx 2^{-5.6}$.

Keywords: Random Probing Model, Masking, Side-Channel Protection, AES.

1 Introduction

Cryptographic systems deployed on physical devices inadvertently expose information about their internal operations through side-channel measurements, which can be exploited by an adversary. *Masking* [CJRR99] is among the techniques used to mitigate such leakages, and it can provide provable security. It works by randomizing internal operations and splitting their intermediates into *shares*. The aim of masking is to make it difficult for the adversary to reconstruct the original intermediates by observing leakage from individual shares.

To rigorously prove the effectiveness of masking, a *leakage model* is necessary. This model serves as a mathematical representation of side-channel leakage information. One such model is the *Threshold Probing Model* (TPM) introduced by Ishai et al. [ISW03]. In the TPM, the adversary is limited in the number of internal variables they can learn in each execution. This model has been widely used in research, resulting in numerous masking schemes, along with compositional theorems demonstrating how to securely compose masked subcircuits within this model. However, it is important to note that since the TPM restricts the adversary's knowledge to a fixed number of variables, it may not capture all possible leakage information. Consequently, despite having security proofs within this model, a masked implementation could still be vulnerable to successful side-channel attacks such as *horizontal* attacks [BCPZ16].

A decade after the introduction of the TPM, Prouff and Rivain presented the *Noisy Leakage Model* in their EUROCRYPT 2013 paper [PR13]. This model aimed to provide a more realistic representation of side-channel leakage but introduced complexities in evaluating the efficiency of masking countermeasures. Addressing this challenge, Duc et al. [DDF14] presented their work at EUROCRYPT 2014, demonstrating a *reduction* from the noisy leakage model to a more tractable *Random Probing Model* (RPM).

Licensed under Creative Commons License CC-BY 4.0.
Received: 2024-04-15 Accepted: 2024-06-15 Published: 2024-09-05

Within the RPM, the adversary learns each variable of the implementation independently with a probability p, often referred to as the *leakage rate*. This model offers a convenient bridge to TPM: a masking scheme proven secure against t probes in TPM will remain secure in the RPM if the adversary learns fewer than t intermediates. The probability of this event depends on the total number of circuit variables and p, and can be quantified using methods such as the Chernoff bound [DDF14].

Direct Versus Indirect Security Evaluation. The theoretical soundness of masking schemes is typically demonstrated by assessing the difficulty for adversaries to extract useful information about secret operands, such as bytes of a round key, in the targeted algorithm. Techniques for achieving this can be broadly categorized into two classes: *direct* and *indirect* methods [PGMP19].

An *indirect* security proof, also known as a *simulation-based* proof, renders the adversary's knowledge ineffective by proving that certain leakages are reproducible (via a *simulator*) from scratch, without accessing secret data, and thus carry no useful information. Indirect security proofs have been utilized in [ISW03, DDF14, BCP+20]. While these security claims offer more robust guarantees regarding the mitigation of side-channel leakage, they may sometimes be overly general, potentially overestimating the adversary's capabilities and resulting in loose practical relevance. As a main technical tool, simulation-based approaches typically require re-masking the circuit iteratively, and this significantly increases the size of the final circuit.

Direct security evaluations, on the other hand, align more closely with existing side-channel attacks and are simpler to comprehend and estimate. In this approach, the effectiveness of masking is demonstrated by how a success metric, such as *key rank* in a key guessing attack or the *mutual information* between a targeted secret and leakage, decreases with increasing masking order n. This approach has been employed in [CJRR99, PR13, MS23]. Using this approach, one can identify the amount of leakage contribution due to each component of the masked circuit. Hence, it provides a clearer understanding of the effectiveness of the deployed masking strategy.

1.1 Open Challenges in the RPM

Dependency of Leakage Rate on Masking Order. A primary challenge in this domain arises from the fact that, for typical masking schemes, security claims hold only if, with an increase in masking order n, the leakage rate p decreases. More precisely, a condition of $p < \frac{1}{n}$ is required for security [PR13, DDF14]. This condition may seem counterintuitive because, for high leakage rates, increasing the masking order might inadvertently benefit the adversary.

Battistello et al. [BCPZ16] illustrated that this requirement is not an artifact of the proof methods but is instead crucial for the security of the widely used ISW multiplication gadget [ISW03]. In essence, when multiplying two shared secrets X and Y at order n, n^2 intermediates are computed, with each share of secrets appearing in n intermediates. Therefore, the expected number of times a share gets revealed through leakage is pn, limiting the allowed leakage rate to $p < \frac{1}{n}$.

Dependency of Security Claims on Circuit Size. In the TPM, a masked circuit can maintain its security order regardless of the number of composing gadgets [BBD+16]. However, in the RPM, security claims tend to degrade with increasing circuit size, a phenomenon not entirely aligned with experimental results. For example, the success rate of a state-of-the-art side-channel attack does not effectively increase with the number of cipher rounds [KPP20]. The main reason for this discrepancy is the lack of compositional theorems inherent to the RPM.

In the case of simulation-based proofs, the RPM is translated to the TPM, limiting the expected number of leaked variables $p|\mathsf{C}|$ (with $|\mathsf{C}|$ denoting the size of the masked circuit) to some threshold t. The relationship $p|\mathsf{C}| < t$ demonstrates the dependency of the claimed security on $|\mathsf{C}|$. On the other hand, for direct proofs, security bounds are obtained assuming *leak-free* refresh gadgets. This assumption enables the use of union bounds (for mutual information) and allows for deriving a security claim by summing over all components of the masked circuit. Consequently, the claim derived in this manner will depend on $|\mathsf{C}|$.

1.2 Related Work

Ajtai [Ajt11] explored the design of theoretical compilers using *expander graphs* to enhance resistance to leakage in RPM. Andrychowicz et al. [ADF16] demonstrated the feasibility of achieving a constant leakage rate for linear circuits by leveraging the expander graphs framework.

In a practical context, Ananth et al. [AIS18] introduced the *expansion* method at CRYPTO 2018, which employs iterative masking to achieve constant leakage rates. Although this approach can tolerate leakage rates up to 2^{-25}, subsequent refinements by Belaïd et al. [BCP+20, BRT21, BRTV21] have reduced this margin to approximately 2^{-7}. However, the expansion approach still suffers from significant computational overhead, limiting its practical applicability.

Dziembowski et al. [DFŻ19], in their work on the *leakage diagram* technique [BFO23], demonstrated that certain existing masked gadgets are inherently secure against constant leakage p without requiring further compilation. Their study primarily focused on a linear refreshing gadget without any non-linear components, which inspired our work.

In CRYPTO 2021, Cassiers et al. [CFOS21] introduced the Probe Distribution Table (PDT) technique to evaluate the security of $\mathcal{O}(n^2)$ masking approaches based on the ISW method [RP10], particularly for protecting the S-box of AES. However, the ISW multiplication used in their masked circuit is not secure at constant leakage rates.

Prouff and Rivain [PR13] introduced the use of mutual information metric as a direct approach for assessing the security of masking without additional compilation techniques. This approach was further developed by Masure and Standaert [MS23] within the noisy leakage model. However, a limitation of mutual information-based approaches is their reliance on the existence of leak-free refresh gadgets [BCGR24].

1.3 Our Contribution

We contribute towards both of the RPM challenges (see Section 1.1) by proposing a new direct approach for estimating security in the RPM framework. We additionally propose a masking scheme that can withstand relatively high leakage rates. In more detail, our contributions are as follows:

- We start with an important observation that for linear masked circuits, leakage either fully discloses secret values or remains completely uninformative about them, and the probability of a disclosure event is independent of the specific leakage realization. We develop tools to efficiently estimate the probability of the disclosure event and express this probability as a function of leakage rate p and masking order n. See Section 3.

- Subsequently, in Section 4, aiming to reason about the RPM security in more complex structures, we look closely at refreshing gadgets and model their impact with an increased value of the leakage rate at their input/output interfaces. Notably, we do this without a leak-free refresh assumption. The result forms a *composition* theorem that is native to the RPM domain.

- As our next contribution, we pick the multiplication gadget of Battistello et al. [BCPZ16] that has a conjectured RPM security, and apply certain modifications to reduce its complexity from $\mathcal{O}(n^2 \log(n))$ to $\mathcal{O}(n^2)$ while demonstrating its RPM security at a specific range of (n, p) parameters. Our main technical tool is linearization, which paves the way for using the results already developed for linear circuits. These materials are treated in Section 5.
- The composition theorem and secure multiplication are enough to propose an $\mathcal{O}(n^2)$ masking scheme for the S-box of AES, which can withstand leakage rates of $p \leq 0.03$. The derived security bound depends on the number of gadgets that have a dominant contribution to the adversary's post-leakage information. See Section 6.
- Our final contribution is the extension of the RPM security evaluation to a full $\mathcal{O}(n^2)$ masked AES with guaranteed security for $p \leq 0.02$ leakage rates. See Section 7. Notably, the evaluated security bound exhibits no direct dependency on the overall number of gadgets.

1.4 Our Methodology

Our security evaluation technique is based on the *direct* approach to estimate the success probability of an adversary targeting secrets in a masked implementation. The presented security claims are derived using a combination of analytical methods and Monte Carlo approaches. Specifically, we use analytical methods to derive lemmas and theorems, while numerical bounds for specific gadgets are derived using the Monte Carlo approach. Here, we briefly highlight how we employ the Monte Carlo technique.

We assume that the probability of occurrence of an event bad depends on the leakage rate p and the masking order n, i.e., $\Pr(\mathsf{bad}) = f(n, p)$ for some function f. In cases where f is unknown or impractical to determine analytically, an alternative approach is to empirically estimate $f(n, p)$ by running N trials at the targeted (n, p) pair and recording the number of times bad occurs. Let N_{bad} denote the number of occurrences of bad. According to the Law of Large Numbers (LLN) [PP02], we have:

$$\Pr(\mathsf{bad}) = \lim_{N \to \infty} \frac{N_{\mathsf{bad}}}{N}. \tag{1}$$

The accuracy of empirical probability estimations depends on the number of trials N, and by choosing a sufficiently large N, the credibility of the estimation increases.

Limitation and Reliability of the Monte Carlo Approach. As a limitation, we note that it is not possible to measure the probability of event bad if it is less than $\frac{1}{N}$. Therefore, if in an experiment we obtain $P_{\mathsf{bad}} = \frac{N_{\mathsf{bad}}}{N} = 0$, we can only infer that $\Pr(\mathsf{bad}) < \frac{1}{N}$. For the reliability, we note that P_{bad} is itself a random variable with mean $\Pr(\mathsf{bad})$ and variance σ_N^2, where $\lim_{N \to \infty} \sigma_N = 0$. Using Chebyshev's inequality [PP02], for any positive $k \in \mathbb{R}$, we have:

$$\Pr\left(|\Pr(\mathsf{bad}) - P_{\mathsf{bad}}| \geq k\sigma_N\right) \leq \frac{1}{k^2}. \tag{2}$$

The bound demonstrates that P_{bad} can be made arbitrarily close to $\Pr(\mathsf{bad})$. For actual computation, an empirical value of σ_N can be used.

Interpolation of the Estimations. Having a collection of estimations $f(n, p)$ at various (n, p) pairs, we might be able to express the results with some function $g(n, p)$ as $\Pr(\mathsf{bad}) \leq g(n, p)$. This helps to derive explicit numerical bounds for specific compositions of gadgets. However, the interpretation of the results obtained following this approach should consider that $\Pr(\mathsf{bad}) \leq g(n, p)$ might not hold for the (n, p) pairs that are outside the tested region.

2 Preliminaries

Notation. Lowercase letters (e.g., x) represent secrets and single variables. Uppercase letters (e.g., X) denote a set of shares corresponding to a secret. Boldface letters (e.g., $\mathbf{X}$) represent matrices. Circuits/functions are denoted with Sans-serif font (e.g., X). To refer to the masked counterpart of an algorithm, we append $\mathbb{S}$ to its name (e.g., $\mathbb{S}\mathsf{X}$). To specify a range of elements in a matrix between a and b, we use $a : b$, and to denote all the elements, we use a colon (e.g., $\mathbf{X}(1, :)$ represents the first row). The notation $|X|$ returns the number of elements in X.

Masking Countermeasure. A *secret* u-bit variable $v \in \mathbb{F}_{2^u}$ is typically masked to enhance its side-channel security. This masking process, at *order* n, involves encoding v into a random n-tuple of shares, denoted as $V = \{v_1, \ldots, v_n\} \in \mathbb{F}_{2^u}^n$, with the condition that the bitwise XOR operation ($\oplus$) applied to all shares ($\oplus_{i=1}^n v_i$) equals to v. We refer to V as an n-sharing of *native* v. Any subset of V with at most $n-1$ elements is *independent* of v.

When deploying this masking approach to protect a targeted cipher, it is possible to encode the inputs independently. The primary challenge lies in developing routines for performing the intermediate operations over the shares while preserving the security of the native values. However, for a cipher described with a circuit C using *basic* logical gates, such as XOR and AND, a secure computation on shares can be achieved by replacing these gates with their masked counterparts, often called *gadgets*.

Gadgets. We denote the masked counterpart of a gate G as $\mathbb{S}\mathsf{G}$, where $\mathbb{S}\mathsf{G}$ is a *family* of circuits, each corresponding to a specific value of n. When G supports the additive property $\mathsf{G}(x_1 \oplus x_2) \oplus \mathsf{G}(0) = \mathsf{G}(x_1) \oplus \mathsf{G}(x_2)$, the architecture of $\mathbb{S}\mathsf{G}$ is relatively straightforward: for a native input x and native output $y = \mathsf{G}(x)$ with $\mathsf{G}\colon \mathbb{F}_{2^u} \to \mathbb{F}_{2^u}$, we set $y_1 = \mathsf{G}(x_1)$ and $y_i = \mathsf{G}(x_i) \oplus \mathsf{G}(0)$ for $2 \le i \le n$, to define an n-shared output $Y = \mathbb{S}\mathsf{G}(X)$ based on the input X.

However, the construction of an AND gadget, denoted as $\mathbb{S}$AND, is intricate. Numerous proposals for $\mathbb{S}$AND have been put forward in the literature [ISW03, BDF+17, CS20, WJZY23, BBP+17]. Nonetheless, none of them is proved to be secure in the RPM.

Remark 1. $\mathbb{S}$AND gadgets can be compiled to achieve RPM security using the expansion strategy developed in [BCP+20, BRT21]. It is worth noting that the compiled gadget will exhibit significantly higher computational complexity.

In addition to these gadgets, for some constructions, a refreshing gadget is employed at the interfaces of basic gadgets. A refreshing gadget, denoted as $\mathbb{S}\mathsf{R}$, takes as input an n-sharing X and incorporates fresh randomness to produce a new n-sharing for the same underlying native value x.

2.1 Random Probing Model

At order n, let $\Sigma_{\mathbb{S}\mathsf{C}} = \{x_1, x_2, \ldots, x_{m(n)}\}$ be the set of $m(n)$ variables involved in the computation of $\mathbb{S}\mathsf{C}$. By definition, $\Sigma_{\mathbb{S}\mathsf{C}}$ encompasses all shares of the native values of C. Within the scope of a single execution of $\mathbb{S}\mathsf{C}$, each x_i obtains a unique value. For simplicity, we assume that all these variables belong to the same field $\mathbb{F}_{2^u}$.

Definition 1 (Random Probing Leakage)**.** At a leakage rate $p \in [0, 1]$, we define random probing leakage as the process through which the adversary learns the values assigned to each variable in $\Sigma_{\mathbb{S}\mathsf{C}}$ via an independent *erasure channel* with parameter p.

Definition 2 (Erasure Channel)**.** An erasure channel is a probabilistic mapping $\phi\colon \mathbb{F}_{2^u} \to \{\mathbb{F}_{2^u}, \perp\}$, for a given parameter p, defined by the following relation:

$$\phi(x) = \begin{cases} x & \text{with probability } p, \\ \perp & \text{otherwise.} \end{cases} \tag{3}$$

Here, $\perp$ is a special symbol used to indicate the erasure of the input.

We denote the leakage that an adversary obtains as $\mathcal{L} = \phi(\Sigma_{\mathbb{SC}})$, which is a shorthand for $\mathcal{L}(n,p) = \{\phi_1(x_1), \ldots, \phi_{m(n)}(x_{m(n)})\}$, where the ϕ_is are independent.

MAP Adversary and RPM Security. A MAP adversary, also called a Bayesian adversary, uses the *Maximum A Posteriori* (MAP) probability to estimate a native v given leakage $\mathcal{L}(n,p)$. It outputs the *best estimate* for the value of v based on the following rule:

$$\widetilde{v} = \underset{\alpha \in \mathbb{F}_{2^u}}{\operatorname{argmax}} \Pr(v = \alpha \mid \mathcal{L}(n,p)). \tag{4}$$

This choice maximizes the *success rate* defined as $\Pr(\widetilde{v} = v^*)$ [SMY09], where v^* is the correct value of v [HRG14]. In this paper, we will only consider the MAP adversary.

Our main security metric is the *advantage* of the adversary over random guessing, which for a uniform v with $q = |\mathbb{F}_{2^u}|$ is:

$$\mathsf{Adv}_v(n,p) \triangleq \Pr(\widetilde{v} = v^*) - \frac{1}{q}. \tag{5}$$

We define RPM security based on $\mathsf{Adv}_v(n,p)$ for a targeted secret v. The definition is more aligned with experimental side-channel attacks, where the success rate in attacking a native value measures the effectiveness of a masking countermeasure.

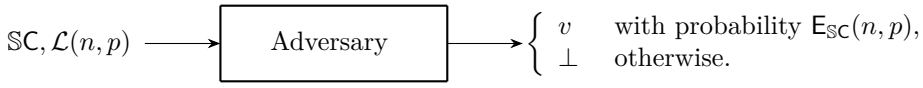
Definition 3 (RPM Security)**.** A circuit family $\mathbb{SC}$ that processes a native variable v is secure in the RPM framework if there exists a threshold p^o such that, given leakage $\mathcal{L} = \phi(\Sigma_{\mathbb{SC}})$ with $p \leq p^o$, $\mathsf{Adv}_v(n,p)$ monotonically decreases to 0 as n increases.

Remark 2. For a standalone v with an n-sharing V, since all the shares are required to reconstruct v, we have $\mathsf{Adv}_v(n,p) = [p^n + (1-p^n)\frac{1}{q}] - \frac{1}{q} = \frac{q-1}{q}p^n$. However, for a native v inside a masked cipher, the derivation of $\mathsf{Adv}_v(n,p)$ is more challenging.

For the circuits in this paper, if RPM security holds, we will express $\mathsf{Adv}_v(n,p)$ as $\mathsf{Adv}_v(n,p) \leq \alpha(\beta p)^{\gamma n}$, for some α, β, and $\gamma < 1$ constants. A lower $\mathsf{Adv}_v(n,p)$ will reduce the required order n and, consequently, decrease the computational overhead of masking.

Limitation of the Given Security Definitions. Definition 3 assumes that the adversary targets a single secret of the masked cipher. While this is often the case in practical attacks, from a theoretical perspective, the adversary might attempt to derive information about a function of multiple secrets. It is worth noting that most structures, such as S-boxes, have one secret. However, in a complete cipher such as AES, there are multiple secrets (such as the round keys of different rounds). Security definitions in simulation-based approaches [BCP+20] demonstrate that certain leakages can be reproduced from scratch and are thus independent of the secret. This approach does not make assumptions about the number of targeted secrets. However, by overestimating the adversary's capabilities, the resulting bounds may have limited practical relevance.

Equivalent Erasure Channel. For a linear $\mathbb{SC}$ processing a single native v, we will demonstrate that $\mathcal{L}(n,p)$ either reveals the value of v or provides no information about it. This phenomenon can be encapsulated using an Equivalent Erasure Channel (EEC) with a parameter $\mathsf{E}_{\mathbb{SC}}(n,p)$. We can consider the following setup.



When the adversary learns nothing, it still has the opportunity to guess the value of v. Therefore, we can derive the following expression for Adv_v:

$$\mathsf{Adv}_v(n,p) = \mathsf{E}_{\mathbb{SC}}(n,p) + \frac{1}{q}[1 - \mathsf{E}_{\mathbb{SC}}(n,p)] - \frac{1}{q} = \frac{q-1}{q}\mathsf{E}_{\mathbb{SC}}(n,p). \tag{6}$$

For non-linear masked circuits, using linearization technique, we will also develop an EEC. Nevertheless, these EECs will be approximations and will be valid only within certain limited leakage rates.

3 EEC for a Linear $\mathbb{SC}$

Let us consider a masked circuit $\mathbb{SC}$ processing a native variable v. In each execution of $\mathbb{SC}$, leakage will reveal some member of $\Sigma_{\mathbb{SC}}$ to the adversary. Native variable v has relationships with variables in $\Sigma_{\mathbb{SC}}$, and the adversary can leverage these dependencies, referred to as *parity relations*, to its advantage, obtaining information about v.

Definition 4 (Linear $\mathbb{SC}$)**.** We define a masked circuit $\mathbb{SC}$ as linear if all its natives and intermediaries belong to the same field $\mathbb{F}_{2^u}$, and the parity relations among $[v, \Sigma_{\mathbb{SC}}]$ are $\mathbb{F}_{2^u}$-linear.

For instance, the $\mathbb{SR}$-Simple refresh gadget [RP10] described in Algorithm 1 processes a single native and is linear.

Algorithm 1 $\mathbb{SR}$-Simple

Input $V^1 = (v_1^1, ..., v_n^1)$
Output $V^2 = (v_1^2, ..., v_n^2)$
$r_n = 0$
for $i = 1$ **to** $n-1$ **do**
  $r \xleftarrow{\$} \mathbb{F}_{2^u}$
  $v_i^2 = v_i^1 \oplus r$
  $r_n = r_n \oplus r$
$v_n^2 = v_n^1 \oplus r_n$
return V^2

In this section, our primary objective is to identify linear dependencies within $[v, \Sigma_{\mathbb{SC}}]$ and then to evaluate the extent to which they empower the adversary.

Extracting Parity Relations. For a linear $\mathbb{SC}$, the assumption that there are linear relations over $[v, \Sigma_{\mathbb{SC}}]$ implies the existence of a parity matrix $\mathbf{P}_n \in \mathbb{F}_{2^u}^{p(n) \times m(n)+1}$ such that, in each run, the values of $[v, \Sigma_{\mathbb{SC}}]$ satisfy

$$\mathbf{P}_n \cdot [v, \Sigma_{\mathbb{SC}}]^\top = \mathbf{0}_{[p(n)\times 1]}, \tag{7}$$

where $p(n)$ is the number of relations, and $^\top$ denotes matrix transpose. The rows of $\mathbf{P}_n$ are *linearly independent*, and any other parity relation over $[v, \Sigma_{\mathbb{SC}}]$ can be described as a linear combination of these rows.

Taking Leakage into Account. By receiving an instance L of leakage, some of the values in $\Sigma_{\mathbb{SC}}$ will be disclosed to the adversary. With substituting these leaked values, the system of equations defined in (7) transforms into:

$$\mathbf{P}_n^r \cdot [v, \Sigma_{\mathbb{SC}}]^\top = \mathbf{b}_{[p(n)\times 1]}. \tag{8}$$

Here, $\mathbf{b}$ (of size $p(n) \times 1$) is a known vector, and $\mathbf{P}_n^r$ is essentially $\mathbf{P}_n$ with the columns corresponding to the leaked variables replaced with all-zero. By finding the set of solutions of (8), the adversary can estimate the value of v.

Remark 3. In a finite field, a system of equations has either one, zero, or finitely many solutions.

For the system defined in (8), the realized values of $[v, \Sigma_{\mathbb{SC}}]$ will always be a solution. Hence, this system has either one or a finite number of solutions (based on the above remark). Let $\mathcal{S}$ be the set of unique solutions. Each $S_i \in \mathcal{S}$ is of length $m(n)+1$, and its first entry $S_i[1]$ corresponds to v. S_i also agrees with the leakage at the leaked variables. Based on $\mathcal{S}$, the probability distribution of $\widetilde{v}$ is

$$\Pr(\widetilde{v} = \alpha \mid \mathcal{L} = L) = \frac{|S_i \in \mathcal{S},\ S_i[1] = \alpha|}{|\mathcal{S}|}. \tag{9}$$

The adversary will consider $\operatorname{argmax}_{\alpha \in \mathbb{F}_{2^u}} \Pr(\widetilde{v} = \alpha \mid \mathcal{L} = L)$ as the realized value of v. To gain more insight into the shape of $\Pr(\widetilde{v} \mid \mathcal{L} = L)$, we process the system defined in (8) further by computing the *row-echelon form* [LDPDP06] of $\mathbf{P}_n^r$.

Let $\mathbf{G}$ be the row-echelon form of $\mathbf{P}_n^r$. $\mathbf{G}$ defines a new set of parity relations, which means that, for some constant vector $\mathbf{c}$ (that is computed from $\mathbf{b}$), we have:

$$\mathbf{G} \cdot [v, \Sigma_{\mathbb{SC}}]^\top = \mathbf{c}. \tag{10}$$

By construction, v will be a *pivot* variable and thus participate in only one relation, which is described by the first row of $\mathbf{G}$. If v is the sole variable in this relation, it can be uniquely identified. Otherwise, $\Pr(\widetilde{v} \mid \mathcal{L} = L)$ is uniform.

Lemma 1. *In an $\mathbb{F}_{2^u}$-linear circuit, if the first row of $\mathbf{G}$ does not include any variables other than v, the adversary can uniquely determine v. Otherwise, any other variable in the first row, which will be a* free *variable, completely conceals v.*

Proof. Let $[v, \Sigma_{\mathbb{SC}}]_p$ satisfy the non-homogeneous system defined by $\mathbf{G} \cdot [v, \Sigma_{\mathbb{SC}}]^\top = \mathbf{c}$, the set of solutions of this system can be described as:

$$[v, \Sigma_{\mathbb{SC}}] = [v, \Sigma_{\mathbb{SC}}]_p \oplus [v, \Sigma_{\mathbb{SC}}]_f, \tag{11}$$

where $[v, \Sigma_{\mathbb{SC}}]_f$ is any solution for the homogeneous system $\mathbf{G} \cdot [v, \Sigma_{\mathbb{SC}}]^\top = \mathbf{0}$. The number of $[v, \Sigma_{\mathbb{SC}}]_f$ tuples depends on the number of free variables in $\mathbf{G}$. Any value assigned to the free variables will give a new tuple $[v, \Sigma_{\mathbb{SC}}]_f$. Recall that columns corresponding to the instance of leakage are replaced with all-zero, and we do not consider them as free variables.

Computing $\mathbf{G}$ will start from the first column with a non-zero element. Since v is not part of leakage and there is at least one relation between v and $\Sigma_{\mathbb{SC}}$, v will be a pivot variable. Assume that $\mathbf{G}$ has l free variables, and label them as $\{z_1, \ldots, z_l\}$. The solution of v, based on (11), will be

$$v = v_p \oplus z_1 a_1 \oplus \ldots \oplus z_l a_l. \tag{12}$$

Here, v_p is the value of v in $[v, \Sigma_{\mathbb{SC}}]_p$, and a_i is the value of v in $\mathbf{G} \cdot [v, \Sigma_{\mathbb{SC}}]^\top = \mathbf{0}$ when all the free variables are set to zero, except z_i, which is set to 1. In (12), z_1 to z_l are free

to take any value in $\mathbb{F}_{2^u}$. Therefore, if at least one a_i is non-zero, irrespective of v_p, all values of v will have the same frequency.

To identify v uniquely, it is necessary to have all $a_i = 0$. In the row-echelon form, a_i for $1 < i \leq (m(n)+1)$ equals to $-\mathbf{G}(1,i)$. Therefore, if the condition $\mathbf{G}(1, 2 : m(n)+1) = \mathbf{0}$ is satisfied, then v is uniquely determined. Any non-zero value in $\mathbf{G}(1, 2 : m(n)+1)$ with its accompanying free variable will hide the particular solution v_p. □

Lemma 1 expresses the adversary's post-leakage information about v with an EEC, for which the parameter is defined as:

$$\mathsf{E}_{\mathbb{SC}}(n,p) = \Pr_{L \leftarrow \mathcal{L}(n,p)}[\mathbf{G}(1, 2 : m(n)+1) = \mathbf{0}_{[1 \times m(n)]}], \tag{13}$$

where each instance of leakage L yields a derived matrix $\mathbf{G}$ depending on the members of $\Sigma_{\mathbb{SC}}$ leaked to the adversary. Importantly, $\mathbf{G}$ and, consequently, $\mathsf{E}_{\mathbb{SC}}$ do not depend on the actual leakage values. This observation facilitates the efficient estimation of $\mathsf{E}_{\mathbb{SC}}$: for a fixed pair (n,p), with a sufficient number of trials, we can estimate $\mathsf{E}_{\mathbb{SC}}(n,p)$ by generating instances of leakage, obtaining the corresponding matrix $\mathbf{G}$, and counting the frequency of the event $\mathbf{G}(1, 2 : m(n)+1) = \mathbf{0}$. We provide the necessary algorithms and describe the technical details in the following section.

3.1 Recovering Parity Relations and Estimating $\mathsf{E}_{\mathbb{SC}}$

This section presents algorithms for extracting a parity matrix from a given $\mathbb{F}_{2^u}$-linear circuit, computing the row-echelon form of this parity matrix along with an instance of leakage, and approximating the corresponding E parameter.

Packing $[v, \Sigma_{\mathbb{SC}}]$ Values in $\mathbf{M}_n$. For a circuit family $\mathbb{SC}$ processing a native value v at the specified order n, we can construct a matrix $\mathbf{M}_n$ consisting of $m(n)+1$ columns and R rows. The process of populating $\mathbf{M}_n$ involves running $\mathbb{SC}$ a total of R times, each time with a randomly selected v and fresh randomness. During each run, the values assigned to v and $\Sigma_{\mathbb{SC}}$ fill a new row of $\mathbf{M}_n$, as outlined in Algorithm 2. Recall that $\Sigma_{\mathbb{SC}}$ is the list of variables of $\mathbb{SC}$, and $m(n)$ is the number of elements of this list.

Algorithm 2 Create-M

Input $\mathbb{SC}$, n
Output $\mathbf{M}_n$

1: **for** $i = 1$ **to** R **do**
2: $\quad v \xleftarrow{\$} \mathbb{F}_{2^u}$
3: $\quad \mathbf{M}_n(i,1) = v$
4: $\quad$ Execute $\mathbb{SC}$ ▷ With fresh shares for the native inputs
5: $\quad \mathbf{M}_n(i, 2 : m(n)+1) = \Sigma_{\mathbb{SC}}$
6: **return** $\mathbf{M}_n$

Example 1. In the case of $\mathbb{SR}$-Simple, as outlined in Algorithm 1, when $n = 3$, the set of variables $\Sigma_{\mathbb{SR}\text{-Simple}}$ comprises $\{v_1^1, v_2^1, v_3^1, r_3, r_a, v_1^1 \oplus r_a, r_3 \oplus r_a, r_b, v_2^1 \oplus r_b, r_3 \oplus r_b, v_3^1 \oplus r_3\}$. Here, r_a and r_b represent the values of the randomness variable r at two different iterations of the for loop. The value of r_3 in the list updates according to the algorithm's execution. It's important to note that the actual composition of $\Sigma_{\mathbb{SR}\text{-Simple}}$ may vary based on specific implementation details. For instance, if the output is copied elsewhere, $\{v_1^2, v_2^2, v_3^2\}$ may also become part of $\Sigma_{\mathbb{SR}\text{-Simple}}$. Recall that elements of $\Sigma_{\mathbb{SC}}$ leak independently. Therefore, a variable appearing multiple times in the list will have a higher probability of leaking.

The $\mathbf{M}_n$ matrix thus formed consists of $R \times (m(n)+1)$ elements from $\mathbb{F}_{2^u}$. In the following, we discuss the utilization of standard Gaussian elimination to extract a set of parity relations capable of describing $\mathbb{F}_{2^u}$-linear dependencies among these variables.[1]

Extracting $\mathbf{P}_n$ From $\mathbf{M}_n$. By computing the row-echelon form of $\mathbf{M}_n$ using Algorithm 3 ($\mathbf{D}_n = \mathsf{Gaussian\text{-}Elim}(\mathbf{M}_n)$), we can determine the linear dependencies between columns of $\mathbf{M}_n$. Since we are using numerical realizations for random variables, the output of the algorithm might be incomplete or incorrect. That is, we might miss some parity relations or include wrong parity relations. However, the accuracy and completeness of the recovered parity relations improve as the number of rows in $\mathbf{M}_n$ increases.

While the exact number of parity relations, denoted as $p(n)$, is initially unknown, we can deduce that $p(n) = m(n) + 1 - \mathsf{rank}(\mathbf{M}_n)$, and consequently, $p(n) \leq m(n)+1$. Hence, we need to set $R > m(n)$ to extract $p(n)$ linearly independent parity relations.

On the other hand, to minimize the probability of a set of columns summing to zero without being part of parity relations, we need to add more rows to $\mathbf{M}_n$. For instance, by setting $R > m(n) + W$, the probability that at least one of the $2^{m(n)+1} - p(n)$ possible incorrect parity relations among the columns satisfies all the rows will be approximately upper bounded by 2^{-uW}, where W is a positive integer, and u denotes the bit-width of the underlying field. It is important to note that the rows of $\mathbf{M}_n$ are valued using fresh randomness.

Algorithm 3 Gaussian-Elim

Input $\mathbf{M}_n$
Output Row-echelon form of $\mathbf{M}_n$

$pivot\_row = 1$ ▷ Initialize the pivot row
for $col = 1$ **to** $m(n)+1$ **do**
  for $row = pivot\_row$ **to** R **do**
    if $\mathbf{M}_n(row, col) \neq 0$ **then**
      $\mathbf{M}_n(row, :) = \mathbf{M}_n(row, :)/\mathbf{M}_n(row, col)$ ▷ Normalize to 1
      $\mathbf{M}_n(pivot\_row, :) \leftrightarrows \mathbf{M}_n(row, :)$ ▷ Swap the rows
      for $i = 1$ **to** R , $i \neq pivot\_row$ **do**
        $\mathbf{M}_n(i, :) = \mathbf{M}_n(i, :) - \frac{\mathbf{M}_n(i,col)}{\mathbf{M}_n(pivot\_row,col)} \mathbf{M}_n(pivot\_row, :)$
      $pivot\_row = pivot\_row + 1$ ▷ Increase the pivot row
      break
$\mathbf{D}_n = \mathbf{M}_n$
return $\mathbf{D}_n$

Within $\mathbf{D}_n$, there are two types of variables: *pivot* and *free*. Each free variable is associated with a parity relation, which is a set of columns in $\mathbf{D}_n$ whose sum equals zero. To extract these parity relations, we employ Algorithm 4 ($\mathbf{P}_n = \mathsf{Extract\text{-}Linear}(\mathbf{D}_n)$).

Example 2. Consider the following, where $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{P}$ are matrices in a binary field:

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \xrightarrow{\mathsf{Gaussian\text{-}Elim}} \mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\mathsf{Extract\text{-}Linear}} \mathbf{P} = \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix}. \quad (14)$$

The first three columns of $\mathbf{D}$ are pivot columns, but the fourth one is a free column. This column, specified in blue, is the sum of columns 1 and 3, as reflected in $\mathbf{P}$.

[1] Alternatively, one might derive a set of parity equations directly between $[v, \Sigma_{\mathbb{SC}}]$ from the defining relations of $\mathbb{SC}$. While this approach may result in a different representation for the parity matrix, the final value obtained for $\mathsf{E}_{\mathbb{SC}}$ remains indifferent to the particular representation chosen.

Algorithm 4 Extract-Linear

Input $\mathbf{D}_n$
Output Parity matrix $\mathbf{P}_n$

$row_D = 1$, $row_P = 1$
for $col_D = 1$ **to** $m(n) + 1$ **do**
  if $\mathbf{D}_n(row_D, col_D) = 0$ & $\mathbf{D}_n(:, col_D) \neq \mathbf{0}$ **then** ▷ A free variable
    $\mathbf{P}_n(row_P, col_D) = 1$ ▷ Add a row to $\mathbf{P}_n$
    for $j = 1$ **to** $row_D - 1$ **do**
      $\mathbf{P}_n(row_P, \mathsf{Pivots}(j)) = \mathbf{D}_n(j, col_D)$
    $row_P = row_P + 1$
  elseif $\mathbf{D}_n(:, col_D) \neq \mathbf{0}$ **then**
    $\mathsf{Pivots}(row_D) = col_D$ ▷ Record pivot column of each row
    $row_D = row_D + 1$;
return $\mathbf{P}_n$

Sampling $\mathbf{P}_n^r$ From $\mathbf{P}_n$. $\mathbf{P}_n$ encompasses all the linear dependencies of $\mathbb{SC}$, with its number of rows equaling the number of these relations. The next step is to sample leaking members of $\Sigma_{\mathbb{SC}}$ and derive the corresponding system of equations as in (8). To sample an instance of $\mathbf{P}_n^r$, we proceed as follows: First, $\mathbf{P}_n^r$ is set to $\mathbf{P}_n$. Then, each column (apart from column 1) of $\mathbf{P}_n^r$ is randomly and independently set to all-zero with probability p, indicating that the intermediate associated with this column has leaked.

Remark 4. In sampling $\mathbf{P}_n^r$, the decision for each column is made separately. Consequently, it is possible to consider non-identical values of the leakage rate p for different members of $\Sigma_{\mathbb{SC}}$. In practice, side-channel noise over different intermediates may not be equal, and hence, their corresponding p values will be dissimilar [DFS15].

Estimating $\mathsf{E}_{\mathbb{SC}}$ From Sampled $\mathbf{P}_n^r$s. Algorithm 5 outlines the steps to approximate $\mathsf{E}_{\mathbb{SC}}$ using N number of trials. The quality of the approximation improves as the value of N increases. For the numerical results presented, we used a step size of 10^{-2} for sweeping p values.

Algorithm 5 Approximate $\mathsf{E}_{\mathbb{SC}}$

Input The family $\mathbb{SC}$
Output $\mathsf{E}_{\mathbb{SC}}(n, p)$

for $n = 2$ **to** $n_{\max}$ **do**
  $\mathbf{M}_n = \mathsf{Create\text{-}M}(\mathbb{SC}, n)$
  $\mathbf{D}_n = \mathsf{Gaussian\text{-}Elim}(\mathbf{M}_n)$
  $\mathbf{P}_n = \mathsf{Extract\text{-}Linear}(\mathbf{D}_n)$
  for $p = p_{\min}$ **to** $p_{\max}$ **do**
    $N_{\mathsf{recovered}} = 0$
    for $i = 0$ **to** N **do**
      $\mathbf{P}_n^r \leftarrow \mathsf{Sampl}(\mathbf{P}_n, p)$
      $\mathbf{G} = \mathsf{Gaussian\text{-}Elim}(\mathbf{P}_n^r)$
      if $\mathbf{G}(1, 2 : m(n) + 1) = \mathbf{0}$ **then** ▷ The secret is recovered
        $N_{\mathsf{recovered}} = N_{\mathsf{recovered}} + 1$
    $\mathsf{E}_{\mathbb{SC}}(n, p) = \frac{N_{\mathsf{recovered}}}{N}$ ▷ Approximation with empirical mean
return $\mathsf{E}_{\mathbb{SC}}$

After obtaining a table of empirical mean values for $\mathsf{E}_{\mathbb{SC}}(n, p)$, efforts are made to bound them, if possible, such that $\mathsf{E}_{\mathbb{SC}}(n, p) \leq \alpha(\beta p)^{\gamma n}$, where α, β, and γ are constants with $\gamma < 1$. This expression facilitates the deployment of the results obtained for various gadgets in the security evaluation of more complex circuits, such as an S-box. Moreover, deriving a single expression that fits all the tested (n, p) pairs enhances the credibility of estimations. However, we note that the derived bound may not hold out of the tested region.

3.2 Case Study on Linear Gadgets

The serial composition of k (linear) $\mathbb{S}\mathsf{R}$ gadgets, as depicted in Figure 1, represents an example of a linear $\mathbb{S}\mathsf{C}$ with a single native variable. As highlighted by Dziembowski et al. [DFŻ19], this construction is not only theoretically significant but also holds practical relevance. We will also make use of the results for the case where $k = 1$ later.

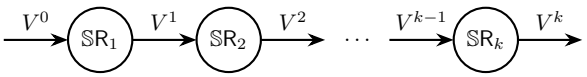


Figure 1: Multiple refresh gadgets working in tandem.

Instances of $\mathbb{S}\mathsf{R}$ include $\mathbb{S}\mathsf{R}$-Simple [RP10], $\mathbb{S}\mathsf{R}$-SNI [BBD+16], and $\mathbb{S}\mathsf{R}$-Rot [BDF+17] that are described in Algorithms 1, 6, and 7, respectively. The input to these algorithms is $V^1 = \{v_1^1, \ldots, v_n^1\}$, and the output is $V^2 = \{v_1^2, \ldots, v_n^2\}$.

Algorithm 6 $\mathbb{S}\mathsf{R}$-SNI

Input $V^1 = (v_1^1, \ldots, v_n^1)$
Output $V^2 = (v_1^2, \ldots, v_n^2)$

for $i = 1$ **to** n **do**
  for $j = i + 1$ **to** n **do**
    $r \xleftarrow{\$} \mathbb{F}_{2^u}$
    $v_i^1 = v_i^1 \oplus r$
    $v_j^1 = v_j^1 \oplus r$
$V^2 = V^1$
return V^2

Algorithm 7 $\mathbb{S}\mathsf{R}$-Rot

Input $V^1 = (v_1^1, \ldots, v_n^1)$
Output $V^2 = (v_1^2, \ldots, v_n^2)$

for $i = 1$ **to** n **do**
  $r_i \xleftarrow{\$} \mathbb{F}_{2^u}$
for $i = 1$ **to** 2 **do**
  for $j = 1$ **to** n **do**
    $r = r_{(i+j-2)\%n+1}$
    $v_j^1 = v_j^1 \oplus r$
$V^2 = V^1$
return V^2

$\mathbb{S}\mathsf{R}$-Simple requires the least amount of fresh randomness and has the least computational complexity. In contrast, $\mathbb{S}\mathsf{R}$-SNI involves more fresh randomness and is computationally more demanding. However, the additional randomness and increased processing complexity do not make $\mathbb{S}\mathsf{R}$-SNI more secure (based on the $\mathsf{E}_{\mathbb{S}\mathsf{R}}$ value) compared to $\mathbb{S}\mathsf{R}$-Simple and $\mathbb{S}\mathsf{R}$-Rot. For any tuple $(k, n > 2, p)$ studied, $\mathsf{E}_{\mathbb{S}\mathsf{R}\to\ldots\to\mathbb{S}\mathsf{R}}(n, p)$ is higher with $\mathbb{S}\mathsf{R}$-SNI than with the other two. Refer to the results in Figure 2. Nonetheless, $\mathbb{S}\mathsf{R}$-SNI shows good properties that help to prove the RPM security of combinations of gadgets. This point will be treated in Section 4.

Remark 5. Within our tested range ($n \leq 30$), $\mathbb{S}\mathsf{R}$-Simple and $\mathbb{S}\mathsf{R}$-Rot exhibit RPM security across the entire range of p. However, $\mathbb{S}\mathsf{R}$-SNI was found to be secure only for $p < 0.45$. For rates higher than this threshold, $\mathsf{E}_{\mathbb{S}\mathsf{R}\text{-}\mathsf{SNI}}(n, p)$ increases with the masking order n.

Estimating $\mathsf{E}_{\mathbb{S}\mathsf{R}\text{-}\mathsf{SNI}}(n, p)$. The value of $\mathsf{E}_{\mathbb{S}\mathsf{R}\text{-}\mathsf{SNI}}(n, p)$ is used for the security evaluation of certain circuits in this paper. Using Algorithm 5, we can empirically estimate $\frac{1}{n}\log_{10}(\mathsf{E}_{\mathbb{S}\mathsf{R}\text{-}\mathsf{SNI}}(n, p))$ for various orders and leakage rates. These estimations for the range $p < 0.15$ are plotted in Figure 3. From these results, it becomes apparent that there exists a value γ such that $\frac{1}{n}\log_{10}(\mathsf{E}_{\mathbb{S}\mathsf{R}\text{-}\mathsf{SNI}}(n, p)) \leq \gamma \log_{10}(p)$. By setting $\gamma = 0.6$, we satisfy this bound for the range $p < 0.15$. Hence, we can derive the following expression for $\mathsf{E}_{\mathbb{S}\mathsf{R}\text{-}\mathsf{SNI}}(n, p)$, which is valid for $(n \geq 2, p < 0.15)$:

$$\mathsf{E}_{\mathbb{S}\mathsf{R}\text{-}\mathsf{SNI}}(n, p) \leq p^{0.6n}. \tag{15}$$

Upon closer inspection of the results in Figure 3, it becomes evident that for lower leakage rates, we can derive tighter bounds for $\mathsf{E}_{\mathbb{S}\mathsf{R}\text{-}\mathsf{SNI}}(n, p)$ by choosing larger values for γ. Additionally, we observe that as p increases, γ should decrease. However, $\mathsf{E}_{\mathbb{S}\mathsf{R}\text{-}\mathsf{SNI}}(n, p)$

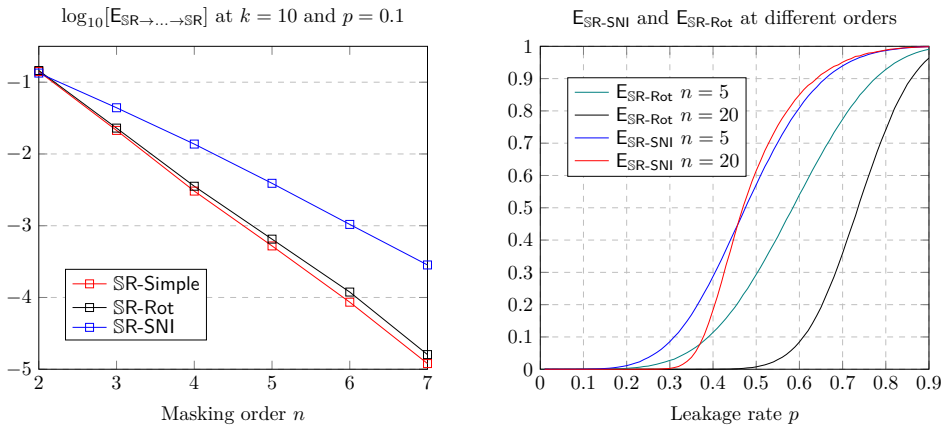


Figure 2: (Right) For higher leakage rates (at around $p > 0.45$), $\mathsf{E}_{\mathbb{SR}\text{-SNI}}$ increases with the order n, implying that the adversary can, with a higher probability, recover the native variable v.

in the range $p \leq 0.15$ is a decreasing function of n. We illustrate this with depictions of the estimated $\log_{10}(\mathsf{E}_{\mathbb{SR}\text{-SNI}}(n,p))$ in Figure 3-(Right).

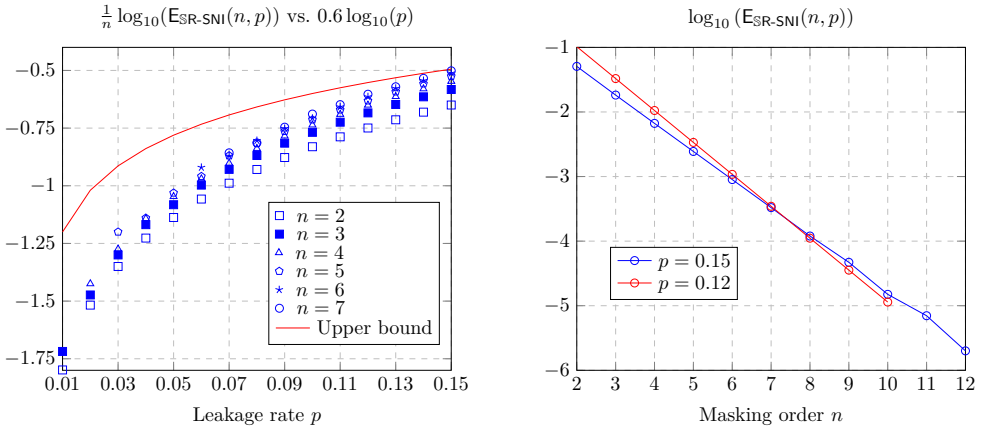


Figure 3: (Left) Estimation results for $\mathsf{E}_{\mathbb{SR}\text{-SNI}}$. (Right) The slope of decay of $\log_{10}(\mathsf{E}_{\mathbb{SR}\text{-SNI}}(n,p))$ is independent of order n, but depends on leakage rate p.

Remark 6. By deploying more complex functions of (n,p), one can derive tighter expressions for $\mathsf{E}_{\mathbb{SR}\text{-SNI}}(n,p)$. However, for the illustration of our proposed methodology, the given upper bound in (15) suffices.

As the order of masking increases, empirical estimation of a probability that exponentially decays with n becomes more challenging. This difficulty arises not only because the size of gadgets such as $\mathbb{SR}$-SNI grows in $\mathcal{O}(n^2)$, limiting the number of trials that can be performed in terms of computational power, but also because the events become rare and their probability falls below $\frac{1}{N}$. Employing Algorithm 5 with $N = 10^8$ trials, we conclude that for our tested range $n \leq 30$, no violation of the bound given in (15) was observed. We also note that there is no guarantee that the derived bound holds outside the tested region of n values. Specifically, for understanding the behavior of $\mathsf{E}_{\mathbb{SR}\text{-SNI}}(n,p)$ at $n \to \infty$,

we need rigorous analytical approaches, which we leave for future research.

4 Reduction in RPM Framework

Suppose $\mathbb{SG}_1$ and $\mathbb{SG}_2$, that both can be non-linear, are processing the same native variable v sequentially. Instead of directly passing the output V^0 from $\mathbb{SG}_1$ to $\mathbb{SG}_2$, there is an intermediary (linear) $\mathbb{SR}$ gadget. See Figure 4. The adversary's objective is to exploit leakages of the three gadgets and estimate the value of v.

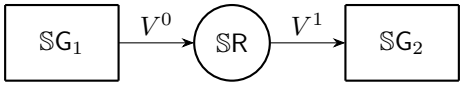


Figure 4: Composition using a refresh gadget.

We will transfer leakage of $\mathbb{SR}$ to its input/output interfaces and thereby *reduce* the security of this chain to the security of its composing gadgets. More precisely, we propose the following theorem.

Theorem 1. *For a bounded region of p values, the gadgets, and hence the composition, behave as an erasure channel for which*

$$\mathsf{E}_{\mathbb{SG}_1 \to \mathbb{SR} \to \mathbb{SG}_2}(n,p) \leq \mathsf{E}_{\mathbb{SG}_1}(n,p') + \mathsf{E}_{\mathbb{SR}}(n,p) + \mathsf{E}_{\mathbb{SG}_2}(n,p') \tag{16}$$

holds, where $p' \geq p$ is a function of (n,p) and the structure of $\mathbb{SR}$.

This reduction is a significant step toward estimating the RPM security of more intricate compositions. The remainder of this section consists of technical steps to derive (16). Before delving into details, let us briefly develop some intuition about the impact of internal leakage of $\mathbb{SR}$.

4.1 Reduction With Leak-Free $\mathbb{SR}$ Gadget

By assuming that the $\mathbb{SR}$ gadget is leak-free (i.e., an adversary cannot probe its internal variables), we can decompose the security of the construction $\mathbb{SG}_1 \to \mathbb{SR} \to \mathbb{SG}_2$ in mutual information metric to the security of $\mathbb{SG}_1$ and $\mathbb{SG}_2$. A leak-free $\mathbb{SR}$ gadget is required to preserve the conditional independence of V^0 and V^1, which is critical for the proofs in [PR13, MS23, BCGR24]. However, the $\mathbb{SR}$ gadget (like the other $\mathbb{SG}$s) is a piece of code, and the adversary can access its internal variables like any other variable of the entire masked algorithm. Therefore, the requirement of leak-free $\mathbb{SR}$ gadgets is not practically sound.

Effectively incorporating the leakage from $\mathbb{SR}$ gadgets, instead of neglecting it, is a long-standing challenge in the RPM. As a contribution to this problem, we provide a relatively tight bound on the additional benefit the adversary can gain from the leakage of an $\mathbb{SR}$ gadget in our security metric.

Effects of Leakage From $\mathbb{SR}$ Gadgets. In the following, we intuitively describe the two effects that leakage from $\mathbb{SR}$ gadgets causes:

1. Since the secret v depends on $\Sigma_{\mathbb{SR}}$, if the adversary learns a significant portion of the variables of $\Sigma_{\mathbb{SR}}$, it may uncover the secret v directly with this knowledge.
2. As an indirect effect, leakage of $\Sigma_{\mathbb{SR}}$ leads to the emergence of extra parity relations between V^0 and V^1. These new equations interconnect the parity relations of $\mathbb{SG}_1$ and $\mathbb{SG}_2$, thereby reducing the RPM security of the entire structure.

4.2 A Describing System of Equations

Let the following set of equations define $\mathbb{SG}_1 \to \mathbb{SR} \to \mathbb{SG}_2$, where the boundary variables, V^0 and V^1, are separated from the list of internal variables, and $\mathtt{NL}$ (resp. $\mathtt{L}$) denotes a non-linear (resp. linear) set of equations:

$$\mathtt{A}_n = \begin{cases} \mathtt{NL}_{\mathbb{SG}_1}(\Sigma_{\mathbb{SG}_1}, V^0, v) = \mathbf{0}, \\ \mathtt{L}_{\mathbb{SR}}(\Sigma_{\mathbb{SR}}, V^0, V^1, v) = \mathbf{0}, \\ \mathtt{NL}_{\mathbb{SG}_2}(\Sigma_{\mathbb{SG}_2}, V^1, v) = \mathbf{0}. \end{cases} \tag{17}$$

Our primary focus will be on the structure of the linear subsystem; thus, it is highlighted with a different color to assist following the operations. We denote the system with $\mathtt{A}_n$ to emphasize the dependency of the relations on n. We have deviated slightly from our notation: for the gadgets, we have excluded boundary variables, i.e., V^0, V^1 or both, from the corresponding Σ sets. Hence, $\Sigma_{\mathbb{SG}_1}$, $\Sigma_{\mathbb{SR}}$, and $\Sigma_{\mathbb{SG}_2}$ contain only the internal variables. This way, we avoid accounting for the leakage of boundary variables twice.

Remark 7. Joint equation sets in $\mathtt{A}_n$ empower the adversary more than separated subsets. As an extreme case, the three sets combined may uniquely determine the secret v, but v appears uniformly random while considering each system alone.

For the subsequent discussions, we place two requirements on the gadgets: *symmetry* over the shares for all the gadgets and *maximal independence* for the $\mathbb{SR}$.

Definition 5 (Symmetry)**.** A gadget is symmetric if any reordering of the shares inside it does not change the parity description.

Definition 6 (Maximal Independence)**.** In an $\mathbb{SR}$ gadget, input V^0 and output V^1 are *maximally independent* if, apart from $\oplus_{i=1}^n v_i^0 = \oplus_{i=1}^n v_i^1$, no other relation exists among the input and output shares.

For instance, $\mathbb{SR}$-SNI and $\mathbb{SR}$-Rot are symmetric, but $\mathbb{SR}$-Simple is not because the equation defining the last share differs from that of the other shares. For these two $\mathbb{SR}$s, input and output shares are maximally independent.[2]

Example 3. To clarify these definitions, we provide a simple instance of a refresh gadget that is symmetric but does not have maximally independent input-output shares. For $n = 4$, consider a refresh gadget that operates as follows:

$$\begin{aligned} (v_1^1, v_2^1, v_3^1, v_4^1) &\leftarrow (v_1^0, v_2^0, v_3^0, v_4^0) \oplus (r_1, r_2, r_3, r_4), \\ (v_1^1, v_2^1, v_3^1, v_4^1) &\leftarrow (v_1^1, v_2^1, v_3^1, v_4^1) \oplus (r_2, r_1, r_4, r_3). \end{aligned} \tag{18}$$

Here, r_1 to r_4 are randomness variables, and the $\oplus$ operation for two vectors is element-wise. While it is clear that $\oplus_{i=1}^4 v_i^0 = \oplus_{i=1}^4 v_i^1$, this is not the only relation. Indeed, we additionally have $v_1^0 \oplus v_2^0 = v_1^1 \oplus v_2^1$ and $v_3^0 \oplus v_4^0 = v_3^1 \oplus v_4^1$.

Definition 7 (Primary Solution)**.** $\mathtt{A}_n$ in each run of the underlying circuit has a specific solution, which is the realized value of its variable. We call it the primary solution and denote it by S^*.

4.3 Modeling the Refresh Gadget

We first focus on a single instance of leakage and then extend the obtained results to the general case of random leakage. Let the adversary learn a chunk of the primary solution

[2] This property can be verified by collecting many input and output shares and obtaining parity relations among them by deploying chain of Create-M, Gaussian-Elim, and Extract-Linear algorithms, with $\mathbf{M}$ having only $2n$ columns that are input-output shares

S^* via leakage. By substituting it into $\mathtt{A}_n$, a new system of equations as (19), denoted $\mathtt{A}_n^r$, will emerge:

$$\mathtt{A}_n^r = \begin{cases} \mathtt{NL}^r_{\mathbb{SG}_1}(\Sigma^r_{\mathbb{SG}_1}, V^{0,r}, v) = \mathbf{b}_1, \\ \mathtt{L}^r_{\mathbb{SR}}(\Sigma^r_{\mathbb{SR}}, V^{0,r}, V^{1,r}, v) = \mathbf{b}_2, \\ \mathtt{NL}^r_{\mathbb{SG}_2}(\Sigma^r_{\mathbb{SG}_2}, V^{1,r}, v) = \mathbf{b}_3. \end{cases} \tag{19}$$

The superscript ^{r} specifies the remaining unknowns and the relations among them. $\mathbf{b}_1$ to $\mathbf{b}_3$ are known vectors that appear after the substitution. Based on the leaked shares of V^0 and V^1, we have the following identities:

$$\begin{cases} v = (\oplus V^{0,r}) \oplus b_4, \\ v = (\oplus V^{1,r}) \oplus b_5, \end{cases} \tag{20}$$

where $\oplus$ before a set means XOR of all its entries, and scalar values b_4 and b_5 are respective XOR of the leaked shares of V^0 and V^1. The first row of (20) is a parity relation that is common to $\mathtt{NL}^r_{\mathbb{SG}_1}$ and $\mathtt{L}^r_{\mathbb{SR}}$, and the second row is common to $\mathtt{NL}^r_{\mathbb{SG}_2}$ and $\mathtt{L}^r_{\mathbb{SR}}$.

4.3.1 Extracting Non-Informative Parity Relations

Our aim is to simplify $\mathtt{A}_n^r$ by decomposing parity relations in $\mathtt{L}^r_{\mathbb{SR}}$ into two groups of non-informative and informative parity relations.

Lemma 2. *Without changing the set of solutions of* $\mathtt{A}_n^r$, $\mathtt{L}^r_{\mathbb{SR}}$ *can be grouped into the following two linearly independent sets of equations:*

$$\begin{cases} \mathtt{L}_1(\Sigma^r_{\mathbb{SR}}) = \mathtt{L}_2(V^{0,r}, V^{1,r}, v) \oplus \mathbf{b}_{2,1}, \\ \mathtt{L}_3(V^{0,r}, V^{1,r}, v) = \mathbf{b}_{2,2}, \end{cases} \tag{21}$$

where $\mathbf{b}_{2,1}$ *and* $\mathbf{b}_{2,2}$ *are constant vectors, and equations in* $\mathtt{L}_1$ *are linearly independent of each other.*

Proof. A coefficients matrix $\mathbf{P}$ can describe $\mathtt{L}^r_{\mathbb{SR}}(\Sigma^r_{\mathbb{SR}}, V^{0,r}, V^{1,r}, v) = \mathbf{b}_2$ as

$$\mathbf{P} \cdot [\Sigma^r_{\mathbb{SR}}, V^{0,r}, V^{1,r}, v)]^\top = \mathbf{b}_2. \tag{22}$$

Using $\mathbf{P}$, we create an *augmented matrix* $\mathbf{P'}$ by appending $\mathbf{b}_2$ to it as $\mathbf{P'} = [\mathbf{P}|\mathbf{b}_2]$. Let the row-echelon form of $\mathbf{P'}$ be $\mathbf{G} = \mathsf{Gaussian\text{-}Elim}(\mathbf{P'})$. From $\mathbf{G}$, we can derive $\mathtt{L}_1$, $\mathtt{L}_2$, and $\mathtt{L}_3$ as defined in the lemma.

Inside $\mathbf{G}$, because it is a row-echelon matrix, if the ith row is all-zero, then all the rows beneath i are all-zero. Let I be the first row of $\mathbf{G}$ that is all-zero. In any row $i < I$, there exists a column J such that $\mathbf{G}(i, J) = 1$, and for every $j < J$, $\mathbf{G}(i, j) = 0$. Column J is called a *pivot* variable. Let function $\mathsf{Pivots}(.)$, for each row $i < I$, give its corresponding column J. $\mathsf{Pivots}(.)$ is a monotonically increasing function. We define $\mathsf{Pivots}^{-1}(j)$ as the biggest number r such that $\mathsf{Pivots}(r) \leq j$.

Inside $\mathbf{P}$, define the last column containing elements of $\Sigma^r_{\mathbb{SR}}$ with $k = |\Sigma^r_{\mathbb{SR}}|$. Note that $\Sigma^r_{\mathbb{SR}}$ may contain an exact copy of V^0 and V^1. However, this will not alter our procedure or results.

We split $\mathbf{G}$'s columns into three parts as $\mathbf{G} = [\mathbf{G}_1|\mathbf{G}_2|\mathbf{G}_3]$. Where, $\mathbf{G}_1 = \mathbf{G}(:, 1:k)$, $\mathbf{G}_2 = \mathbf{G}(:, k+1 : \text{end}-1)$, and $\mathbf{G}_3 = \mathbf{G}(:, \text{end})$. We also split the rows of $\mathbf{G}$ into the following two disjoint sets.

- Rows $i = 1$ to $T = \min[I-1, \mathsf{Pivots}^{-1}(k)]$. The parity equations corresponding to these rows have a pivot variable from $\Sigma^r_{\mathbb{SR}}$. That is because, at least, $\mathbf{G}(i, \mathsf{Pivots}(i)) = 1$, and since $\mathsf{Pivots}(.)$ is monotonically increasing, we have:

$$\mathsf{Pivots}(i) \leq \mathsf{Pivots}(T) = \min[\mathsf{Pivots}(I-1), k] \leq k.$$

For these parity equations, we can write:

$$\mathbf{G}_1(i,:) \cdot [\Sigma^r_{\mathbb{SR}}]^\top \oplus \mathbf{G}_2(i,:) \cdot [V^{0,r}, V^{1,r}, v]^\top = \mathbf{G}_3(i). \tag{23}$$

Based on (23), we label the equations defined by $\mathbf{G}_1(1:T,:)$ as $\mathtt{L}_1$, the equations defined by $\mathbf{G}_2(1:T,:)$ as $\mathtt{L}_2$, and the constants at the column vector $\mathbf{G}_3(1:T)$ as $\mathbf{b}_{2,1}$. Each row of $\mathbf{G}_1(1:T,:)$ has a pivot variables. Therefore, these rows (that are also denoted by $\mathtt{L}_1$) are linearly independent.

- Rows $i = T+1$ to $I-1$. These rows exist only if $\mathsf{Pivots}^{-1}(k) < I-1$. For these rows, we have $\mathsf{Pivots}(i) > k$. Consequently, $\mathbf{G}_1(i,:) = \mathbf{0}$. Hence, no variable from $\Sigma^r_{\mathbb{SR}}$ will exist in the equations defined by these rows, for rows $T < i < I$, we can write:

$$\mathbf{G}_2(i,:) \times [V^{0,r}, V^{1,r}, v]^\top = \mathbf{G}_3(i). \tag{24}$$

Based on (24), we label the equations defined by $\mathbf{G}_2(T+1:I-1,:)$ as $\mathtt{L}_3$, and the constants at the column vector $\mathbf{G}_3(T+1:I-1)$ as $\mathbf{b}_{2,2}$.

Finally, the equations in $\mathbf{G}$ will be of the following structure, and this completes proof of the lemma.

$$\begin{cases} \mathtt{L}_1(\Sigma^r_{\mathbb{SR}}) = \mathtt{L}_2(V^{0,r}, V^{1,r}, v) \oplus \mathbf{b}_{2,1}, \\ \mathtt{L}_3(V^{0,r}, V^{1,r}, v) = \mathbf{b}_{2,2}. \end{cases} \tag{25}$$

□

Example 4. As a toy example, to illustrate the proof of the lemma, consider the following instance of matrix $\mathbf{G}$:

$$\mathbf{G} = \left[\begin{array}{ccccc:cccc:c} 1 & 0 & 0 & 0 & \alpha_0 & 0 & \beta_0 & 0 & \gamma_0 & \delta_0 \\ 0 & 1 & 0 & 0 & \alpha_1 & 0 & \beta_1 & 0 & \gamma_1 & \delta_1 \\ 0 & 0 & 0 & 1 & \alpha_2 & 0 & \beta_2 & 0 & \gamma_2 & \delta_2 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 1 & \beta_3 & 0 & \gamma_3 & \delta_3 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{26}$$

In this example, we have set $k = 5$, so the first 5 columns correspond to the variables of Σ^r. We also consider the next four columns as $[V^{0,r}, V^{1,r}, v]$. The last column is the augmentation with the $\mathbf{b}_2$ vector, see (22). Columns 1, 2, 4, and 6 are pivots. Column 3 and 8 are part of leakage, and column 5, 7, and 9 are free. The respective pivot columns of the rows $(1,2,3,4)$ are $(1,2,4,6)$. The 5th row of $\mathbf{G}$ is the first all-zero row. Hence, we have $I = 5$, and $T = \min[I-1, \mathsf{Pivots}^{-1}(k)] = \min[4,3] = 3$. Therefore, the first 3 rows identify $\mathtt{L}_1$, $\mathtt{L}_2$, and $\mathbf{b}_{2,1}$. The 4th row is the only relation in $\mathtt{L}_3$.

Next, we show that the subsystem given in the first row of (21) will not change the probability distribution of $\widetilde{v}$ (the adversary's output). Therefore, we label them as non-informative parity relations. Lemma 3 gives the necessary technical conditions.

Lemma 3. *Consider the following system of equations.*

$$\begin{cases} \mathtt{L}_4(\Sigma_1) = \mathtt{L}_5(v, \Sigma_2), \\ \mathtt{NL}(v, \Sigma_2, \Sigma_3) = \mathbf{0}. \end{cases} \tag{27}$$

If Σ_1, Σ_2, and Σ_3 are disjoint, and the equations in $\mathtt{L}_4$ are linearly independent, then the probability distribution of $\widetilde{v}$ will only depend on $\mathtt{NL}(v, \Sigma_2, \Sigma_3) = \mathbf{0}$. Consequently, as much as the estimation of v is concerned, $\mathtt{L}_4(\Sigma_1) = \mathtt{L}_5(v, \Sigma_2)$ can be ignored from (27).

Proof. A system of equations in a finite field has a limited number of solutions. Assume that S^1 to S^N are all of the solutions of the system defined by $\mathtt{NL}(v, \Sigma_2, \Sigma_3) = \mathbf{0}$, where S^i is a vector of values as $S^i = (v^i, \Sigma_2^i, \Sigma_3^i)$.

Note that it suffices to demonstrate that, corresponding to each S^i, there is a fixed set of unique values for Σ_1 such that the system $\mathtt{L}_4(\Sigma_1) = \mathtt{L}_5(v, \Sigma_2)$ is satisfied. See relation (9), and note that multiplying a constant value with both numerator and denominator of a fraction will cancel out.

For each S^i, $\mathtt{L}_5(v, \Sigma_2)$ is a known and fixed vector, which we denote with D^i. Now, we aim to compute the number of solutions of $\mathtt{L}_4(\Sigma_1) = D^i$. Since the equations in $\mathtt{L}_4$ are independent of each other, from linear algebra, we know that this system has at least one solution. We label this solution as $[\Sigma_1]_p$. The structure of the complete set of the solutions of $\mathtt{L}_4(\Sigma_1) = D^i$ is given in the following:

$$\Sigma_1 = [\Sigma_1]_p \oplus [\Sigma_1]_f, \tag{28}$$

where $[\Sigma_1]_f$ is any vector satisfying the homogeneous system defined by

$$\mathtt{L}_4(\Sigma_1) = \mathbf{0}. \tag{29}$$

The cardinality of the solutions in (28) is independent of the value of D^i. Hence, we conclude that the frequency of solutions for v is solely controlled by the subsystem defined by $\mathtt{NL}(v, \Sigma_2, \Sigma_3) = \mathbf{0}$. □

Example 5. To illustrate how the decomposition of parity relations of $\mathbb{SR}$ into two groups of informative and non-informative works, we provide a simple example of a refresh gadget for $n = 3$ as follows:

$$(v_1^1, v_2^1, v_3^1) \leftarrow (v_1^0, v_2^0, v_3^0) \oplus (r_1, r_2, r_1 \oplus r_2). \tag{30}$$

Assume the leakage has only revealed the value of r_1 to the adversary as $r_1 = b$. We can arrange the parity equations describing the given refresh instance as follows:

$$\begin{cases} v_1^0 \oplus v_2^0 \oplus v_3^0 \oplus v = 0, \\ v_1^0 \oplus v_1^1 = b, \\ r_2 \oplus v_2^0 \oplus v_2^1 = 0, \\ r_2 \oplus v_3^0 \oplus v_3^1 = b. \end{cases} \tag{31}$$

Using the approach of (the proof of) Lemma 2, this parity system can be reordered to yield:

$$\begin{cases} r_2 \oplus v_2^0 \oplus v_2^1 = b, \\ v_1^0 \oplus v_2^1 \oplus v_3^1 \oplus v = b, \\ v_2^0 \oplus v_3^0 \oplus v_2^1 \oplus v_3^1 = b, \\ v_1^1 \oplus v_2^1 \oplus v_3^1 \oplus v = 0. \end{cases} \tag{32}$$

The first equation, in the format of the first row of (27), is non-informative (per Lemma 3). Intuitively, this equation increases the number of unknowns by one, and this unknown (r_2) is not related to the rest of the system. Note that if such dummy equations were helpful, the adversary could add many of them, each introducing a new unknown, in the hope of increasing the chance of gaining more knowledge about v. However, we know that these artificial parities are not helpful.

4.3.2 Processing the Informative Parity Relations

By application of Lemmas 2 and 3, we can decompose $\mathtt{A}_n^r$ as follows:

$$\mathtt{A}_n^r = \begin{cases} \mathtt{NL}^r_{\mathbb{SG}_1}(\Sigma^r_{\mathbb{SG}_1}, V^{0,r}, v) = \mathbf{b}_1, \\ \mathtt{L}^r_{\mathbb{SR},3}(V^{0,r}, V^{1,r}, v) = \mathbf{b}_{2,2}, \\ \mathtt{NL}^r_{\mathbb{SG}_2}(\Sigma^r_{\mathbb{SG}_2}, V^{1,r}, v) = \mathbf{b}_3. \end{cases} \tag{33}$$

The linear equation set $\mathtt{L}^r_{\mathbb{SR},3}$, which only depends on the boundary shares and v, is obtained via the process in Lemma 2, and they are the only informative parities in $\mathtt{L}^r_{\mathbb{SR}}$.

From Lemma 1, we know that a linear system as $\mathtt{L}^r_{\mathbb{SR},3}(V^{0,r}, V^{1,r}, v) = \mathbf{b}_{2,2}$ either recovers v or gives nothing about it. Let $e_{\mathbb{SR}}$ be the event that $\mathtt{L}^r_{\mathbb{SR},3}$ alone determines the secret v. If $e_{\mathbb{SR}}$ occurs, v is recovered. Otherwise, we proceed with narrowing down the equations in $\mathtt{L}^r_{\mathbb{SR},3}$: for each equation containing v, we add either of the identities in (20) and, consequently, remove v from the resulting equation. At this point, the trivial relation

$$(\oplus V^{0,r}) \oplus b_4 = (\oplus V^{1,r}) \oplus b_5 \tag{34}$$

will always exist in this system. Based on the maximally independence requirement, there will be no other *omnipresent equation*[3] in this system.

Relation (34) is a dependent equation: it is XOR of the first line of (20), present in $\mathtt{NL}^r_{\mathbb{SG}_1}$, and the second line of (20), present in $\mathtt{NL}^r_{\mathbb{SG}_2}$. Therefore, we ignore (34) and denote the final system with $\mathtt{L}^r_{\mathbb{SR},4}(V^{0,r}, V^{1,r}) = \mathbf{b}_{2,3}$.[4] By substituting this into (33), we get

$$\mathtt{A}_n^r = \begin{cases} \mathtt{NL}^r_{\mathbb{SG}_1}(\Sigma^r_{\mathbb{SG}_1}, V^{0,r}, v) = \mathbf{b}_1, \\ \mathtt{L}^r_{\mathbb{SR},4}(V^{0,r}, V^{1,r}) = \mathbf{b}_{2,3}, \\ \mathtt{NL}^r_{\mathbb{SG}_2}(\Sigma^r_{\mathbb{SG}_2}, V^{1,r}, v) = \mathbf{b}_3. \end{cases} \tag{35}$$

In this phase, the subsystem defined by $\mathtt{L}^r_{\mathbb{SR},4}$ is without v. Our next goal is to separate $\mathtt{A}_n^r$ into systems governed by $\mathtt{NL}^r_{\mathbb{SG}_1}$ and $\mathtt{NL}^r_{\mathbb{SG}_2}$.

Example 6. Building on Example 5, we illustrate how the three informative equations in (32) can be narrowed down to derive $\mathtt{L}^r_{\mathbb{SR},4}$. The subsystem consisting of the three last equations of (32) corresponds to $\mathtt{L}^r_{\mathbb{SR},3}$. We aim to derive $\mathtt{L}^r_{\mathbb{SR},4}$ in this example. Here, the boundary parity relations are:

$$\begin{cases} v = v_1^0 \oplus v_2^0 \oplus v_3^0, \\ v = v_1^1 \oplus v_2^1 \oplus v_3^1. \end{cases} \tag{36}$$

Based on the discussion around (34), these two boundary relations are already present outside of $\mathbb{SR}$ (in $\mathbb{SG}_1$ and $\mathbb{SG}_2$, respectively). With the help of boundary parities, we eliminate v from $\mathtt{L}^r_{\mathbb{SR},3}$ and obtain:

$$\begin{cases} v_1^0 \oplus v_1^1 = b, \\ v_2^0 \oplus v_3^0 \oplus v_2^1 \oplus v_3^1 = b. \end{cases} \tag{37}$$

Now, any parity relation in $\mathtt{L}^r_{\mathbb{SR},3}$ that is linearly dependent in the presence of these boundary relations can be eliminated. For instance, if we select the parity relation $v_1^0 \oplus v_1^1 = b$, the other one will be dependent and can be ignored. Consequently, $\mathtt{L}^r_{\mathbb{SR},4}$ will be $v_1^0 \oplus v_1^1 = b$. Note that we could have selected the other parity, namely $v_2^0 \oplus v_3^0 \oplus v_2^1 \oplus v_3^1 = b$, as $\mathtt{L}^r_{\mathbb{SR},4}$. The first choice has only two unknowns. Hence, it is *sparser* than the second one, which contains four unknowns. However, both can be considered as $\mathtt{L}^r_{\mathbb{SR},4}$.

[3] Omnipresent in the sense that it exists irrespective of the realization of leakage.

[4] Typically at this point, with small p, there remains no equations in $\mathtt{L}^r_{\mathbb{SR},4}$.

4.3.3 Disclosing Unknowns of Residual Parity Equations

To facilitate the separation of $\mathtt{A}_n^r$ into its constituent subsystems, we allow the adversary to learn the *remaining unknowns* of $\mathtt{L}_{\mathbb{SR},4}^r$. This additional leakage increases the success probability of the adversary and consequently loosens the tightness of the final security bound. Nevertheless, this intervention is crucial to our proposed security reduction.

We denote the unknowns of $\mathtt{L}_{\mathbb{SR},4}^r$ that are members of V^0 and V^1 with $V^{0,\dagger}$ and $V^{1,\dagger}$, respectively. The disclosure of this extra leakage enables us to split $\mathtt{A}_n^r$ into two (almost) disjoint systems $\mathtt{A}_n^1$ and $\mathtt{A}_n^2$, with the only common variable being v:

$$\mathtt{A}_n^1 = \begin{cases} \mathtt{NL}_{\mathbb{SG}_1}^r(\Sigma_{\mathbb{SG}_1}^r, V^{0,r}, v) = \mathbf{b}_1, \\ V^{0,\dagger} = \mathbf{b}_6, \end{cases} \tag{38}$$

$$\mathtt{A}_n^2 = \begin{cases} \mathtt{NL}_{\mathbb{SG}_2}^r(\Sigma_{\mathbb{SG}_2}^r, V^{1,r}, v) = \mathbf{b}_3, \\ V^{1,\dagger} = \mathbf{b}_7. \end{cases} \tag{39}$$

In the context of Example 6, with $\mathtt{L}_{\mathbb{SR},4}^r$ being $v_1^0 \oplus v_1^1 = b$, $V^{0,\dagger}$ and $V^{0,\dagger}$ are v_1^0 and v_1^1, and their values are given to the adversary in accordance with the primary solution (See Definition 7). Choosing a sparser representation for $\mathtt{L}_{\mathbb{SR},4}^r$ helps to derive a tighter security bound. However, finding a sparse representation is not a trivial problem. We leave optimizations of the results based on this direction for future work.

4.3.4 Counting the Number of Informative Parity Relations

The discussion up to here was based on a single instance of leakage. To proceed the reduction, we zoom out and make use of the probabilistic nature of leakage. In the random leakage setting, $V^{0,\dagger}$ and $V^{1,\dagger}$ are not fixed. We utilize the symmetry (Definition 5), and model the extra leakage provided through $V^{0,\dagger}$ and $V^{1,\dagger}$ as an *equivalent leakage rate* p' on the interface vectors V^0 and V^1.

For each interface, we define $\mathsf{T}_{\mathbb{SR}}(n,p)$ as:

$$\mathsf{T}_{\mathbb{SR}}(n,p) = \mathbb{E}[|V^{i,\dagger}|]. \tag{40}$$

Here, the expectation is over instances of leakage and i is in $\{0,1\}$. For the particular case of $\mathbb{SR}$-SNI, due to the present symmetry, we have $\mathbb{E}[|V^{0,\dagger}|] = \mathbb{E}[|V^{1,\dagger}|]$.

For a target $\mathbb{SR}$ at a desired pair (n,p), we estimate $\mathsf{T}_{\mathbb{SR}}(n,p)$ using a Monte Carlo approach with N trials, where in the kth trial, we sample a random leakage and compute $h(k) = |V^{i,\dagger}|$. For a sufficiently large N, the sample average, $\frac{1}{N}\sum_{k=1}^{N} h(k)$, approaches $\mathsf{T}_{\mathbb{SR}}(n,p)$. We give a detailed algorithmic approach in the following.

Algorithm for Estimating $\mathsf{T}_{\mathbb{SR}}(n,p)$. Algorithm 8 outlines the procedure described in this section to estimate $\mathsf{T}_{\mathbb{SR}}(n,p)$ for a given $\mathbb{SR}$ gadget.

The function Count-Unknowns($\mathbf{G}_2$) performs three tasks: first, it adds either $v = \oplus V^{0,r} \oplus b_4$ or $v = \oplus V^{1,r} \oplus b_5$ (as in (20)) to any row that has a non-zero coefficient for v. Then, it removes $\oplus V^{0,r} \oplus b_4 = \oplus V^{1,r} \oplus b_5$ from the set of rows of $\mathbf{G}_2$ (see discussion around (34)). Finally, it counts the number of shares of V^0 or V^1 that have at least one non-zero coefficient in $\mathbf{G}_2$, as in (40).

Estimation of $\mathsf{T}_{\mathbb{SR}\text{-SNI}}(n,p)$. For later reference, we have estimated $\mathsf{T}_{\mathbb{SR}\text{-SNI}}$ using Algorithm 8 at various pairs of (n,p). The results are depicted in Figure 5, and they fit in the following bound:

$$\mathsf{T}_{\mathbb{SR}\text{-SNI}}(n,p) \leq \frac{1}{3}np, \quad \text{valid for } (n \geq 3, p \leq 0.1). \tag{41}$$

Algorithm 8 Approximate $\mathsf{T}_{\mathbb{SR}}$

Input The family $\mathbb{SR}$
Output $\mathsf{T}_{\mathbb{SR}}(n,p)$

for $n=2$ **to** $n_{\max}$ **do**
  $\mathbf{M}_n = \mathsf{Create\text{-}M}(\mathbb{SR},n)$ ▷ Order of columns: $[\Sigma_{\mathbb{SR}} \mid V^0 \mid V^1 \mid v]$, $m(n)+1$ columns.
  $\mathbf{D}_n = \mathsf{Gaussian\text{-}Elim}(\mathbf{M}_n)$
  $\mathbf{P}_n = \mathsf{Extract\text{-}Linear}(\mathbf{D}_n)$ ▷ $\mathtt{L}_{\mathbb{SR}}(\Sigma_{\mathbb{SR}}, V^0, V^1, v)$ as in (17)
  for $p = p_{\min}$ **to** $p_{\max}$ **do**
    $N_{\mathsf{recovered}} = 0, h = 0$
    for $i = 0$ **to** N **do**
      $\mathbf{P}_n^r \leftarrow \mathsf{Sampl}(\mathbf{P}_n, p)$ ▷ Only internal variables, i.e., member of $\mathbb{SR}$, leak. $\mathtt{L}^r_{\mathbb{SR}}$ as in (19)
      $\mathbf{G} = \mathsf{Gaussian\text{-}Elim}(\mathbf{P}_n^r)$ ▷ Approach of Lemma 2
      if $\mathbf{G}(\mathsf{end}, 1 : m(n)) = 0$ **then** ▷ $e_{\mathbb{SR}}$ event. v is in the last column of $\mathbf{M}$
        $N_{\mathsf{recovered}} = N_{\mathsf{recovered}} + 1$
        Continue
        $\mathbf{G}_2 = \mathsf{Extract}(\mathbf{G})$ ▷ In accordance with (24)
        $h = h + \mathsf{Count\text{-}Unknowns}(\mathbf{G}_2)$
    $\mathsf{T}_{\mathbb{SR}}(n,p) = \frac{h}{N - N_{\mathsf{recovered}}}$ ▷ Approximation with empirical mean
return $\mathsf{T}_{\mathbb{SR}}$

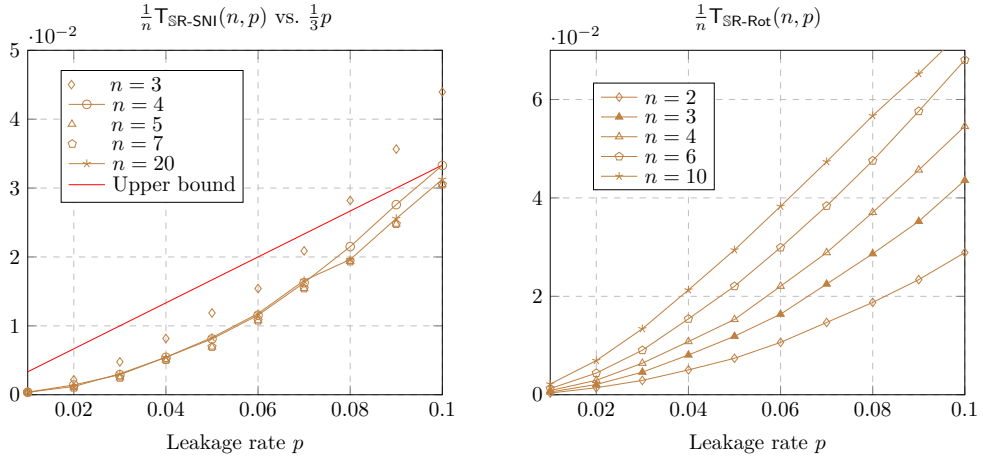


Figure 5: (Left) Estimation results for $\frac{1}{n}\mathsf{T}_{\mathbb{SR}\text{-}\mathsf{SNI}}(n,p)$. The given upper bound holds with acceptable accuracy. Note that $\frac{1}{n}\mathsf{T}_{\mathbb{SR}\text{-}\mathsf{SNI}}(n,p)$ decreases with n. (Right) The estimated value for $\frac{1}{n}\mathsf{T}_{\mathbb{SR}\text{-}\mathsf{Rot}}(n,p)$ increases with n. Hence, we cannot device a similar upper bound.

With $n=3$, the given bound is only valid for $p < 0.8$. One can find a better matching bound with $n \geq 4$. However, for our illustration purpose, the given bound suffices.

Remark 8. $\mathbb{SR}$-Rot satisfies Definitions 5 and 6. However, $\mathsf{T}_{\mathbb{SR}\text{-}\mathsf{Rot}}(n,p)/n$ grows with n for any tested region of p. See Figure 5-(Right).

4.3.5 Devising a Union Bound

In the following lemma, we use $\mathsf{T}_{\mathbb{SR}}(n,p)$ to compute an equivalent leakage rate p'.

Lemma 4. *In the RPM setting, at pair (n,p), the distribution of $\widetilde{v}$ conditioned on*

$$\begin{cases} \mathtt{NL}_{\mathbb{SG}}(\Sigma_{\mathbb{SG}}, V, v) = \mathbf{0}, \\ V^\dagger = \mathbf{b}, \end{cases} \tag{42}$$

with $\mathsf{T} = \mathbb{E}[|V^\dagger|]$ is statistically equivalent to the distribution of $\widetilde{v}$ conditioned on

$$\mathtt{NL}_{\mathbb{SG}}(\Sigma_{\mathbb{SG}}, V, v) = \mathbf{0}, \tag{43}$$

where the shares in V leak at rate:

$$p' = p + \frac{\mathsf{T}}{n} - p\frac{\mathsf{T}}{n} \approx p + \frac{\mathsf{T}}{n}. \tag{44}$$

Proof. The set of equations defined by $\mathtt{NL}_{\mathbb{SG}}$ is deterministic, with only the collection of known variables being probabilistic. Our goal is to demonstrate that the distribution of the known elements in the input V is identical in both systems.

In the first system, members of V are known to the adversary from two sources: the usual p leakage from the $\mathbb{SG}$ side and $V^{\dagger}$ resulting from the $\mathbb{SR}$ side. As the parities corresponding to the $\mathbb{SR}$ gadget are symmetric for the members of V, each member of V is in $V^{\dagger}$ independently with a probability of $\frac{\mathsf{T}}{n}$. Hence, the combined probability of leakage for each member of V is:

$$p' = p + \frac{\mathsf{T}}{n} - p\frac{\mathsf{T}}{n}. \tag{45}$$

In the second system, members of V are directly leaking with a probability of p'. Therefore, the inputs to both systems are statistically equivalent, and they should produce statistically equivalent results. □

Remark 9. If $\mathbb{SR}$ is leak-free, due to the maximal independence requirements on the input-output shares V^0 and V^1, there will be no other relation than trivial $\oplus_{i=1}^{n} v_i^0 = \oplus_{i=1}^{n} v_i^1$. Therefore, sets $V^{0,\dagger}$ and $V^{1,\dagger}$ will be empty, and consequently, $\mathsf{T}(n,p)$ will be zero, which results in $p' = p$.

To proceed our approach to prove (16), assume that there exists an EEC for (non-)linear $\mathbb{SG}_1$ and $\mathbb{SG}_2$ gadgets. Developing an EEC for a multiplication gadget will be treated in Section 5.
The following lemma establishes how to merge information from multiple EECs.

Lemma 5. *Learning v through a set of independent EECs with parameters $\{\mathsf{E}_1, \ldots, \mathsf{E}_k\}$ is equivalent to learning it via an EEC with parameter $\mathsf{E}' \leq \sum_{i=1}^{k} \mathsf{E}_i$.*

Proof. The lemma is obtained by applying the union bound on the probabilities. □

We are now able to compute an EEC for the chain $\mathbb{SG}_1 \to \mathbb{SR} \to \mathbb{SG}_2$:

$$\begin{aligned}
\mathsf{E}_{\mathbb{SG}_1 \to \mathbb{SR} \to \mathbb{SG}_2}(n,p) &= \Pr(\widetilde{v} = v^*) \\
&= \Pr(\widetilde{v} = v^* \mid e_{\mathbb{SR}}) \Pr(e_{\mathbb{SR}}) + \Pr(\widetilde{v} = v^* \mid \overline{e_{\mathbb{SR}}}) \Pr(\overline{e_{\mathbb{SR}}}) \\
&\leq \Pr(e_{\mathbb{SR}}) + \Pr(\widetilde{v} = v^* \mid \overline{e_{\mathbb{SR}}}) \\
&= \mathsf{E}_{\mathbb{SR}}(n,p) + \Pr(\widetilde{v} = v^* \mid \overline{e_{\mathbb{SR}}}) \\
&\leq \mathsf{E}_{\mathbb{SR}}(n,p) + \mathsf{E}_{\mathbb{SG}_1}(n,p') + \mathsf{E}_{\mathbb{SG}_2}(n,p'),
\end{aligned} \tag{46}$$

where p' is as given in (44), and p should be bounded such that EECs for $\mathbb{SG}_1$ and $\mathbb{SG}_2$ remain in their valid region.

4.4 Discussion on the Methodology and Results

To decompose the security of combinations in the RPM framework, we introduced a novel technique based on categorizing the parity relations of the deployed $\mathbb{SR}$ gadget into non-informative and informative groups. Our main technical contribution is Theorem 1, which is rigorously stated and proved for gadgets that satisfy the symmetry (Definition 5) and maximal independence (Definition 6) properties. However, the application of this approach using specific $\mathbb{SR}$ gadgets requires computing parameters such as $\mathsf{T}_{\mathbb{SR}}$ and $\mathsf{E}_{\mathbb{SR}}$,

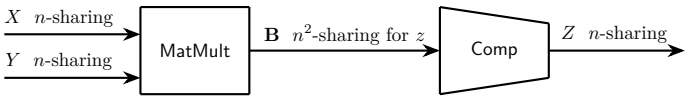
which directly depend on the details of the interconnection of the intermediates in the gadget. Since the derivation of $\mathsf{T}_{\mathbb{S}\mathsf{R}}(n,p)$ and $\mathsf{E}_{\mathbb{S}\mathsf{R}}(n,p)$ with an analytical approach seems complicated, we have employed a Monte Carlo method to estimate these parameters in a practical range of (n,p) tuples.

Relations With Threshold Probing Model. Among the three $\mathbb{S}\mathsf{R}$ gadgets considered, only $\mathbb{S}\mathsf{R}$-SNI satisfies the requirement of our proposed security reduction. This gadget, also useful in the Threshold Probing Model (TPM), is proven to be valuable for secure compositions [BBD+16]. More specifically, $\mathbb{S}\mathsf{R}$-SNI satisfies the t-SNI security definition [BBD+16], which informally implies that any $n-1$ output shares are independent of input shares (if no intermediate leaks). t-SNI security also implies that the independence of input-output shares can be maintained if a limited number of internal intermediates are disclosed to the adversary. These intuitive interpretations of t-SNI bear some similarity to the maximal independence in our approach. However, we leave the development of more technical connections between the definitions for future work.

5 A Multiplication Gadget With RPM Security

A multiplication gadget, denoted as $\mathbb{S}$AND, takes two n-sharings X and Y and computes an n-sharing Z in a way that preserves the security of the three native values while ensuring that $z = xy$. This section considers the $\mathbb{S}$AND candidate introduced by Battistello et al. in [BCPZ16], labeled as $\mathbb{S}$AND-Rec. With certain modifications, we demonstrate the conjectured RPM security of this gadget. Our primary technique is inclusion or exclusion of specific leakage information, which enables linearization of the describing parity relations.

Structure of $\mathbb{S}$AND-Rec. $\mathbb{S}$AND-Rec consists of two algorithms: MatMult and Comp. MatMult takes two n-sharings X and Y as input and generates an n^2-sharing for $z = xy$. These n^2 values are organized into an $n \times n$ matrix, denoted as $\mathbf{B}$. Comp then operates on $\mathbf{B}$ using fresh randomness to compress it into an n-sharing, whose secret is z.



MatMult Algorithm. MatMult, outlined in Algorithm 9, is a recursive process. At the initial invocation, it takes an n_X-sharing X and an n_Y-sharing Y with $n_X = n_Y = n$. In the subsequent calls, if n_X and n_Y are both 1, the algorithm sets $\mathbf{B}$ to $x_1 y_1$ and the computation terminates. Otherwise, the input vectors X and Y are divided into left and right subvectors. We denote the left subvectors as X_L and Y_L and the right subvectors as X_R and Y_R as defined in lines 4 and 5 of the algorithm.

With $n_X = 1$ and $n_Y > 1$ (resp. $n_Y = 1$ and $n_X > 1$), X_L (resp. Y_L) will be an empty vector. Empty vectors are represented by a $\varnothing$ symbol. For length-one inputs, the refreshing gadget is the identity function. When X_L is empty, both $\mathbf{B}_{LL}$ and $\mathbf{B}_{LR}$ are empty. Similarly, if Y_L is empty, then $\mathbf{B}_{LL}$ and $\mathbf{B}_{RL}$ will be empty as well.

Algorithm 9 MatMult

Input X, Y
Output An n^2-sharing for $z = xy$ saved in $\mathbf{B}$

1: **if** $n_X = 1$ and $n_Y = 1$ **then**
2: $\quad \mathbf{B} = x_1 y_1$
3: **else**
4: $\quad X_L = (x_1, \ldots, x_{\lfloor n_X/2 \rfloor})$ $X_R = (x_{\lfloor n_X/2 \rfloor + 1}, \ldots, x_{n_X})$
5: $\quad Y_L = (y_1, \ldots, y_{\lfloor n_Y/2 \rfloor})$ $Y_R = (y_{\lfloor n_Y/2 \rfloor + 1}, \ldots, y_{n_Y})$
6: $\quad$ **if** $X_L \neq \varnothing$ & $Y_L \neq \varnothing$ **then**
7: $\quad\quad \mathbf{B}_{LL} = \mathsf{MatMult}(\mathbb{SR}(X_L), \mathbb{SR}(Y_L))$
8: $\quad$ **if** $X_L \neq \varnothing$ **then**
9: $\quad\quad \mathbf{B}_{LR} = \mathsf{MatMult}(\mathbb{SR}(X_L), \mathbb{SR}(Y_R))$
10: $\quad$ **if** $Y_L \neq \varnothing$ **then**
11: $\quad\quad \mathbf{B}_{RL} = \mathsf{MatMult}(\mathbb{SR}(X_R), \mathbb{SR}(Y_L))$
12: $\quad \mathbf{B}_{RR} = \mathsf{MatMult}(\mathbb{SR}(X_R), \mathbb{SR}(Y_R))$
13: $\quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_{LL} & \mathbf{B}_{LR} \\ \mathbf{B}_{RL} & \mathbf{B}_{RR} \end{bmatrix}$
14: **return** $\mathbf{B}$

Algorithm 10 Comp

Input The n^2-sharing in $\mathbf{B}_{[n \times n]}$
Output An n-sharing Z for z

1: **for** $i = 1$ **to** n **do**
2: $\quad$ **for** $j = i + 1$ **to** n **do**
3: $\quad\quad r_{i,j} \xleftarrow{\$} \mathbb{F}_{2^u}$
4: $\quad\quad r_{j,i} = (r_{i,j} \oplus \mathbf{B}_{i,j}) \oplus \mathbf{B}_{j,i}$
5: **for** $i = 1$ **to** n **do**
6: $\quad z_i = \mathbf{B}_{i,i}$
7: $\quad$ **for** $j = 1$ **to** n , $j \neq i$ **do**
8: $\quad\quad z_i = z_i \oplus r_{i,j}$
9: **return** $Z = (z_1, z_2, \ldots, z_n)$

Comp Algorithm. Comp, described in Algorithm 10 [BCPZ16], squeezes the n^2-sharing in $\mathbf{B}$ into an n-sharing in Z. Comp is linear, and its input and output are sharings for the same secret. It is symmetric, and, except for $\oplus\mathbf{B} = \oplus Z$, there is no relation between its input and output shares. These similarities with $\mathbb{SR}$ gadgets enable the application of techniques developed in Section 4 to estimate the RPM security of MatMult $\rightarrow$ Comp composition as:

$$\mathsf{E}_{\mathsf{MatMult} \rightarrow \mathsf{Comp}}(n, p) \leq \mathsf{E}_{\mathsf{MatMult}}(n, p') + \mathsf{E}_{\mathsf{Comp}}(n, p). \tag{47}$$

Here, p' exceeds p and depends on the structure of Comp.

5.1 RPM Security of $\mathbb{S}$AND-Rec

For MatMult, we exhibit an EEC, with parameter E^+, that serves as an upper bound on the adversary's post-leakage knowledge. To illustrate the tightness of this upper bound, we put forward a lower bound modeled via another EEC with parameter E^-. The crucial observation is that the output of the multiplication (Line 2 in Algorithm 9) is not involved in any further computation except storing for later processing in Comp. This observation is visually represented in Figure 6. Moreover, according to Algorithm 9, the two linear sets of operations before multiplications, which are processing X and Y, are independent and identical.

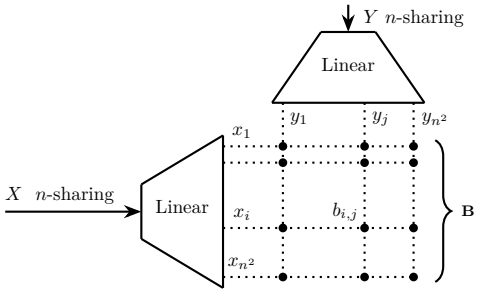


Figure 6: Interpretation of MatMult. The n^2 x_i (and y_i) values represent the output of the respective linear operations. Black-filled circles denote multiplication.

Devising E^-. Non-linear relations among the variables in $\Sigma_{\mathsf{MatMult}}$ stem from operands stored in $\mathbf{B}$. By excluding $\mathbf{B}$ from $\Sigma_{\mathsf{MatMult}}$, we ensure that all relations are linear. Therefore, if we assume that the adversary does not receive any leakage from $\mathbf{B}$, the parity description for the native x and y becomes linear. As a result, we can represent the adversary's post-leakage information about x (resp. y) via an EEC with parameter E^-.

Devising E^+. As depicted in Figure 6, for each entry $b_{i,j}$ in $\mathbf{B}$, there exist corresponding x_i and y_j in $\Sigma_{\mathsf{MatMult}}$ such that $b_{i,j} - x_i y_j = 0$. Other than this equation, $b_{i,j}$ does not participate in any other relation.

If $b_{i,j}$ is not leaked to the adversary, this parity relation does not help the adversary, and we can safely ignore it.[5]

If $b_{i,j}$ leaks, it creates a non-linear relation that cannot be ignored. In this scenario, we let the adversary know both x_i and y_j. This intervention increases the adversary's success probability in learning information about the natives and eliminates the non-linear parity equation. This trick allows us to attribute MatMult with a linear (and more informative) alternative parity description. The resulting linear system, as per Lemma 1, either reveals x (resp. y) or provides no information. Hence, we can model the post-leakage knowledge with an EEC, denoted by E^+.

Choosing an Appropriate Secret. As shown in Figure 6, computations on X and Y vectors are carried out separately until the final multiplications. Consequently, in both E^+ and E^-, the parity equations for X and Y form two separate, identical, and independent subsystems. This implies that the EEC for each of the native inputs will be independent and similar. Later for deriving the upper bound in (47), we assume that recovery of each native variable will disclose all the native variables.

Computing E^+ and E^-. In Figure 7, we present estimations for $\mathsf{E}^+(n,p)$ and $\mathsf{E}^-(n,p)$ at various orders. These results are based on the instantiation of $\mathbb{S}\mathsf{R}$ with $\mathbb{S}\mathsf{R}$-Simple.

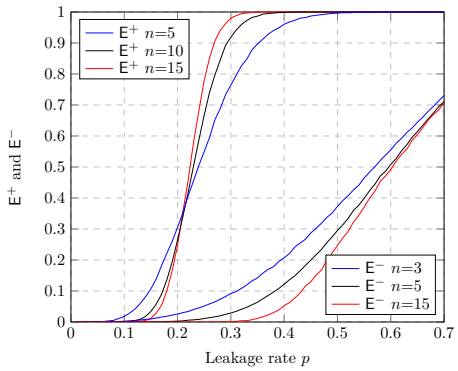


Figure 7: E^+ and E^- for MatMult at different orders.

Based on experiments for $(n \leq 30, p \leq 0.15)$ using Algorithm 5, we have the following approximations:

$$\mathsf{E}^+(n,p) \leq p^{0.3n}, \quad \text{and} \quad \mathsf{E}^-(n,p) \geq p^{0.8n}. \tag{48}$$

Remark 10. From the results in Figure 7, it is evident that for leakage rates $p > 0.22$, as the masking order increases, E^+ also increases. Consequently, MatMult with $\mathbb{S}\mathsf{R}$-Simple may be insecure at these rates.

[5] A similar proof to that of Lemma 3 can demonstrate that this parity relation does not affect the posterior distribution of the native values.

Remark 11. $\mathbb{S}$R-SNI is utilized as the $\mathbb{S}$R component of MatMult in [BCPZ16], resulting in an overall complexity of $\mathcal{O}(n^2 \log(n))$. However, in our numerical experiments, we observed that for any tested pair (n, p), using $\mathbb{S}$R-Simple as the instantiation for MatMult led to a lower success probability for the adversary compared to when MatMult was instantiated with $\mathbb{S}$R-SNI. Additionally, using $\mathbb{S}$R-Simple, complexity of the gadget drops to $\mathcal{O}(n^2)$.

Including Leakage of Comp. Both MatMult and Comp are symmetric (Definition 5). Comp is linear, and its input and output shares are maximally independent (Definition 6), and they represent sharings of the same secret. This allows us to apply a similar procedure used in Section 4 to bound $\mathsf{E}_{\mathbb{S}\text{AND-Rec}}(n, p)$. The main difference is that Comp's input vector consists of n^2 variables. Consequently, the computation of p' is altered as follows:

$$p' \approx p + \frac{\mathsf{T}_{\mathsf{Comp}}(n, p)}{n^2}. \tag{49}$$

Here, $\mathsf{T}_{\mathsf{Comp}}(n, p)$ represents the expected number of extra variables that will be revealed to the adversary (for the derivation of the upper bound (47)). In the notation of Section 4, we have:

$$\mathsf{T}_{\mathsf{Comp}}(n, p) = \mathbb{E}[|\mathbf{B}^{\dagger}|]. \tag{50}$$

Using Algorithm 8, for $n \leq 30$, we obtained the following estimations:

$$\begin{aligned} &\mathsf{T}_{\mathsf{Comp}}(n, p) \leq pn^2, \quad &&\text{valid for } (n \geq 4, p \leq 0.07), \\ &\mathsf{E}_{\mathsf{Comp}}(n, p) \leq p^{0.6n}, \quad &&\text{valid for } (n \geq 2, p \leq 0.15). \end{aligned} \tag{51}$$

By substituting the estimations in (51) and (48) into the $\mathsf{E}_{\mathbb{S}\text{AND-Rec}}(n, p)$ relation given in (47), for $(n \geq 4, p \leq 0.07)$, we have:

$$\mathsf{E}_{\mathbb{S}\text{AND-Rec}}(n, p) \leq \mathsf{E}_{\mathsf{MatMult}}(n, p') + \mathsf{E}_{\mathsf{Comp}}(n, p) \leq_{(a)} \left(p + \frac{pn^2}{n^2}\right)^{0.3n} + p^{0.6n} \approx (2p)^{0.3n}. \tag{52}$$

This bound demonstrates the RPM security of the proposed $\mathbb{S}$AND-Rec structure in the tested region of the (n, p) pairs. Note that the left side of (a) is analytically proved, but the right side is based on our probability estimation approach and might not hold outside the tested region.

6 RPM Secure Masking of S-box in AES

The $\mathbb{S}$AND-Rec and $\mathbb{S}$R-SNI gadgets enable a masked instantiation of the AES's S-box based on the work of Prouff and Rivain [RP10]. We denote this masked circuit as $\mathbb{S}$S-box, and in addition to demonstrating RPM security at $p \leq 0.03$ rates, we outline an EEC for it.

For an n-sharing V corresponding to a native v in a specified $\mathbb{F}_{2^8}$, $\mathbb{S}$S-box needs to compute an n-sharing of v^{-1}. Prouff and Rivain's approach for computing this inverse is based on the following relation:

$$v^{-1} = v^{254} = \left[(v^2 v)\ (v^2 v)^4\right]^{16}\ (v^2 v)^4\ v^2. \tag{53}$$

Since $(.)^{2^i}$ is an additive operation in $\mathbb{F}_{2^8}$, only four non-affine multiplications are required to compute (53). These multiplications are performed using $\mathbb{S}$AND gadgets. Thus, $\mathbb{S}$S-box operates as visualized in Figure 8. To prepare for RPM security reduction, we have included more refresh gadgets compared to [RP10].

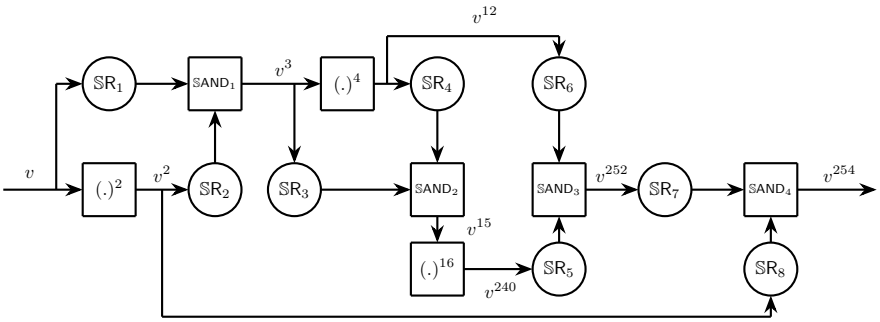


Figure 8: Structure of $\mathbb{S}$S-box. Powers of v represent the corresponding native values.

6.1 RPM Security of the Proposed $\mathbb{S}$S-box

In the structure of $\mathbb{S}$S-box, $\mathbb{S}$R gadgets enable us to apply the reduction method developed in Section 4 to translate the RPM security of the composition into the RPM security of its constituent gadgets. Furthermore, since we have already developed an EEC for the individual gadgets, by applying Lemma 5, we can derive an EEC for the entire $\mathbb{S}$S-box.

Unifying the Secret. Secrets of the gadgets are different powers of v. To unify the secrets, at the cost of increasing the upper bound, we assume that knowledge of v^j reveals v to the adversary. This assumption also allows us to disregard leakage from additive $(.)^{2^i}$ gadgets, as they do not introduce new intermediates and only modify the exponent of v for the next gadget.

The Case of Multiple $\mathbb{S}$R Gadgets. When feeding multiple, say k, $\mathbb{S}$R gadgets with the same n-sharing, we can obtain the corresponding p' as $p' \approx p + \frac{k\mathsf{T}_{\mathbb{S}\mathsf{R}}}{n}$.

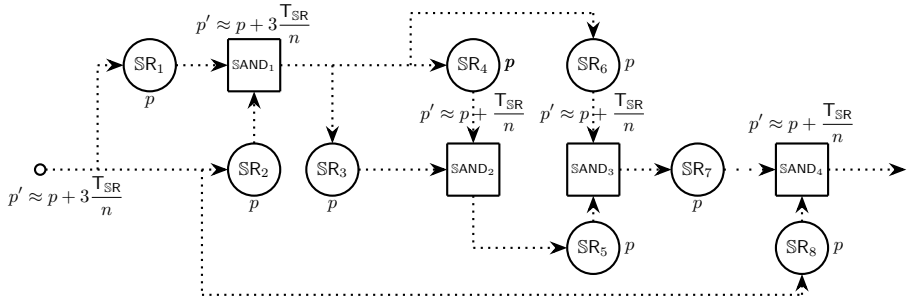


Figure 9: Reduction of RPM security of $\mathbb{S}$S-box to RPM security of its composing gadgets.

As illustrated in Figure 9 and with the application of the reduction method in Section 4 and Lemma 5, for $(n \geq 4, p \leq 0.03)$, we can use (15), (41), and (52) to obtain:

$$\begin{aligned}\mathsf{E}_{\mathbb{S}\mathsf{S}\text{-box}}(n,p) &\leq 8\mathsf{E}_{\mathbb{S}\mathsf{R}}(n,p) \\ &\quad + 3\mathsf{E}_{\mathbb{S}\mathsf{AND}}\left(n, p + \frac{\mathsf{T}_{\mathbb{S}\mathsf{R}}(n,p)}{n}\right) + \mathsf{E}_{\mathbb{S}\mathsf{AND}}\left(n, p + 3\frac{\mathsf{T}_{\mathbb{S}\mathsf{R}}(n,p)}{n}\right) \\ &\leq_{(a)} 8p^{0.6n} + 3\left(\frac{8}{3}p\right)^{0.3n} + (4p)^{0.3n} \approx 4(4p)^{0.3n}.\end{aligned} \tag{54}$$

Note that at $p \leq 0.03$, the derived p' and EECs are valid. Also, note that the left side of (a) is based on analytical approach. However, the right side of it is based on probability estimation.

7 RPM Secure Masking of AES

Having established an RPM secure $\mathbb{S}$S-box, we now aim to explore RPM security in a more extensive composition: a masked AES cipher. In this cipher, operations other than S-box are affine and, consequently, straightforward to mask (see [RP10] for details). This section reviews the structure of a masked AES, denoted as $\mathbb{S}$AES, and approximates the RPM security for a specific target: the first sub-key of the first round.

Overview of $\mathbb{S}$AES. AES [Nat01] is a block cipher supporting various key sizes. In this paper, we focus on the 128-bit key variant, consisting of 10 rounds. Each round updates the *state*, which is a 16-byte vector initially filled with the plaintext. The state is transformed using a 16-byte round key derived from the input key. Our goal is to approximate the adversary's post-leakage information on K^1, the first byte of the key for the first round. The operations of a $\mathbb{S}$AES in the initial two rounds are outlined in Figure 10.

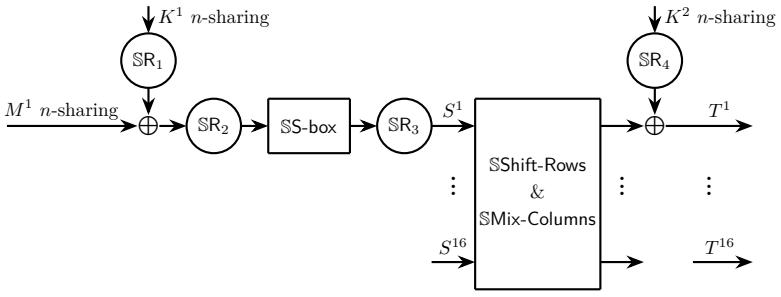


Figure 10: Targeting k^1 (secret of K^1) in $\mathbb{S}$AES. M^1 is an n-sharing for the first byte of plaintext, and K^2 is an n-sharing for the first sub-key of the second round. S^1 to S^{16} and T^1 to T^{16} are n-shared bytes of the state.

$\mathbb{S}$Shift-Rows and $\mathbb{S}$Mix-Columns are affine and operate on the n-shared counterpart of the state. The S-box, aside from computing the inverse, also applies a linear operation at its output, which is considered part of the large affine block in Figure 10. The refresh gadget in our experiments is $\mathbb{S}$R-SNI.

Remark 12. Since round keys only depend on the input key, they can be computed once. In a masked domain, this translates to saving n-sharings of the sub-keys and refreshing them before each invocation.

7.1 Estimating the RPM Security

To assess the adversary's post-leakage information about k^1, that is the secret of K^1, we employ the reduction framework detailed in Section 4, focusing on $\mathbb{S}R_3$. We calculate EEC for both the left and right components of this $\mathbb{S}$R and combine the results using Lemma 5. For the derivation of results in this section, we assume that K^2 and K^1 are independent of each other.

On the left side of $\mathbb{S}R_3$, secret of the gadgets are either $m^1 \oplus k^1$ or k^1. However, since m^1 (the secret of M^1) is known to the adversary, these two native values are equivalent.

Hence, using the reduction, we have:

$$\mathsf{E}_{\text{left}}(n,p) = \mathsf{E}_{\mathbb{S}\text{S-box}}\left(n, p + \frac{\mathsf{T}_{\mathbb{S}\mathsf{R}}}{n}\right) + 2\mathsf{E}_{\mathbb{S}\mathsf{R}}(n,p). \tag{55}$$

On the right side of $\mathbb{S}\mathsf{R}_3$, we have an $\mathbb{F}_{2^8}$-affine block, for which the approach developed in Section 3 is sufficient to sketch an EEC. To account for the leakage of the rounds (3 to 10), at the cost of slightly increasing the gap of the upper bound, we assume that shares of T^1 to T^{16} are known to the adversary. We also include secrets of the other 15 $\mathbb{S}$S-boxes as auxiliary intermediates inside the affine block and assume that each of these secrets is leaking with a probability $\mathsf{E}_{\text{left}}(n,p)$. More specifically, we make the following tweaks to Algorithm 5.

Estimating $\mathsf{E}_{\text{right}}$. In the Create-M sub-algorithm, the secrets s^i (secrets of $\mathbb{S}$S-boxes) are placed in the first 16 columns of $\mathbf{M}_n$. The remaining columns are assigned to intermediates inside the affine block. No adjustments are necessary for the Extract-Linear sub-algorithm. However, the $\mathbf{P}_n^r = \mathsf{Sampl}(\mathbf{P}_n, p)$ function should apply different probabilities of leakage to its columns. Columns 2 to 16 will leak with probability $\mathsf{E}_{\text{left}}(n,p)$. Shares in T^1 to T^{16} will leak with probability one. The remaining columns will leak independently with probability p. The rest of Algorithm 5 is computed as usual.

Estimation Results. By running the algorithm with sufficiently many trails, for $p \leq 0.02$, we obtain the following upper bound:

$$\mathsf{E}_{\text{right}}(n,p) \leq 3p^{n-1}. \tag{56}$$

Adding the leakage of $\mathbb{S}\mathsf{R}_3$ to the combination (55) and (56), for $(n \geq 4, p \leq 0.02)$, we derive the following estimation for the targeted k^1, which uses (15) and (54):

$$\begin{aligned}
\mathsf{E}(n,p) &\leq \mathsf{E}_{\text{right}}(n,p) + \mathsf{E}_{\mathbb{S}\mathsf{R}}(n,p) + \mathsf{E}_{\text{left}}(n,p) \\
&\leq \mathsf{E}_{\mathbb{S}\text{S-box}}\left(n, p + \frac{\mathsf{T}_{\mathbb{S}\mathsf{R}}}{n}\right) + 3\mathsf{E}_{\mathbb{S}\mathsf{R}}(n,p) + \mathsf{E}_{\text{left}}(n,p) \\
&\leq_{(a)} \mathsf{E}_{\mathbb{S}\text{S-box}}\left(n, \frac{4}{3}p\right) + 3p^{0.6n} + 3p^{n-1} \approx 4(5.3p)^{0.3n}.
\end{aligned} \tag{57}$$

The left side of (a) is analytically obtained. However, the right side of it is based on our probability estimation approach and, hence, is only valid in the tested region of (n,p) pairs. The bound based on these estimations is exponentially decaying. Thus, it shows the RPM security of the scheme. Note that the threshold $p \leq 0.02$ is chosen such that the maximum leakage values for the constituent gadgets are all in their tested region.

7.2 Comparison With Previous Results

The expansion-based compiler technique, initially introduced by Ananth et al. [AIS18] and subsequently refined by Belaïd et al. in a series of works [BCP+20, BRT21, BRTV21], offers RPM security through an indirect (simulation-based) proof. In contrast, our approach involves direct security evaluation, as discussed in the Introduction. While both research directions provide quantitative claims based on their respective security metrics within the same leakage model, a comparison of their complexity and achieved security level can provide valuable insights. Below, we provide an overview of the expansion technique.

Expansion Approach. The expansion approach aims to boost the security of a circuit C through iterative masking. Initially, C undergoes masking, resulting in $\mathbb{S}\mathsf{C}$ where the gates of C are substituted with gadgets. This process repeats recursively, with $\mathbb{S}\mathsf{C}$ itself undergoing masking, replacing its gates with gadgets. This iteration persists until the desired security level is achieved. Although this method exponentially inflates the size of the circuit, it can enhance security under specific conditions.

Comparison of Security Claims. In our approach, the size of the final circuit is $\mathcal{O}(|\mathsf{C}|n^2)$, with the security bound being $4(5.3p)^{0.3n}$ as given in (57), specifically for the case of AES. The factor $\mathcal{O}(|\mathsf{C}|n^2)$ represents the standard size expansion in TPM. Our work achieves this size expansion by utilizing only $\mathbb{S}$AND-Rec and $\mathbb{S}$R-SNI gadgets, which are of size $\mathcal{O}(n^2)$.

In contrast, the expansion approach for achieving a security bound of $2^{-\kappa}$ increases the size of the circuit to $\mathcal{O}(|\mathsf{C}|\kappa^e)$, where e is determined by the structure of the basic gadgets used. The value of e ranges from 7.5 in early works to 3.2 in more recent refinements [BCP+20, BRT21]. Therefore, to achieve the same security level as our work, the expansion approach results in a final circuit size of $\mathcal{O}(|\mathsf{C}||2 + 0.3n\log_2(5.3p)|^e)$, exhibiting an asymptotic n^e increase in circuit size, that is higher than our approach.

Furthermore, our approach withstands $p < 0.02$, while the expansion approach withstands a smaller rate $p < 0.007$. However, our work's final security bound expression as a function of (n, p) is based on probability estimations of constituent gadgets. Hence, its validity is only guaranteed in the tested region of (n, p) values.

Acknowledgments

We would like to thank the reviewers and shepherd for their valuable feedback and comments. Bart Mennink is supported by the Dutch Research Council (NWO) under grant VI.Vidi.203.099. Lejla Batina is supported by the Dutch Research Council (NWO) through the PROACT project (NWA.1215.18.014) and TTW PREDATOR project 19782.

References

[ADF16] Marcin Andrychowicz, Stefan Dziembowski, and Sebastian Faust. Circuit Compilers with O(1/ log n) Leakage Rate. In *Proceedings, Part II, of the 35th Annual International Conference on Advances in Cryptology — EUROCRYPT 2016 - Volume 9666*, page 586–615, Berlin, Heidelberg, 2016. Springer-Verlag.

[AIS18] Prabhanjan Ananth, Yuval Ishai, and Amit Sahai. Private Circuits: A Modular Approach. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 427–455, Cham, 2018. Springer International Publishing.

[Ajt11] Miklós Ajtai. Secure Computation with Information Leaking to an Adversary. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 715–724, 2011.

[BBD+16] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. Strong Non-Interference and Type-Directed Higher-Order Masking. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 116–129, 2016.

[BBP+17] Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud. Private Multiplication over Finite Fields. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 397–426, Cham, 2017. Springer International Publishing.

[BCGR24] Julien Béguinot, Wei Cheng, Sylvain Guilley, and Olivier Rioul. Formal Security Proofs via Doeblin Coefficients: Optimal Side-channel Factorization from Noisy Leakage to Random Probing. Cryptology ePrint Archive, Paper 2024/199, 2024. To appear at CRYPTO 2024.

[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, *Advances in Cryptology – CRYPTO 2020*, pages 339–368, Cham, 2020. Springer International Publishing.

[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, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 23–39. Springer, 2016.

[BDF+17] Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel Implementations of Masking Schemes and the Bounded Moment Leakage Model. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 535–566, Cham, 2017. Springer International Publishing.

[BFO23] Francesco Berti, Sebastian Faust, and Maximilian Orlt. Provable Secure Parallel Gadgets. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(4):420–459, 2023.

[BRT21] Sonia Belaïd, Matthieu Rivain, and Abdul Rahman Taleb. On the Power of Expansion: More Efficient Constructions in the Random Probing Model. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 313–343, Cham, 2021. Springer International Publishing.

[BRTV21] Sonia Belaïd, Matthieu Rivain, Abdul Rahman Taleb, and Damien Vergnaud. Dynamic Random Probing Expansion with Quasi Linear Asymptotic Complexity. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021*, pages 157–188, Cham, 2021. Springer International Publishing.

[CFOS21] Gaëtan Cassiers, Sebastian Faust, Maximilian Orlt, and François-Xavier Standaert. Towards Tight Random Probing Security. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 185–214, Cham, 2021. Springer International Publishing.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO' 99*, pages 398–412, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[CS20] Gaëtan Cassiers and François-Xavier Standaert. Trivially and Efficiently Composing Masked Gadgets With Probe Isolating Non-Interference. *IEEE Transactions on Information Forensics and Security*, pages 2542–2555, 2020.

[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, *Advances in Cryptology – EUROCRYPT 2014*, pages 423–440, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[DFS15] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making Masking Security Proofs Concrete. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, pages 401–429, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[DFŻ19] Stefan Dziembowski, Sebastian Faust, and Karol Żebrowski. Simple Refreshing in the Noisy Leakage Model. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019*, pages 315–344, Cham, 2019. Springer International Publishing.

[HRG14] Annelie Heuser, Olivier Rioul, and Sylvain Guilley. Good Is Not Good Enough. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems – CHES 2014*, pages 55–74, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, pages 463–481, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[KPP20] Matthias J. Kannwischer, Peter Pessl, and Robert Primas. Single-Trace Attacks on Keccak. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(3):243–268, 2020.

[LDPDP06] Steven J Leon, Lisette G De Pillis, and Lisette G De Pillis. *Linear Algebra with Applications*. Pearson Prentice Hall Upper Saddle River, NJ, 2006.

[MS23] Loïc Masure and François-Xavier Standaert. Prouff and Rivain's Formal Security Proof of Masking, Revisited. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 343–376, Cham, 2023. Springer Nature Switzerland.

[Nat01] National Institute of Standards and Technology. FIPS PUB 197: Advanced Encryption Standard (AES). Technical report, National Institute of Standards and Technology, 2001.

[PGMP19] Thomas Prest, Dahmun Goudarzi, Ange Martinelli, and Alain Passelègue. Unifying Leakage Models on a Rényi Day. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 683–712, Cham, 2019. Springer International Publishing.

[PP02] A. Papoulis and S.U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill series in electrical and computer engineering. McGraw-Hill, 2002.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against Side-Channel Attacks: A Formal Security Proof. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, pages 142–159, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[RP10] Matthieu Rivain and Emmanuel Prouff. Provably Secure Higher-Order Masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010*, pages 413–427, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[SMY09] François-Xavier Standaert, Tal G. Malkin, and Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, pages 443–461, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[WJZY23] Weijia Wang, Fanjie Ji, Juelin Zhang, and Yu Yu. Efficient Private Circuits with Precomputation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(2):286–309, 2023.