Verification of the security in Boolean masked circuits

Vahid Jahandideh
v.jahandideh@gmail.com

Abstract. We introduce a novel method for reducing an arbitrary δ-noisy leakage function to a collection of ϵ-random probing leakages. These reductions combined with linear algebra tools are utilized to study the security of linear Boolean masked circuits in a practical and concrete setting. The secret recovery probability (SRP) that measures an adversary's ability to obtain secrets of a masked circuit is used to quantify the security. Leakage data and the parity-check relations imposed by the algorithm's structure are employed to estimate the SRP
Both the reduction method and the SRP metric were used in the previous works. Here, as our main contribution, the SRP evaluation task is decomposed from the given $\mathbb{F}_q$ field to $q-1$ different binary systems indexed with i. Where for the ith system, the equivalent δ_i-noisy leakage is reduced optimally to a ϵ_i-random probing leakage with $\epsilon_i = 2\delta_i$. Each binary system is targeting a particular bit-composition of the secret. The $q-1$ derived $\delta_i \leq \delta$ values are shown to be a good measure for the informativeness of the given δ-noisy leakage function.
Our works here can be considered as an extension of the work of TCC 2016. There, only δ-noisy leakage from the shares of a secret was considered. Here, we also incorporate the leakages that are introduced by the computations over the shares.

Keywords: Boolean masking · Noisy leakage · Random probing · Security assessment.

1 Introduction

Side-channel attacks are a significant concern for the security of physically implemented cryptographic algorithms. Their successful application to various realizations of primitives, especially block ciphers, has triggered intense research to understand and prevent these threats. See [25, 24, 15] for pioneering works on the side-channel analysis, and [34, 21, 33, 12] for more recent works.

The first step in developing a mitigation plan for a type of side-channel attack is to find an appropriate mathematical model, known as the *leakage model*, for expressing what an adversary can gain by conducting these attacks. A leakage model is a probabilistic function or a condition that formulates side-information of the adversary. A good survey on the leakage models is provided by [22]. In this work, we will use the *noisy leakage* model to study the security of well-known *Boolean masking* countermeasures in arbitrary-order software implementations of masked linear circuits.

Noisy leakage model. The leakage from power consumption, or electromagnetic emanations, can be modeled by supplying the adversary with a *noisy* version of each variable of the running cryptographic algorithm $\mathcal{A}$. The collection of the obtained noisy information is called the *leakage vector* and is denoted by $\mathbf{L}$. The adversary is also aware of public inputs/outputs of $\mathcal{A}$. Her ultimate goal is to use the public information and the acquired vector $\mathbf{L}$ to recover some non-trivial knowledge about the private inputs/outputs of $\mathcal{A}$. The noisy-leakage model is introduced in [29] and matches well with the experimental outcomes. For a practical use case of this model, see [3].

For a variable $V \in \mathbb{F}_{q=2^k}$ valued during each execution of $\mathcal{A}$, the noisy information that the side-channel adversary will gather in the V's computation interval is modeled by an m-length vector sampled from some probability distribution $\boldsymbol{\nu}(v) \in \mathbb{R}^m$. The input argument v is the actual value taken by V in the current run of $\mathcal{A}$, and $m \geq 1$ is proportional to the sampling frequency. Based on the collected raw leakage data, $\boldsymbol{\nu}(v)$ is an m-dimensional probability distribution, depending on the hardware used in the implementation. The process of characterizing this probability distribution is called *profiling*. Profiling requires access to the cryptographic device with complete control over its inputs and internal randomness [8, 31].

The observed m values are jointly dependent on V. So, by processing these m quantities with some *dimension reduction* techniques, the adversary can obtain a more refined leakage described with a univariate probability distribution $\nu(v) \in \mathbb{R}$. This leakage compression shall not dismiss a noticeable amount of useful data. For dimension reduction methods, see [5].

In the noisy leakage model, for each variable V of $\mathcal{A}$, the adversary knows the probabilistic observation function $\nu(.)$, and receives a single value $\nu(v)$ as part of the leakage vector $\mathbf{L}$.

Software implementation. The noisy leakage model is suitable for software implementation. In software, we assume that a single instruction is being performed at each instance of time, and leakage related to each variable is independent of the leakage corresponding to any other variable.

In a software implementation, the code script of $\mathcal{A}$ is known to the adversary. We assume that this script is only composed of $\mathbb{F}_q$'s basic operations such as AND, XOR, COPY, and RAND. Where RAND is for randomness generation, it has no input and outputs a single uniformly distributed random element of $\mathbb{F}_q$ in each invocation. More precisely, we assume that $\mathcal{A}$ is a *circuit* with no feedback loop [19]. If the compiler performs no *processor optimization* and *multi-threading*, the list of $\mathcal{A}$'s variables is unambiguously defined by the given script. As in [20], we denote this list by $\Sigma_{\mathcal{A}}$.

The adversary is allowed to collect leakage vector $\mathbf{L}$ on each run of $\mathcal{A}$. For unprotected implementations, eventually, $\mathbf{L}$ may leak all the private information of $\mathcal{A}$. To make $\mathcal{A}$ resistant against this leakage, a masking countermeasure is usually adopted.

Boolean masking. In the Boolean masking, a stand-alone variable $V \in \mathbb{F}_q$, by using fresh randomness, is encoded to an n-element vector $\overrightarrow{V} = (V_1, \ldots, V_n)$. Where V_is are random members of $\mathbb{F}_q$ such that $\oplus_{i=1}^{n} V_i = V$ is satisfied. Based on the similarities with secret sharing schemes, V_is are called *shares* of V, and V is referred to as *secret*. parameter $n \in \mathbb{N}$ is the masking order. Vector $\overrightarrow{V}$ is also referred to as an n-sharing for the secret V.

Intuitively, by increasing n, the adversary has to combine more noisy observations to obtain V, making V less accessible to her. In [7], for binary secret with Gaussian distributed noisy observations of its shares, this intuition is proved. However, in general, it may not hold for any noise function $\nu(.)$ [11].

For each variable $V \in \Sigma_{\mathcal{A}}$, by Boolean masking of $\mathcal{A}$, there will be a corresponding n-sharing $\overrightarrow{V}$. This vector replaces V, and all the operations in the masked circuit are equivalently carried out with $\overrightarrow{V}$, without directly invoking the secret V. In the masked circuit, all the variables in $\Sigma_{\mathcal{A}}$ are considered to be secret.

The theory of Boolean masking (at least for block ciphers) is well devolved. There are clear steps defined for masking a given input circuit $\mathcal{A}$ at an arbitrary order n. We denote the masked counterpart of $\mathcal{A}$ with $\mathcal{A}'$. For each value of order n, $\mathcal{A}'_n$ is a separate circuit. The collection of $\{\mathcal{A}'_2, \mathcal{A}'_3, \ldots\}$ forms a *non-uniform* circuit family. $\mathcal{A}'$ -when parameter n is not mentioned- is a representative for this family.

Building blocks of Boolean masking. $\mathcal{A}'_n$ is composed of *shared gadgets*. These gadgets are masked equivalents of basic AND and XOR operations. The shared gadgets operate on n-sharing vectors and produce an n-sharing vector of the output.

The shared gadget for AND is denoted by SAND, and for XOR is denoted by SXOR. $\mathcal{A}'_n$ may also incorporate another shared gadget denoted by SR, which is used to *refresh* the shares of a secret.

The shared gadgets themselves are made up of basic AND, XOR, COPY, and RAND instructions. In the literature, various constructions for the shared gadgets are introduced. In [30, 2, 1], candidates for SR are given, and examples of SAND structures are provided in [19, 3, 13]. For $\mathbb{F}_q$ affine operations, devising a shared gadget is straightforward. See [30] for a more elaborate discussion.

Linear circuits. We define a circuit composed of only $\mathbb{F}_2$-linear basic operations, namely XOR, RAND, and COPY, as a linear circuit. $\mathbb{F}_q$-linear operations such as field squaring and scalar multiplication are not allowed. Most of the practically used SR gadgets are linear circuits.

In [20], it was demonstrated that a linear algorithm can be equivalently described by a *linear system of equations*. For an SR gadget, at any order n, we represent the secret with V and Σ_{SR_n} with $\Sigma_{\mathsf{SR}_n} = \{X_1, X_2, \ldots, X_{T(n)}\}$. Where $T(n) = |\Sigma_{\mathsf{SR}_n}|$ is the number of variables of this shared gadget, including input, output, randomness, and the remaining intermediate variables. X_is and V are

all in $\mathbb{F}_q$. During each execution of the SR gadget, members of Σ_{SR_n} are valued exactly once.

At each order n, a linear SR gadget is equivalently described by a linear system of equations defined as $\mathfrak{P}_n(V, X_1, X_2, \ldots, X_{T(n)}) = 0$. Equivalence for the two representations means that the empirical distribution induced on $\{V, \Sigma_{\mathsf{SR}_n}\}$ are identical for the two cases. In the rest of this paper, we concentrate on this system of equations. $\mathfrak{P}_n$ can be described by a matrix relation as

$$\mathbf{P}_n \times [V, X_1, X_2, \ldots, X_{T(n)}]^\dagger = \mathbf{0}. \tag{1}$$

Where † denotes matrix transpose, and $\mathbf{P}_n$ is a $(T(n)+1) \times D(n)$ matrix with all the entries in $\mathbb{F}_2$. Without loose of generality, we assume that $\mathbf{P}_n$ is full rank and since $D(n) < (T(n)+1)$, we have $\mathsf{Rank}(\mathbf{P}_n) = D(n)$.

According to coding theory, $\mathbf{P}_n$ is the *parity-check* matrix over $\{V, \Sigma_{\mathsf{SR}_n}\}$, and $D(n)$ is the number of independent linear relations between these variables. The adversary will try to solve (1) using side knowledge $\mathbf{L}$ to find the secret variable V. For more in-depth argument, see [20].

Noise amplification. Parity-check relations in $\mathbf{P}_n$ help the adversary recover secret V. Increasing the order n introduces more of these relations. On the other hand, by increasing n, the secret V will be split into more shares, and this puts the adversary in trouble since she needs to combine more noisy observations to get something about V. In combining noisy observations, the chances are that noises will amplify, and V will be more blurred. In this regard, considering relation (1), the main question is that in what circumstances increasing n amplifies the noise of observations and so conceals secret V more. In this paper, we will answer to this dilemma.

MAP adversary. The adversary that makes her decisions with *maximum a posteriori probability* (MAP) rule is called the MAP adversary. Assume that secret V is in $\Sigma_{\mathcal{A}}$, given leakage vector $\mathbf{L}$ corresponding to linear masked circuit $\mathcal{A}'_n$ with describing parity equations $\mathfrak{P}_n$, the adversary tries to find the best estimate for the realized value of V. Let v be that value. The MAP adversary outputs $\widetilde{V}$ for her estimate of v based on the following rule.

$$\widetilde{V} = arg \max_{\alpha \in \mathbb{F}_q} \Pr(V = \alpha | \mathbf{L}, \mathfrak{P}_n) \tag{2}$$

By this choice for $\widetilde{V}$, it is shown that the probability of the correct guess, i.e., $\Pr(\widetilde{V} = v)$, is maximized [18].

Since the MAP adversary is not computationally bounded, the considerable burden of direct computation required by relation (2) is not a challenge. From this viewpoint, the main contribution of this paper is giving a computationally feasible way for bounding $\Pr(\widetilde{V} = v)$ at any masking order n.

Security of SR gadgets. The security of shared gadgets is crucial for the security of their composition in $\mathcal{A}'$. In this paper, we construct a framework for evaluating the security of linear shared gadgets under arbitrary probabilistic leakage observation function $\nu : \mathbb{F}_q \to \mathbb{R}$. Thanks to the reduction to linear systems developed in [20], the proposed framework in this paper can be used to study the security of SAND construction given in [3].

In line with [20], we define *secret recovery probability* (SRP) as a measure for quantifying the security of linear masked algorithms (including linear SR gadgets) that have a single uniform secret $V \in \mathbb{F}_q$. For a fixed noisy observation function $\nu(.)$, $\mathsf{SRP}_V(n)$ for an $\mathcal{A}'$ circuit family is defined as the advantage of the MAP adversary by the following relation.

$$\mathsf{SRP}_V(n) \triangleq \Pr(\widetilde{V} = v|\mathbf{L}, \mathcal{A}'_n) - \frac{1}{|\mathbb{F}_q|} \tag{3}$$

Where v is the realized value of V, and $\widetilde{V}$ is the estimation of the MAP adversary for V based on leakage $\mathbf{L}$ and equations in $\mathfrak{P}_n$.

A smaller value for $\mathsf{SRP}_V^{\mathcal{A}'}(n)$ implies that the adversary has a lower chance of recovering secret V. If $\mathsf{SRP}_V^{\mathcal{A}'}(n)$ decreases by increasing masking order n, then more security will be achieved; at the price of increased complexity.

The circuit family $\mathcal{A}'$ is marked secure against leakage function $\nu(.)$ if $\mathsf{SRP}_V^{\mathcal{A}'}(n)$ is a monotonically decreasing function of n limiting toward zero.

1.1 Problem definition

Given an $\mathbb{F}_2$-linear system of equations $\mathfrak{P}_n(V, X_1, X_2, \ldots, X_{T(n)}) = \mathbf{0}$, with noisy observation vector $\mathbf{L}$ that is randomly sampled based on the distributions in $\{\nu_1(x_1), \nu_2(x_2), \ldots, \nu_{T(n)}(x_{T(n)})\}$, what is the probability that an adversary can obtain secret $V = v$ better than simply guessing it? Where lower-case letters are denoting realized values of their respective random variables.

The framework developed in this paper can be used to study this general case. However, for ease of notations, we assume that the leakage functions ν_1 to $\nu_{T(n)}$ are all equal and are represented by ν. The observation function depends mainly on the structure of the hardware used in the implementation and the ambient noise. So, it seems that sticking to a single function is more relevant to practical use cases of the problem. Note that according to the noisy leakage model, the function ν is completely known to the adversary.

By matrix description of the $\mathbb{F}_2$-linear $\mathfrak{P}_n$, the problem we want to answer can be restated as follows.

For an equation set defined with $\mathbf{P}_n \in \{0,1\}^{(T(n)+1)\times D(n)}$, given a random instance of noisy observation $\mathbf{L}$ as

$$\begin{cases} \mathbf{P}_n \times (V, X_1, X_2, \ldots, X_{T(n)})^\dagger = \mathbf{0} \\ \mathbf{L} \leftarrow \{\nu(x_1), \nu(x_2), \ldots, \nu(x_{T(n)})\} \end{cases}, \tag{4}$$

characterize the behavior of the security measuring function $\mathsf{SRP}_V(n)$ at various masking orders n.

$$\mathsf{SRP}_V(n) \triangleq \Pr(\widetilde{V} = v|\mathbf{L}, \mathbf{P}_n) - \frac{1}{|\mathbb{F}_q|} \tag{5}$$

Where $\widetilde{V}$ is the estimation of the adversary for the value of the initially random secret V. More specifically, at a fixed leakage function $\nu(.)$, for a given family of matrices $\{\mathbf{P}_2, \mathbf{P}_3, \ldots\}$, we want to know that whether increasing n helps to protect V better or not. Practical values for masking order are usually in range of 2 to 5, and in some works even up to 50. Therefore, *concrete* security analysis (for $n \leq 50$) is more demanded than purely *asymptotic* statements.

To make the problem's relevance to actual scenarios explicit, we give realizations for leakage functions and linear SR algorithms in the squeal.

Instances of leakage functions. The first candidate for $\nu(.)$ is the noise-less hamming weight leakage function. For a k-bit value $X \in \{0,1\}^k$, the hamming weight is defined by counting non-zero bits of X by relation $\mathsf{HW}(X) = \Sigma_{i=1}^{k} X^i$, where the X^is are bits of X as $X = [X^1, X^2, \ldots, X^k]$. Hamming distance leakage for X, assuming a constant known reference state R_X, is defined as $\mathsf{HD}(X, R_X) = \mathsf{HW}(X \oplus R_X)$. See [4] for a justification.

Hamming weight leakage is very informative; the adversary receiving $\mathsf{HW}(X)$ can unambiguously determine $\oplus_{i=1}^{k} X^i$ with the following simple identity, where $\mathsf{mod}(., 2)$ means reminder of division with 2.

$$\oplus_{i=1}^{k} X^i = \mathsf{mod}(\mathsf{HW}(X), 2) \tag{6}$$

So, even for an isolated n-sharing $\overrightarrow{V} = \{V_1, V_2, \ldots, V_n\}$, with leakage observation $\mathbf{L} = \{\mathsf{HW}(V_1), \mathsf{HW}(V_2), \ldots, \mathsf{HW}(V_n)\}$, nothing prevents the adversary from obtaining XOR of bits of the k-bit secret V as

$$\oplus_{i=1}^{k} V^i = \oplus_{i=1}^{n} \mathsf{mod}(\mathsf{HW}(V_i), 2). \tag{7}$$

In reality, leakage measurements are always buried in noise and interference. Hence, a more realistic nomination for leakage function is scaled hamming weight leakage perturbed by an additive random value. Such as, $\nu(V) = a\mathsf{HW}(V) + b$. With $a \leq 1$, and b sampled from a normal distribution with known mean and variance.

Some processors leak a much-restricted amount of usable data. The following probabilistic function is an example [26].

$$\mathsf{ZV}(v) = \begin{cases} a \leftarrow \mathcal{N}(\mu_A, \sigma_A) & \text{if } v = 0 \\ b \leftarrow \mathcal{N}(\mu_B, \sigma_B) & \text{otherwize} \end{cases} \tag{8}$$

Where $\mathcal{N}(\mu, \sigma)$ is a normal distribution with the specified parameters. This leakage function assumes that power consumption for zero value is different from other values . This difference is exploited as the leakage source.

Device specific observation function. Generic leakage functions are utilized when the adversary has not complete access to the targeted hardware. Nevertheless, if a copy of targeted hardware is at her disposal, she can estimate the relevant observation function by profiling, then dimension reduction, and finally, PDF approximation techniques.

Instances of linear circuits. The $\mathsf{SRP}(n)$ estimation method developed in this paper is applicable to any linear circuit. However, our numeric examples are only calculated for a particular refreshing algorithm proposed in [1]. This SR that we will refer to as SR-SNI has very suitable properties for composing with other shared gadgets [1, 20].

We denote the input n-sharing of SR-SNI with the vector $\overrightarrow{V}_0$ and the output n-sharing with $\overrightarrow{V}_1$. Both vectors are n-sharing for the same secret V, So, $\oplus_{i=1}^{n} V_{i,0} = \oplus_{i=1}^{n} V_{i,1}$. Where $V_{i,j}$ represents the ith share of vector $\overrightarrow{V}_j$.

The pseudocode for SR-SNI is given in algorithm 1.

Algorithm 1 SR-SNI

Input The n-sharing $\overrightarrow{V}_0 = (V_{1,0}, V_{2,0}, \ldots, V_{n,0})$
Output An n-sharing $\overrightarrow{V}_1 = (V_{1,1}, V_{2,1}, \ldots, V_{n,1})$ for the same secret V

1: **for** $i = 1$ **to** n **do**
2: **for** $j = i + 1$ **to** n **do**
3: $r \leftarrow^{\$} \mathbb{F}_q$
4: $V_{i,0} = V_{i,0} \oplus r$
5: $V_{j,0} = V_{j,0} \oplus r$
6: **for** $i = 1$ **to** n **do**
7: $V_{i,1} = V_{i,0}$
8: **return** $\overrightarrow{V}_1$

SR-SNI is a linear circuit. For input size n, it uses $(n^2 - n)/2$ randomness variables and $n^2 - n$ times XOR operations. By direct counting, we obtain $T(n) = n + 3(n^2 - n)/2 + n$; the first and last terms are for the input and output variables. Since only randomness and input variables are independent, we will have $d(n) = T(n) - (n^2 - n)/2 - n + 1$; the reason for addition with 1 is that the secret is also a linearly dependent variable. Parity-check relations in $\mathbf{P}_n$ are derived with the Gaussian-elimination technique. See [20] for their automated tool of obtaining $\mathbf{P}_n$.

Of course, $\mathbf{P}_n$'s order of variables is not a matter, but their set and count are influential. For example, in algorithm 1, by some obvious optimization, one may omit the last **for** loop, which is for copying the results and hence decrease the number of internal variables. In this way, a new $\mathbf{P}_n$ will result. Therefore, exact results are dependent on the details of code implementations. Nevertheless, subtle changes like a single variable-copy-loop do not affect the final calculations for $\mathsf{SRP}(n)$ in a detrimental way.

1.2 Our contributions

We initiate a framework for assessing the security of Boolean masked circuits at arbitrary noisy leakage distributions. For instance, we provide an answer for the mentioned problem in 1.1.

A formerly known reduction from noisy leakage to random probing leakage (introduced in [9]) has been used to answer this problem in [20], by invoking simple linear-algebra tools. Unfortunately, the reduction is not tight at $\mathbb{F}_q$. As a result, for most of the leakage functions, security evaluation is not directly possible. We close the gap for linear circuits by giving a new $\mathbb{F}_2$-based equivalent analysis. In this path, we use the fact that reduction from noisy leakage to random probing leakage in $\mathbb{F}_2$ is optimal.

We also extend the results of [11]. For noisy observation of shares of V_i, in [11], a minimum requirement on the output distribution of $\nu(.)$ for noise amplification was derived. In this paper, we tackle the same question, but this time noisy observations are not limited to only isolated shares; as detailed in 1.1, a complete set of variables appearing in a linear circuit are targeted. Quite surprisingly, we obtain a similar requirement on the structure of $\nu(.)$.

Finally, it is to note that the method we have applied to answer the problem in 1.1 is just a proof technique, which is based on reduction and simulation. Hence it can not be used to recover the value of V in actual attack scenarios.

1.3 Preliminaries

Notations. A k-bit extended field, with $q = 2^k$ members, is denoted by $\mathbb{F}_q$. Random variables (RVs) are shown by capital letters as V, and their realizations are shown by corresponding lower-case letters as v. A random variable with a vector over it, is used for indication of an n-sharing. Such as $\overrightarrow{V} = \{V_1, \ldots, V_n\}$, where subscript-indexed RVs are shares. For a k-bit variable V, the corresponding bits are shown by $V = [V^1, \ldots, V^k]$, where V^is are binary and superscript-indexed, with V^1 being MSB.

For two RVs X and Y, their joint probability distribution at $(X = x, Y = y)$ is denoted by $\mathbb{P}_{X,Y}(x, y)$, and their marginal distribution are denoted by $\mathbb{P}_X(x)$ and $\mathbb{P}_Y(y)$, respectively. Conditional distribution of Y at $Y = y$ conditioned on $X = x$ is written as $\mathbb{P}_{Y|X=x}(y)$. Support of a random value X is denoted by $\Omega(X)$ and is collection of all values x in the domain of X that $\mathbb{P}_X(x) \neq 0$.

Cardinality of a set like $\mathbb{F}_q$ is shown by $|\mathbb{F}_q|$. For uniform assignment of a random variable V from a set as $\mathbb{F}_q$, we write $v \leftarrow^{\$} \mathbb{F}_q$, and for assignment of value of V based on a distribution as (μ,σ)-normal distribution, we write $v \leftarrow \mathcal{N}(\mu, \sigma)$.

Boldface letters are used for arrays and matrices, calligraphic font is used for algorithms, and known probability distributions. A prim over an algorithm name is used for identifying the masked counterpart of that algorithm. Field operations and some functions are written by sans-serif font family, for more clarity.

Other notations used in this paper are described at first place that they appear.

1.4 Metrics for quantifying the noise

For any value $\alpha \in \mathbb{F}_q$, $\nu(\alpha)$ is a known probability distribution over $\mathbb{R}$. To measure how much usable data the random variable $\zeta = \nu(V)$ can deliver about an initially-uniform random variable V, *mutual information* and *statistical distance* are commonly used metrics. The mutual information is mainly used by applied side-channel investigators and is defined as

$$\gamma = \mathsf{MI}(V;\zeta) \triangleq \sum_{\alpha \in \mathbb{F}_q} \int_{\beta \in \Omega(\zeta)} \mathbb{P}_{V,\zeta}(\alpha,\beta) \log \frac{\mathbb{P}_{V,\zeta}(\alpha,\beta)}{\mathbb{P}_V(\alpha)\mathbb{P}_\zeta(\beta)} \, d\beta \tag{9}$$

Value of γ when the base of log is 2, lies in $[0, k]$. A smaller γ implies a less amount of valuable data is given by this leakage function.

Conditional statistical distance is used mainly in theoretical studies and is defined as

$$\delta = \mathsf{SD}(V;V|\zeta) \triangleq \frac{1}{2} \sum_{\alpha \in \mathbb{F}_q} \int_{\beta \in \Omega(\zeta)} |\mathbb{P}_{V,\zeta}(\alpha,\beta) - \mathbb{P}_V(\alpha)\mathbb{P}_\zeta(\beta)| \, d\beta \tag{10}$$

Value of δ, for initially-uniform V, resides in $[0, 1/2]$. A smaller δ implies a less informative leakage function. A leakage function with conditional statistical distance δ is referred to as a δ-noisy leakage function.

Parameters δ and γ are related through several inequalities. For low values of δ, the (generalized) Pinsker's inequality, as in relation (11), can tightly link these two metrics [14].

$$2\delta^2 + \frac{4}{9}\delta^4 + O(\delta^6) \leq \gamma \tag{11}$$

In section 3.2, we will introduce a new approach for quantifying informativeness of $\nu(.)$. There, instead of directly targeting V, we will consider information that learning $\nu(V)$ can deliver about $h(V)$. Where $h : \mathbb{F}_q \to \{0, 1\}$ can be any of Boolean combinations of bits of V.

1.5 Random probing model

Another leakage model that is considered in the side-channel analysis is the *random probing model*. This model is mainly of theoretical interest; it is essentially the same as the noisy leakage model with the observation function $\nu(V) = \phi(V)$. Where $\phi : \mathbb{F}_q \to \{\mathbb{F}_q, \perp\}$ is a memory-less probabilistic *erasure* function, with parameter $\epsilon \in [0, 1]$. Where 1-ϵ is the erasure probability, and the special symbol $\perp \notin \mathbb{F}_q$ represents the erasure event. This model is also known as ϵ-random probing model.

For every value $v \in \mathbb{F}_q$, $\phi(v) = v$ with probability ϵ, and $\phi(v) = \perp$ with probability $1 - \epsilon$ [9]. For the random probing model, we can write

$$\forall v \in \mathbb{F}_q; \quad \Pr(\phi(v) = v) = \epsilon. \tag{12}$$

Where Pr is only taken on the ϕ's internal randomness. With this leakage model, the problem mentioned in 1.1 can be readily answered by standard linear-algebra techniques [20].

A closely related model is the *average probing model* introduced in [10]. In this new model, the erasure probability is averaged over the entire input alphabet, and the probability of the erasure event depends on the value of the input variable. Therefore, receiving $\perp$ may leak some data about V. This makes difficulties in the usage of linear-algebra tools for solving our main problem. In the rest of this paper, we will use only the results developed for the random probing model.

2 Known results

2.1 Relations with coding theory

In coding theory, for linear codes, symbols of code-words have known linear dependencies, usually referred to as parity-check relations. The channel through which code-words are transmitted determines the observation function ν. A decoder algorithm receiving a noisy code-word uses parity-check relations to recover some code-word elements. Motivated with these similarities, we briefly review the general decoder algorithms that apply to our problem in the follow-up.

Belief Propagation. The common practice for solving a set of equations given an initial noisy side-information about its engaged unknowns, the same as our problem in 1.1, is the well-known *belief propagation* method.

Belief propagation, also known as *sum-product*, is a sub-optimal recursive algorithm for characterizing a solution for a (linear or some non-linear) system of equations. Its convergence to the correct solution (if there is any) depends on the structure of the composing equations. For *spars* parity relations, as in low-density parity-check (LDPC) codes, belief propagation performs well in the presence of enough side information [28, 27].

Belief propagation ideas are used in some side-channel attack scenarios [17, 32, 3]. However, due to its sub-optimal performance, we cannot use it to certify a Boolean masking's security. More concretely, the adversary may find an approach with a success rate higher than what is inferred by a belief prorogation.

Gaussian elimination. For the particular case of random probing leakage that corresponds to the erasure channel in coding theory, Gaussian elimination techniques are adopted to characterize linear systems' solutions. For random probing, we will use this tool here, same as [20]. For application in coding theory, such as LDPC codes, see [6].

Exhaustive search. The ML rule in coding theory is similar to the MAP rule described here. Both need an exhaustive search on the variables space to find all possible solutions. They achieve the theoretically optimum decision outcomes for solving any system of equations with any sort of prior distribution knowledge.

2.2 Minimum requirement for noise amplification

Considering an isolated n-sharing $\overrightarrow{V} = \{V_1, V_2, \ldots, V_n\}$ in $\mathbb{F}_{q=2^k}$, with leakage random variable $\mathbf{L} = \{\nu(V_1), \nu(V_2), \ldots, \nu(V_n)\}$ distributed according to some noise function ν, does increasing n decreases SRP (secret recovery probability) for any noise function ν, or there should be some restrictions on the output distribution of ν? In [11] this was answered by expressing the required noise constraint.

For instance, if only MSB of each variable is leaking, i.e., $\nu(V) = V^1$, it is easy to see that, since the rest $k-1$ bits are independently and uniformly distributed, $\mathsf{SRP}_V(n)$ is calculated as

$$\mathsf{SRP}_V(n) = \frac{1}{2^{k-1}} - \frac{1}{2^k}. \tag{13}$$

Therefore, $\mathsf{SRP}_V(n)$ is independent of n, which means that increasing masking order does not decrease SRP value. For this example, value of metric δ, computed by relation (10), is obtained $1/2$.

In [11], it is proved that if leakage function is δ-noisy for $\delta < 1/2$, then $\mathsf{SRP}_V(n)$ will be a monotonically decreasing function of n, with

$$\lim_{n\to\infty} \mathsf{SRP}_V(n) \to 0. \tag{14}$$

2.3 From noisy leakage to random probing leakage

In the remarkable work of [9], it was observed that for any value v, $\nu(v)$ is simulatable from $\phi(v)$, for some constant (independent of v) parameter ϵ, without any other access to the value v. More concretely, there is a probabilistic function f with domain $\{\mathbb{F}_q, \perp\}$ that for random variables $\xi = f(\phi(V))$ and $\zeta = \nu(V)$, we can write

$$\forall\ \alpha \in \mathbb{F}_q, \beta \in \Omega(\zeta); \quad \mathbb{P}_{V,\xi}(\alpha, \beta) = \mathbb{P}_{V,\zeta}(\alpha, \beta). \tag{15}$$

Therefore, the leakage vector sampled by function ν is indistinguishable from a leakage vector first sample by the erasure function ϕ and then processed by f. This result shows that noisy leakage is *reducible* to random probing leakage. This one-directional reduction helps to evaluate security in the random probing model and then interpret the obtained results for the noisy leakage model.

In [9], The minimum value for the parameter ϵ is obtained as

$$\epsilon_{min} = 1 - \int_{\beta\in\Omega(\zeta)} \min_{\alpha\in\mathbb{F}_q} \mathbb{P}_{V|\zeta=\beta}(\alpha)\ d\beta. \tag{16}$$

For any $\epsilon \geq \epsilon_{min}$, one can devise a function f. For two random variables V and ζ, parameter $1 - \epsilon_{min}$ is independently defined in communication theory as Doeblin's coefficient [16]. In [9], the relation between δ and ϵ_{min} was shown to be $\epsilon_{min} \leq \delta q$.

This reduction is not tight. Unfortunately, there is a considerable gap between the noisy leakage model and the reduced to random probing model at fields with $q > 2$. In a sense that, for some noise functions, the reduction requires high values of ϵ. For higher values of ϵ, either the masked circuit will be insecure, or its security proof will be difficult. In both cases, the reduction will be useless.

For example, considering $\nu(v) = \mathsf{HW}(v)$, the value of ϵ_{min} is obtained 1, which means that $\phi(v)$ should always give v with probability 1. In this way, the reduction does not distinguish the noise-less identity observation $\nu(v) = v$ and $\mathsf{HW}(v)$. One may think that this is because the hamming weight function is very informative. See the discussion around relation (6). However, for $\nu(V)$, defined as

$$\mathsf{ZV}^{\circ}(v) = \begin{cases} \mu_A & \text{if } v = 0 \\ \mu_B & \text{otherwize} \end{cases} \tag{17}$$

with $\mu_A \neq \mu_B$, ϵ_{min} is again obtained 1. This time, $\nu(V)$ is just distinguishing only one number of its q possible input values. This observation function is the deterministic version of $\mathsf{ZV}(v)$, defined by function (8).

In this paper, we will show that for binary variables, i.e., for $q = 2$, the reduction from noisy leakage to random probing is tight. For this case, we will prove that $\epsilon_{min} = 2\delta$.

Application to our problem. For computing $\Pr(\widetilde{V} = v|\mathbf{L}, \mathbf{P}_n)$, discussed in relation (5), we can use the proposed reduction as follows.

$$\begin{aligned} \Pr(\widetilde{V} = v|\mathbf{L}, \mathbf{P}_n) &=_a \Pr(\widetilde{V} = v|\{\nu(x_1), \ldots, \nu(x_{T(n)})\}, \mathbf{P}_n) \\ &=_b \Pr(\widetilde{V} = v|\{f[\phi(x_1)], \ldots, f[\phi(x_{T(n)})]\}, \mathbf{P}_n) \\ &\leq_c \Pr(\widetilde{V} = v|\{f[\phi(x_1)], \phi(x_1), \ldots, f[\phi(x_{T(n)})], \phi(x_{T(n)})\}, \mathbf{P}_n) \\ &=_d \Pr(\widetilde{V} = v|\{\phi(x_1), \ldots, \phi(x_{T(n)})\}, \mathbf{P}_n) \\ &=_e \Pr(\widetilde{V} = v|\mathbf{L}^r, \mathbf{P}_n) \end{aligned} \tag{18}$$

Where (a) is by direct substitution for the leakage vector. (b) follows from the fact that for each entry of $\mathbf{L}$ as $\nu(x_i)$, $f[\phi(x_i)]$ has an indistinguishable distribution. (c) follows by the fact that extra information increases the probability of correct guess, i.e., the event $\widetilde{V} = v$. (d) follows from the fact that knowledge of $t = \phi(x_i)$ is sufficient and $f(t)$ is a dependent variable and is unnecessary. In (e), the new leakage vector in the random probing model is represented by $\mathbf{L}^r$.

To distinguish the evaluation of SRP based on the random leakage $\mathbf{L}^r$, we use a new notation as $\mathsf{SRP}(n, \epsilon)$. For any $\epsilon \geq \epsilon_{min}$, based on the relation (18), we can write $\mathsf{SRP}(n) \leq \mathsf{SRP}(n, \epsilon)$

In this setting, a linear masked algorithm (with describing matrix family $\mathbf{P}$), is declared secure, if there exists an ϵ_0, such that for any $\epsilon < \epsilon_0$, $\mathsf{SRP}(n, \epsilon)$ is a decreasing function of n, with limiting toward zero at sufficiently big values of n [20].

2.4 Security in the random probing model

Evaluation of $\mathsf{SRP}(n, \epsilon)$ requires the computation of $\Pr(\widetilde{V} = v|\mathbf{L}^r, \mathbf{P}_n)$, which can be simplified by directly substituting the information given by $\mathbf{L}^r$ into the equations in $\mathbf{P}_n$. We show the resultant new matrix by $\mathbf{P}_n^r$. So, we can write

$$\Pr(\widetilde{V} = v|\mathbf{L}^r, \mathbf{P}_n) = \Pr(\widetilde{V} = v|\mathbf{P}_n^r). \tag{19}$$

For each element of $\mathbf{L}^r$ that we have $\phi(x_i) = x_i$, by substitution in $\mathbf{P}_n$, the corresponding column in $\mathbf{P}_n$ will be known. For removing a revealed column, we can replace it with an all-zero column.

As comprehensively discussed in [20], the rest of the work is straightforward. First, the *row reduced echelon form* of $\mathbf{P}_n^r$, with standard row-based Gaussian elimination approach is obtained. We denote the resultant matrix by $\mathbf{G}$, and the Gaussian elimination function with $\mathsf{Gaussian\text{-}Elim}$, that is

$$\mathbf{G} = \mathsf{Gaussian\text{-}Elim}(\mathbf{P}_n^r). \tag{20}$$

For a detailed description of $\mathsf{Gaussian\text{-}Elim}$, refer to [20], or check the wiki page of the row reduced echelon form.
Matrix $\mathbf{G}$ has the same size as $\mathbf{P}_n^r$ and will identify how much information $\mathbf{P}_n^r$ can give about V. If $\mathbf{G}$ has no *free* variables in the first row, i.e., if $\mathbf{G}(1, 2 \text{ to } T(n) + 1) = \mathbf{0}$, then $\mathbf{P}_n^r$ can determine V uniquely. So, we will have

$$\Pr[\widetilde{V} = v|\mathbf{G}(1, 2 \text{ to } T(n) + 1) = \mathbf{0}] = 1 \tag{21}$$

However, if there be a free variable in the first row, which is the row containing V as its *pivot*, then $\mathbf{P}_n^r$ reveals no information about V. So, only the guessing option remains, which means

$$\Pr[\widetilde{V} = v|\mathbf{G}(1, 2 \text{ to } T(n) + 1) \neq \mathbf{0}] = \frac{1}{q}. \tag{22}$$

Finally, by expanding the conditional probability, we can write

$$\begin{aligned} \Pr(\widetilde{V} = v|\mathbf{P}_n^r) = & \\ & \Pr[\mathbf{G}(1, 2 \text{ to } T(n) + 1] = \mathbf{0}) + \frac{1}{q} \Pr[\mathbf{G}(1, 2 \text{ to } T(n) + 1) \neq \mathbf{0}] \\ & = (1 - \frac{1}{q}) \Pr[\mathbf{G}(1, 2 \text{ to } T(n) + 1) = \mathbf{0}] + \frac{1}{q} \end{aligned} \tag{23}$$

Since $\mathbf{P}_n$ is Boolean, $\mathbf{G}$ will also be binary, and as a result, the probability of the event $\mathbf{G}(1, 2 \text{ to } T(n)+1) = \mathbf{0}$ will be independent of the size of the underlying field.

For a fixed pair of (n, ϵ), by the Monte Carlo approach, with sufficient number of random trials, $\mathsf{SRP}(n, \epsilon)$ can be estimated.

2.5 Security of linear masked algorithms

For various masked algorithms, including the SR-SNI refreshing gadget, $\mathsf{SRP}(n, \epsilon)$ is evaluated in [20]. For the SR-SNI, that is described with the pseudocode in algorithm 1, $\mathsf{SRP}(n, \epsilon)$, for $\epsilon \leq 0.15$, is bounded as

$$\mathsf{SRP}(n, \epsilon) \leq (1 - \frac{1}{q})\epsilon^{0.6n}. \tag{24}$$

This bound is obtained by curve-fitting the results for orders $n \leq 30$.

3 Security evaluation under general noise distribution

The security evaluation approach described so far is only applicable for observation functions that their corresponding ϵ_{min} is adequately small. Nevertheless, for leakage functions such as HW and ZV°, since ϵ_{min} is high, this approach fails. Then, it remains unclear how to bound $\mathsf{SRP}(n)$ for these leakages. In this section, we will answer this question. Our approach is based on an optimal reduction and can be used to assess any leakage function.

3.1 Overview of our approach

Instead of evaluating the informativeness of the pair $(\mathbf{P}_n, \mathbf{L})$ regarding the secret V, we will consider XOR of bits of V as a new target. For a k-bit variable V, there are $2^k - 1$ such XOR combinations. We will demonstrate that for each of these binary combinations, parity relations are governed by $\mathbf{P}_n$. However, the leakage vector $\mathbf{L}$ should be tailored. It turns out that there is an equivalent noise description for each XOR combination.

On the other hand, we prove that in $\mathbb{F}_2$, the reduction from noisy leakage to random probing leakage is optimal. So, we can give an optimal estimation for the probability of recovering each composition of bits of V. We also illustrate how these probabilities of XOR combinations collectively determine the overall security of V.

Practitioners not interested in the technical details can jump directly to part 3.3, where we review the step-by-step procedures required for assessing the security of a linear masked algorithm at arbitrary noisy observation function.

3.2 Technical details

Binary projections of the secret. In [20], it is proved that if for a k-bit random variable V, $\mathsf{SRP}_V(n)$ is a decreasing function of n, then for any binary function $h : \{0,1\}^k \to \{0,1\}$, $\mathsf{SRP}_{h(V)}(n)$ is also a decreasing function of n. Here, we will prove that the reverse side of this statement is also valid.

First, we define a family of binary functions $\mathbf{H} = \{h_1, h_2, \ldots, h_{2^k-1}\}$, with $h_I(V)$, for $I \in [1, 2^k - 1]$, computed as

$$h_I(V) = \langle I, V \rangle = \oplus_{i=1}^{k} I^i V^i. \tag{25}$$

Where I^j is the jth bit of I in a k-bit representation, with I^1 being the MSB. For two same-length arrays (here bit-arrays) I and V, $\langle I, V \rangle$ is known as their *inner product*. For the practically interesting case of 8-bit variables, $\mathbf{H}$ has 255 members.

Lemma 1. *If for any $h_I \in \mathbf{H}$, $\mathsf{SRP}_{h_I(V)}(n)$ is decreasing function of n and limiting to zero, then $\mathsf{SRP}_V(n)$ is also a decreasing function of n with limiting to zero.*

Proof. See appendix A.

Remark 1. The idea of investigating bit-combinations of V is inspired by the Goldreich–Levin theorem from the *hard-core predicates* theory, which is of totally independent interest. There, it is proved that if it is computationally hard to obtain V given a side-information, then there is a combination of bits of V that is distributed evenly, from the perspective of any computationally bounded observer, and the bias of distribution decreases as the security parameter increases. See chapter 8 of [23] if interested in this outlined resemblance. Note that the proof method we have used for this lemma does not depend on the hard-core predicates' arguments.

Let's turn back to the study of the inner product of I and V. Given leakage vector $\mathbf{L}$ and the describing Boolean matrix $\mathbf{P}_n$, $\mathsf{SRP}_{h_I(V)}(n)$, for a fixed I, assuming initially-uniform V, is computed as the following.

$$\mathsf{SRP}_{h_I(V)}(n) = |\Pr(\widetilde{h_I}(V) = h_I(v)|\mathbf{L}, \mathbf{P}_n) - \frac{1}{2}| \tag{26}$$

Where $\widetilde{h_I}(V)$ is the random variable representing the outcome of the MAP decision for value of $h_I(v)$ given side-information $\mathbf{L}$ and $\mathbf{P}_n$, and v is the realized value of V.

$$\widetilde{h_I}(V) = arg \max_{\alpha \in \{0,1\}} \Pr(h_I(V) = \alpha|\mathbf{L}, \mathbf{P}_n) \tag{27}$$

From $\mathbb{F}_q$ to $\mathbb{F}_2$. Our main challenge is the conversion of the MAP task from $\mathbb{F}_q$ to $\mathbb{F}_2$. First, we prove a beneficial property for the Boolean $\mathbf{P}_n$ matrix. We show that in an $\mathbb{F}_2$-linear system as

$$\mathbf{P}_n \times [V, X_1, \ldots, X_{T(n)}]^\dagger = \mathbf{0} \tag{28}$$

the value of $\langle I, V \rangle$ can be obtained directly, i.e., without first solving for V.

Lemma 2. *Since entries of $\mathbf{P}_n$ are binary, for any $I \in [1, 2^k - 1]$, the system $\mathbf{P}_n \times [V, X_1, \ldots, X_{T(n)}]^\dagger = \mathbf{0}$ implies that*

$$\mathbf{P}_n \times [\langle I, V \rangle, \langle I, X_1 \rangle, \ldots, \langle I, X_{T(n)} \rangle]^\dagger = \mathbf{0} \tag{29}$$

Proof. See appendix B.

Lemma 2 paves the way for mapping the maximization problem in relation (27) from a q-ary field to a binary one.

To grasp-deeply what condition $\mathbf{P}_n$ means, assume set $\mathcal{S}_n$ is the collection of all possible solutions of the system $\mathbf{P}_n$. At least, the realized values for $\{V, X_1, \ldots, X_{T(n)}\}$ are in $\mathcal{S}_n$. However, there may be other solutions in $\mathcal{S}_n$, as well. Each member of $\mathcal{S}_n$ is a $(T(n)+1)$-element vector that is orthogonal to $\mathbf{P}_n$, i.e., it satisfies relation (28). We show the ith member of $\mathcal{S}_n$ with $\mathbf{s}_i$ and the jth entry of $\mathbf{s}_i$ with scalar $s_{j,i}$. By this notation, the value corresponding to the secret will be $s_{1,i}$.

We can replace the condition $\mathbf{P}_n$ in the maximization of relation (27) with the set $\mathcal{S}_n$, since both have the same meaning, i.e., they both impose the same constraints on the possible values of the variables. By expanding the conditional probability, we can write

$$arg \max_{\alpha \in \{0,1\}} \Pr(h_I(V) = \alpha | \mathbf{L}, \mathcal{S}_n) = arg \max_{\alpha \in \{0,1\}} \frac{1}{|\mathcal{S}_n|} \sum_{\mathbf{s} \in \mathcal{S}_n} \Pr(h_I(V) = \alpha | \mathbf{L}, \mathbf{s}) \tag{30}$$

Likewise, for a fixed $I \in [1, 2^k - 1]$, assume set $\mathcal{S}_n^I$ be the collection of all binary $(T(n)+1)$-tuples satisfying (29). With these definitions, we are ready to tackle with the burden of obtaining $\widetilde{h_I}(V)$.

Lemma 3. *For the task of computation of $\widetilde{h_I}(V)$ we can write*

$$arg \max_{\alpha \in \{0,1\}} \Pr(h_I(V) = \alpha | \mathbf{L}, \mathcal{S}_n) = arg \max_{\alpha \in \{0,1\}} \Pr(h_I(V) = \alpha | \mathbf{L}, \mathcal{S}_n^I). \tag{31}$$

This lemma expresses that for computing some statics on $\langle I, V \rangle$, knowledge of $\mathcal{S}_n^I$ is sufficient. This lemma confines the parity-check relations into a binary field. The next step is to find an equivalent expression for the leakage vector $\mathbf{L} \leftarrow \{\nu(x_1), \nu(x_2), \ldots, \nu(x_{T(n)})\}$ based on the mapped realized values in $\{\langle I, x_1 \rangle, \langle I, x_2 \rangle, \ldots, \langle I, x_{T(n)} \rangle\}$.

Let's define a new bi-variate probability distribution $\mathbb{P}_{A,\zeta_I}(\alpha,\beta)$, with $\Omega(A) = \{0,1\}$ and $\Omega(\zeta_I) = \Omega(\zeta)$, as

$$\mathbb{P}_{A,\zeta_I}(\alpha,\beta) = \sum_{v\in\{0,1\}^k, \langle I,v\rangle=\alpha} \mathbb{P}_{V,\zeta}(v,\beta) \tag{32}$$

In this definition, we explicitly used the fact that for each element of $\mathbb{F}_q$ there is a unique k-bit equivalent representation. With this definition, it is easy to verify that, for any I, we have

$$\begin{aligned} &\mathbb{P}_A(0) = \mathbb{P}_A(1) = \frac{1}{2} \\ &\mathbb{P}_{\zeta_I}(\beta) = \mathbb{P}_{\zeta}(\beta) \end{aligned} \tag{33}$$

We also define a new probabilistic function $\lambda_I(b)$ for sampling values based on the conditional probability $\mathbb{P}_{\zeta_I|A=\alpha}$. For $b \in \{0,1\}$, we have

$$\Pr(\lambda_I(b) = \beta) = \mathbb{P}_{\zeta_I|A=b}(\beta). \tag{34}$$

Note that the random variables A and ζ_I are also related as $\zeta_I = \lambda_I(A)$.

Accordingly, we define the marginal leakage vector, for a given fixed I, as

$$\mathbf{L}_I \leftarrow \{\lambda_I[\langle I, x_1\rangle], \lambda_I[\langle I, x_2\rangle], \ldots, \lambda_I[\langle I, x_{T(n)}\rangle]\}. \tag{35}$$

In the following lemma, it is proved that $\mathbf{L}_I$ is sufficient for our maximization problem.

Lemma 4. *Computation of $\widetilde{h_I}(V)$ can be further simplified with $\mathbf{L}_I$ as*

$$arg \max_{\alpha\in\{0,1\}} \Pr(h_I(V) = \alpha|\mathbf{L}, \mathcal{S}_n^I) = arg \max_{\alpha\in\{0,1\}} \Pr(h_I(V) = \alpha|\mathbf{L}_I, \mathcal{S}_n^I). \tag{36}$$

Proof. This and lemma 3 are proved in appendix C.

By this lemma, the job of bringing the computation of $\widetilde{h_I}(V)$ from $\mathbb{F}_q$ to $\mathbb{F}_2$ is completed. The rest of the work is mainly around applying the reduction from noisy leakage in $\mathbf{L}_I$ to a matching random probing leakage vector.

Reduction from binary noisy leakage. Recall that $\mathcal{S}_n^I$ is the collection of the solutions of $\mathbf{P}_n$ in (29). We can switch between $\mathcal{S}_n^I$ and its matrix format $\mathbf{P}_n$ in the foregoing MAP problem freely. However, to avoid ambiguity, we show the describing matrix corresponding to $\mathcal{S}_n^I$ with $\mathbf{P}_n^I$. This is somewhat an abuse of notations since both matrix $\mathbf{P}_n$ and $\mathbf{P}_n^I$ are equal. The difference is that $\mathbf{P}_n$ is over $\mathbb{F}_q$ variables $\{V, X_1, \ldots\}$, and $\mathbf{P}_n^I$ is over $\mathbb{F}_2$ variables $\{\langle I, V\rangle, \langle I, X_1\rangle, \ldots\}$. With this explanation, we can write

$$arg \max_{\alpha\in\{0,1\}} \Pr(h_I(V) = \alpha|\mathbf{L}_I, \mathcal{S}_n^I) = arg \max_{\alpha\in\{0,1\}} \Pr(h_I(V) = \alpha|\mathbf{L}_I, \mathbf{P}_n^I). \tag{37}$$

What remains is to use the reduction from noisy leakage to random probing leakage in a similar way as was done in eq. (18). To ease tracking of the logic, we repeat the steps here. Let's define

$$\mathbf{L}_I^r = \{\phi_b[\langle I, x_1\rangle], \phi_b[\langle I, x_2\rangle], \ldots, \phi_b[\langle I, x_{T(n)}\rangle]\}$$

for the random probing peer of the leakage vector $\mathbf{L}_I$. For a fixed I, the binary-input probabilistic function $\phi_b : \{0,1\} \to \{0,1,\perp\}$ is a memory-less erasure function, associated with some parameter $\epsilon_I \in [0,1]$. The subscript b in ϕ_b stands for binary and is used for more clarity. For the MAP problem, we can write

$$\begin{aligned}
\Pr(\widetilde{h_I}(V) = h_I(v)|\mathbf{L}, \mathbf{P}_n) &=_a \Pr(\widetilde{h_I}(V) = h_I(v)|\mathbf{L}_I, \mathbf{P}_n^I) \\
&\leq_b \Pr(\widetilde{h_I}(V) = h_I(v)|\mathbf{L}_I^r, \mathbf{P}_n^I) \\
&=_c \Pr(\widetilde{h_I}(V) = h_I(v)|\mathbf{P}_n^{I,r}) \\
&=_d \frac{1}{2}\Pr[\mathbf{G}^I(1, 2 \text{ to } T(n)+1) = \mathbf{0}] + \frac{1}{2}
\end{aligned} \tag{38}$$

Where (a) is by lemma 3 and 4. (b) follows by the reduction from the noisy leakage given by λ_I to a random probing leakage given by ϕ_b. In (c), the leakage given by $\mathbf{L}_I^r$ is substituted in the system of equations described by $\mathbf{P}_n^I$. The resultant matrix is denoted by $\mathbf{P}_n^{I,r}$. Finally, (d) is resulted by application of relation (23), with $q = 2$ and $\mathbf{G}^I = \mathsf{Gaussian\text{-}Elim}(\mathbf{P}_n^{I,r})$.
By direct replacement, $\mathsf{SRP}_{h_I(V)}(n)$ will be

$$\mathsf{SRP}_{h_I(V)}(n) \leq \frac{1}{2}\Pr[\mathbf{G}^I(1, 2 \text{ to } T(n)+1) = \mathbf{0}]. \tag{39}$$

It is seen that $\mathsf{SRP}_{h_I(V)}(n)$ is a function of ϵ_I. According to our notations, for any ϵ_I that the reduction is possible, we have

$$\mathsf{SRP}_{h_I(V)}(n) \leq \mathsf{SRP}_{h_I(V)}(n, \epsilon_I).$$

The parameter ϵ_I for function ϕ_b, at given fixed I, is bounded by $\epsilon_{I,min}$ as $\epsilon_{I,min} \leq \epsilon_I$. By invoking relation (16), $\epsilon_{I,min}$ is calculated as

$$\epsilon_{I,min} = 1 - \int_{\beta \in \Omega(\zeta_I)} \min_{\alpha \in \{0,1\}} \mathbb{P}_{A|\zeta_I=\beta}(\alpha)\, d\beta \tag{40}$$

Since $\epsilon_{I,min}$ is the lowest possible value that the reduction still holds, we can write

$$\mathsf{SRP}_{h_I(V)}(n) \leq \mathsf{SRP}_{h_I(V)}(n, \epsilon_{I,min}).$$

Lemma 5. *For a given I, let $\delta_I = \mathsf{SD}(A; A|\lambda_I(A))$, with A being a uniform binary random variable, then we will have $\epsilon_{I,min} = 2\delta_I$*

Proof. In appendix D.

Back to the estimation of $\mathsf{SRP}_{h_I(V)}(n)$, an exciting observation remains; since we have $\mathbf{P}_n^I = \mathbf{P}_n$, matrix $\mathbf{P}_n^{I,r}$ is distributed identically to $\mathbf{P}_n^r$, provided their corresponding ϵ parameters are equal. As a result, the row reduced echelon form matrices $\mathbf{G}$ and $\mathbf{G}^I$ are distributed identically. Therefore, for $I \in [1, 2^k - 1]$, we can write

$$\begin{aligned}\mathsf{SRP}_{h_I(V)}(n, \epsilon_{I,min}) &= \frac{1}{2}\Pr[\mathbf{G}^I(1, 2 \text{ to } T(n)+1) = \mathbf{0}] \\ &= \frac{1}{2}\Pr[\mathbf{G}(1, 2 \text{ to } T(n)+1) = \mathbf{0}] = \frac{q}{2(q-1)}\mathsf{SRP}_V(n, \epsilon_{I,min})\end{aligned} \tag{41}$$

Corollary 1. *For a linear masked algorithm $\mathcal{A}'$, with secret variable V, if $\mathsf{SRP}_V(n, \epsilon)$ is a descending function of n, for $\epsilon \leq \epsilon_0$, then $\mathcal{A}'$ is secure with observation function ν, provided that $\epsilon_{I,min} \leq \epsilon_0$ for every $I \in [1, 2^k - 1]$.*

Note that to prove the security of $\mathcal{A}'$ against a noisy observation function ν, the reduction in $\mathbb{F}_q$ require $\mathsf{SRP}_V(n, \epsilon_{min})$ to be a decreasing function of n. However, the new approach proposed here requires that $\mathsf{SRP}_V(n, \max_I\{\epsilon_{I,min}\})$ to be a decreasing function of n. It is easy to show that $\max_I\{\epsilon_{I,min}\}$ and ϵ_{min} are related via $\max_I\{\epsilon_{I,min}\} \leq \epsilon_{min}$. In lemma 6, we will prove something similar.

The following examples will help to convey the concept better. For $\nu(V) = \mathsf{HW}(V)$, at $I = 2^k - 1$, we obtain $\epsilon_{I,min} = 1$. This means that with the hamming weight leakage, no linear masked circuit will be secure. The parameter $\epsilon_{2^k-1,min} = 1$ implies that a complete knowledge of XOR of bits of the secret is required in the reduction. Refer to the discussion around equation (6) to see that indeed this knowledge is divulged with the hamming weight leakage.

As another example, with $\nu(V) = \mathsf{ZV}^{\circ}(V)$, for every $I \in [1, 2^k - 1]$, we obtain $\epsilon_{I,min} = 1/2^{k-1}$, which is relatively a very small value, as we were intuitively expecting. Recall that for this leakage, we had $\epsilon_{min} = 1$. Therefore, with our new approach, at sufficiently big fields, chances are that a masked circuit to be secure against the ZV° leakage. Note that this leakage discriminates only a single element of $\mathbb{F}_q$. In a bigger field, more elements are present. So, the relative amount of information that the adversary obtains with this observation function decreases.

Relation of δ_I and δ. Until here, we have shown that $\epsilon_{I,min} = 2\delta_I$. Now, we want to focus on the relation of $\delta = \mathsf{SD}(V; V|\nu(V))$ and δ_I.

Lemma 6. *For a constant I, let $\delta_I = \mathsf{SD}(A; A|\lambda_I(A))$, with A being a uniform binary random variable, then we will have $\delta_I \leq \delta$.*

Proof. In appendix E.

It seems that the collection of δ_I values, specially $\max_I\{\delta_I\}$ is a good metrics for assessment of the noise of observation function ν than compared to δ and γ.

Remark 2. In section 2.2, we reviewed the work of [11]. As discussed there, their work is a particular case of our problem. For a single encoding, with no other computations on the shares, we have only one relation as $V = V_1 \oplus V_2 \oplus \ldots \oplus V_n$. Since in the random probing leakage all the shares are required to recover secret value V, it is not difficult to prove that $\mathsf{SRP}_V(n, \epsilon) = \epsilon^n$. This SRP is a monotonically decreasing function of n, provided that $\epsilon < 1$. Therefore, for a noise function ν, if we have $\delta < 1/2$, we can write

$$\forall I \in [1, 2^k - 1] \;\; \epsilon_{I,min} = 2\delta_I \leq 2\delta < 1 \;\; \Rightarrow \;\; \max_I\{\epsilon_{I,min}\} < 1$$

This is in agreement with the result of [11].

Remark 3. In section 2.5, we uttered that the SR-SNI refreshing algorithm, in [20], is shown to be secure for $\epsilon \leq 0.15$. Therefore, this refreshing is secure for $\nu = \mathsf{ZV}^\circ$, at $k \geq 4$. However, it is not secure with $\nu = \mathsf{HW}$.

3.3 Procedure for checking the security

Here, we briefly restate the required steps for assessing the security of a given linear masked algorithm $\mathcal{A}'$.

First, with tools described in section 2, obtain $\mathsf{SRP}_V(n, \epsilon)$. Next, for the given observation function ν, compute the ϵ_{min} value, which is given by relation (16). If $\mathsf{SRP}_V(n, \epsilon)$ is a decreasing function of n for this ϵ_{min}, then the job is over. If not, or if a more refined assessment is necessary, then $\epsilon_{I,min}$ values for $I \in [1, 2^k - 1]$ should be calculated.

Algorithm 2 Compution of $\epsilon_{\mathsf{I,min}}$ values

Input Field order $q = 2^k$ and the observation function $\nu(.)$
Output The maximum of the $\epsilon_{I,min}$ values

1: **for** $v = 0$ **to** $2^k - 1$ **do**
2: $\quad \mathbb{P}_{V,\zeta}(v, \beta \in \mathbb{R}) = \mathbf{0}$ ▷ Initialize to zero
3: $\quad$ **for** $i = 1$ **to** N **do** ▷ For a sufficiently big N
4: $\quad\quad$ **Evaluate** ν **as** $\beta \leftarrow \nu(v)$ ▷ ν might be probabilistic
5: $\quad\quad \mathbb{P}_{V,\zeta}(v, \beta) = \mathbb{P}_{V,\zeta}(v, \beta) + \frac{1}{N2^k}$ ▷ A Monte-Carlo approach
6: **for** $I = 1$ **to** $2^k - 1$ **do**
7: $\quad$ **Evaluate** $\mathbb{P}_{A,\zeta_I}$ **as** $\mathbb{P}_{A,\zeta_I}(\alpha, \beta) = \sum_{v \in \{0,1\}^k, \langle I, v\rangle = \alpha} \mathbb{P}_{V,\zeta}(v, \beta)$
8: $\quad$ **Compute** $\epsilon_{I,min} = 1 - \int_{\beta \in \Omega(\zeta_I)} \min_{\alpha \in \{0,1\}} \mathbb{P}_{A|\zeta_I = \beta}(\alpha)\; d\beta$
9: **return** $\max_I\{\epsilon_{I,min}\}$

In algorithm 2, we have put forward a systematic approach for calculating $\max_I\{\epsilon_{I,min}\}$. In this algorithm, for a given observation function ν, we have also proposed a simple Monte-Carlo-based method for approximation $\mathbb{P}_{V,\zeta}$.

The security of $\mathcal{A}'$ is dominated by $\mathsf{SRP}_V(n, \max_I\{\epsilon_{I,min}\})$. For instance, if $\mathsf{SRP}_V(n, \max_I\{\epsilon_{I,min}\})$ is a decreasing function of n, then $\mathcal{A}'$ is dubbed secure.

4 Conclusion

In this paper, we founded a systematic approach for evaluating the security of Boolean masked algorithms when an adversary is allowed to probe the internal variables of the algorithm through a noisy leakage function. In contrast to previous works, our approach gives concrete and optimum results for linear algorithms. Moreover, the proposed procedures can be utilized for assessment of arbitrary observation functions.

We also give a new metric for assessing the noise of leakage functions. As demonstrated here, this new metric is a good measure for the informativeness of the raw leakage data that an adversary can gather.

It would be an exciting challenge for future works to expand the developed methods to assess the security of a broader family of algorithms than the linear algorithms considered here.

References

1. Barthe, G., Belaïd, S., Dupressoir, F., Fouque, P.A., Grégoire, B., Strub, P.Y., Zucchini, R.: Strong non-interference and type-directed higher-order masking. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 116–129 (2016). https://doi.org/10.1145/2976749.2978427, https://app.dimensions.ai/details/publication/pub.1000875359 and https://hal.inria.fr/hal-01410216/file/main.pdf
2. Barthe, G., Dupressoir, F., Faust, S., Grégoire, B., Standaert, F.X., Strub, P.Y.: Parallel implementations of masking schemes and the bounded moment leakage model. In: Coron, J.S., Nielsen, J.B. (eds.) Advances in Cryptology – EUROCRYPT 2017. pp. 535–566. Springer International Publishing, Cham (2017)
3. Battistello, A., Coron, J.S., Prouff, E., Zeitoun, R.: Horizontal side-channel attacks and countermeasures on the isw masking scheme. In: Gierlichs, B., Poschmann, A.Y. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2016. pp. 23–39. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
4. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.J. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2004. pp. 16–29. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
5. Bruneau, N., Guilley, S., Heuser, A., Marion, D., Rioul, O.: Less is more. In: Güneysu, T., Handschuh, H. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2015. pp. 22–41. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
6. Burshtein, D., Miller, G.: Efficient maximum-likelihood decoding of ldpc codes over the binary erasure channel. IEEE Transactions on Information Theory **50**(11), 2837–2844 (2004). https://doi.org/10.1109/TIT.2004.836694
7. Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards sound approaches to counteract power-analysis attacks. In: Wiener, M. (ed.) Advances in Cryptology — CRYPTO' 99. pp. 398–412. Springer Berlin Heidelberg, Berlin, Heidelberg (1999)
8. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Kaliski, B.S., Koç, ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2002. pp. 13–28. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)

9. Duc, A., Dziembowski, S., Faust, S.: Unifying leakage models: From probing attacks to noisy leakage. In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology – EUROCRYPT 2014. pp. 423–440. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
10. Dziembowski, S., Faust, S., Skorski, M.: Noisy leakage revisited. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015. pp. 159–188. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
11. Dziembowski, S., Faust, S., Skórski, M.: Optimal amplification of noisy leakages. In: Kushilevitz, E., Malkin, T. (eds.) Theory of Cryptography. pp. 291–318. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
12. Evtyushkin, D., Riley, R., Abu-Ghazaleh, N.C., ECE, Ponomarev, D.: Branchscope: A new side-channel attack on directional branch predictor. p. 693–707. ASPLOS '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3173162.3173204, https://doi.org/10.1145/3173162.3173204
13. Faust, S., Rabin, T., Reyzin, L., Tromer, E., Vaikuntanathan, V.: Protecting circuits from leakage: the computationally-bounded and noisy cases. In: Gilbert, H. (ed.) Advances in Cryptology – EUROCRYPT 2010. pp. 135–156. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
14. Fedotov, A.A., Harremoes, P., Topsoe, F.: Refinements of pinsker's inequality. IEEE Transactions on Information Theory **49**(6), 1491–1498 (2003). https://doi.org/10.1109/TIT.2003.811927
15. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: Concrete results. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems — CHES 2001. pp. 251–261. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
16. Gohari, A., Gunlu, O., Kramer, G.: Coding for positive rate in the source model key agreement problem. IEEE Transactions on Information Theory **66**(10), 6303–6323 (Oct 2020). https://doi.org/10.1109/tit.2020.2990750, http://dx.doi.org/10.1109/TIT.2020.2990750
17. Guo, Q., Grosso, V., Standaert, F.: Modeling soft analytical side-channel attacks from a coding theory viewpoint. IACR Cryptol. ePrint Arch. **2018**, 498 (2018), https://eprint.iacr.org/2018/498
18. Heuser, A., Rioul, O., Guilley, S.: Good is not good enough. In: Batina, L., Robshaw, M. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2014. pp. 55–74. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
19. Ishai, Y., Sahai, A., Wagner, D.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003. pp. 463–481. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
20. Jahandideh, V.: Concrete evaluation of the random probing security (2020)
21. Javed, A.R., Beg, M.O., Asim, M., Baker, T., Al-Bayatti, A.H.: Alphalogger: Detecting motion-based side-channel attack using smartphone keystrokes. Journal of Ambient Intelligence and Humanized Computing pp. 1–14 (2020)
22. Kalai, Y.T., Reyzin, L.: A Survey of Leakage-Resilient Cryptography, pp. 727––794. Association for Computing Machinery, New York, NY, USA (2019), https://doi.org/10.1145/3335741.3335768
23. Katz, J., Lindell, Y.: Introduction to modern cryptography. CRC press (2020)
24. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) Advances in Cryptology — CRYPTO' 99. pp. 388–397. Springer Berlin Heidelberg, Berlin, Heidelberg (1999)

25. Kocher, P.C.: Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In: Koblitz, N. (ed.) Advances in Cryptology — CRYPTO '96. pp. 104–113. Springer Berlin Heidelberg, Berlin, Heidelberg (1996)
26. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security). Springer-Verlag, Berlin, Heidelberg (2007)
27. Mooij, J.M., Kappen, H.J.: Sufficient conditions for convergence of the sum–product algorithm. IEEE Transactions on Information Theory **53**(12), 4422–4437 (2007). https://doi.org/10.1109/TIT.2007.909166
28. Pearl, J.: Reverend bayes on inference engines: A distributed hierarchical approach. In: Proceedings of the Second AAAI Conference on Artificial Intelligence. p. 133–136. AAAI'82, AAAI Press (1982)
29. Prouff, E., Rivain, M.: Masking against side-channel attacks: A formal security proof. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7881, pp. 142–159. Springer (2013). https://doi.org/10.1007/978-3-642-38348-9_9, https://doi.org/10.1007/978-3-642-38348-9_9
30. Rivain, M., Prouff, E.: Provably secure higher-order masking of aes. In: Mangard, S., Standaert, F.X. (eds.) Cryptographic Hardware and Embedded Systems, CHES 2010. pp. 413–427. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
31. Standaert, F.X., Koeune, F., Schindler, W.: How to compare profiled side-channel attacks? In: Abdalla, M., Pointcheval, D., Fouque, P.A., Vergnaud, D. (eds.) Applied Cryptography and Network Security. pp. 485–498. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
32. Veyrat-Charvillon, N., Gérard, B., Standaert, F.X.: Soft analytical side-channel attacks. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology – ASIACRYPT 2014. pp. 282–296. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
33. Wei, L., Luo, B., Li, Y., Liu, Y., Xu, Q.: I know what you see: Power side-channel attack on convolutional neural network accelerators. In: Proceedings of the 34th Annual Computer Security Applications Conference. pp. 393–406 (2018)
34. Yarom, Y., Falkner, K.: Flush+reload: A high resolution, low noise, l3 cache side-channel attack. In: 23rd USENIX Security Symposium (USENIX Security 14). pp. 719–732. USENIX Association, San Diego, CA (Aug 2014), https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/yarom

Appendix A Proof of Lemma 1

At a fixed n, assume $\mathsf{SRP}_{h_I(V)}(n) = \mu_I$ for $I \in [1, 2^k - 1]$. Given values of $\{\mu_1, \mu_2, \ldots, \mu_{2^k-1}\}$, we want to study the behavior of $\mu = \mathsf{SRP}_V(n)$. In fact, we will show that although the knowledge of $\{\mu_1, \mu_2, \ldots, \mu_{2^k-1}\}$ cannot uniquely determine the value of μ, it is enough for bounding the maximum value that μ can take.

Let 2^k values $\{p_0, p_1, \ldots, p_{2^k-1}\}$ be the probability distribution of V given side information $\mathbf{P}_n$ and a random instance of leakage $\mathbf{L}$.

For the computation $\mathsf{SRP}_{h_I(V)}(n)$ of We have

$$\begin{aligned}
\mathsf{SRP}_{h_I(V)} &= \max_{\alpha\in\{0,1\}} \Pr(h_I(V)=\alpha|\mathbf{L},\mathbf{P}_n) - \frac{1}{2} \\
&= \max_{\alpha\in\{0,1\}} \Pr(\langle I,V\rangle=\alpha|\mathbf{L},\mathbf{P}_n) - \frac{1}{2} \\
&= \max_{\alpha\in\{0,1\}} \sum_{\langle I,v\rangle=\alpha} \Pr(V=v|\mathbf{L},\mathbf{P}_n) - \frac{1}{2} \qquad (42)\\
&= \max_{\alpha\in\{0,1\}} \sum_{\langle I,v\rangle=\alpha} p_v - \frac{1}{2} \\
&= \mu_I
\end{aligned}$$

We have omitted the fixed $v \in \{0,1\}^k$ statement beneath the $\sum$ expressions throughout this proof for simplicity.
Note that we can write

$$\sum_{\langle I,v\rangle=0} p_v + \sum_{\langle I,v\rangle=1} p_v = 1 \quad \Rightarrow \quad \sum_{\langle I,v\rangle=0} p_v - \frac{1}{2} = -(\sum_{\langle I,v\rangle=1} p_v - \frac{1}{2})$$

Therefore, we will have

$$\sum_{\langle I,v\rangle=0} p_v - \frac{1}{2} = \pm\mu_I \quad \Rightarrow \quad \sum_{\langle I,v\rangle=0} p_v = \frac{1}{2} \pm \mu_I$$

By collecting these relations for all values of $I \in [1, 2^k-1]$, we will have the following system of equations.

$$\begin{cases}
\sum_{\langle 0,v\rangle=0} p_v = 1 \\
\sum_{\langle 1,v\rangle=0} p_v = \frac{1}{2} \pm \mu_1 \\
\sum_{\langle 2,v\rangle=0} p_v = \frac{1}{2} \pm \mu_2 \\
\vdots \\
\sum_{\langle 2^k-1,v\rangle=0} p_v = \frac{1}{2} \pm \mu_{2^k-1}
\end{cases} \qquad (43)$$

The first equation in the system above is obtained differently. This equation is simply sum of all p_i values, which should be 1.

We multiply equations two to the end by 2 and subtract the first equation from them. The reason for this operation will be apparent soon. With matrix representation for the resultant system, we will have

$$\begin{bmatrix} \langle 0,0\rangle! & \langle 0,1\rangle! & \dots & \langle 0,2^k-1\rangle! \\ \langle 1,0\rangle! & \langle 1,1\rangle! & \dots & \langle 1,2^k-1\rangle! \\ \langle 2,0\rangle! & \langle 2,1\rangle! & \dots & \langle 2,2^k-1\rangle! \\ \vdots & \vdots & \ddots & \vdots \\ \langle 2^k-1,0\rangle! & \langle 2^k-1,1\rangle! & \dots & \langle 2^k-1,2^k-1\rangle! \end{bmatrix} . \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{2^k-1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \pm 2\mu_1 \\ \pm 2\mu_2 \\ \vdots \\ \pm 2\mu_{2^k-1} \end{bmatrix}$$

(44)

Where the symbol (!) is used to mean a simple function as $\langle i,j\rangle! = 2\langle i,j\rangle - 1$. For ease of notations, we represent the matrices participating in (44) with $\mathbf{M}$, $\mathbf{P}$, $\mathbf{C}_1$, and $\mathbf{C}_2$, respectively. So, we can write

$$\mathbf{M}.\mathbf{P} = \mathbf{C}_1 + \mathbf{C}_2.$$

With closer inspection of $\mathbf{M}$, it turns out that it is Hadamard matrix.

$$\mathbf{M} = \mathbf{H}_{2^k} = \begin{bmatrix} \mathbf{H}_{2^{k-1}} & \mathbf{H}_{2^{k-1}} \\ \mathbf{H}_{2^{k-1}} & 1 - \mathbf{H}_{2^{k-1}} \end{bmatrix} \tag{45}$$

With $\mathbf{H}_1 = [1]$. The determinant of Hadamard matrix is non-zero. Moreover, we have

$$\mathbf{M}^{-1} = \frac{1}{2^k}\mathbf{M}^\dagger.$$

For more results about the Hadamard matrix, please refer to its wiki page.
Since $det(\mathbf{M}) \neq 0$, we will have a unique solution for $\mathbf{P}$ corresponding to $\mathbf{C}_1$.

$$\mathbf{M}.\mathbf{P} = \mathbf{C}_1$$

We show this solution with $\mathbf{P}_1$. Finding $\mathbf{P}_1$ is not difficult. It can be readily verified that $\mathbf{P}_1$ is the uniform distribution. i.e., all the entries of $\mathbf{P}_1$ are $1/2^k$. It remains to study the structure of answers corresponding to $\mathbf{C}_2$, i.e., solutions to

$$\mathbf{M}.\mathbf{P} = \mathbf{C}_2$$

We show these solutions with $\mathbf{P}_2$. Note that $\mathbf{C}_2$ is not a single constant matrix. For characterizing $\mathbf{P}_2$, we can write

$$\mathbf{M}.\mathbf{P}_2 = \mathbf{C}_2 \quad \Rightarrow \quad \mathbf{P}_2 = \frac{1}{2^k}\mathbf{M}^\dagger\mathbf{C}_2. \tag{46}$$

Since $\mathbf{M}$'s elements are only 1 and -1, each entry of $\mathbf{P}_2$ will be bounded to

$$\frac{1}{2^k}\sum_{i=1}^{2^k-1} \pm 2\mu_i = \frac{1}{2^{k-1}}\sum_{i=1}^{2^k-1} \pm \mu_i \tag{47}$$

The worst-case condition will be

$$\frac{1}{2^{k-1}}\sum_{i=1}^{2^k-1}\pm\mu_i \le \frac{1}{2^{k-1}}\sum_{i=1}^{2^k-1}\mu_i \le 2\max_i \mu_i \tag{48}$$

By merging these solutions for $\mathbf{C}_1$ and $\mathbf{C}_2$, we can write

$$\forall i \in [0, 2^k-1]; \quad p_i \le \frac{1}{2^k} + 2\max_i \mu_i$$

Therefore, we will have

$$\mathsf{SRP}_V = \max_{v\in\{0,1\}^k} \Pr(V=v|\mathbf{L},\mathbf{P}_n) - \frac{1}{2^k} = \max_i p_i - \frac{1}{2^k} \le 2\max_i \mu_i \tag{49}$$

Based on our hypothesis, all μ_i values are decreasing by increasing n, So SRP_V will also be a decreasing function of n. Moreover, if they limit to zero, so will do SRP_V, and this completes our proof for this lemma.

Appendix B Proof of Lemma 2

For the inner product of k-bit integers I, X_1, and X_2, we can write

$$\langle I, (X_1 \oplus X_2)\rangle = \langle I, X_1\rangle \oplus \langle I, X_2\rangle. \tag{50}$$

For a binary value P_1, by just testing the two possible values of P_1, we can show

$$\langle I, (P_1 X_1)\rangle = P_1\langle I, X_1\rangle. \tag{51}$$

By iterative application of these rules, for binary vector $\{P_1, P_2, \ldots, P_t\}$ and k-bit variables $\{X_1, X_2, ..., X_t\}$, we can show that

$$\langle I, (P_1X_1 \oplus P_2X_2 \oplus \ldots \oplus P_tX_t)\rangle = P_1\langle I, X_1\rangle \oplus P_2\langle I, X_2\rangle \oplus \ldots \oplus P_t\langle I, X_t\rangle. \tag{52}$$

Matrix $\mathbf{P}_n$ is composed of $d(n)$ equations, each one with binary coefficients as in equation (52). Let $\mathbf{P}_n(1, 1 \text{ to } T(n)+1)$ be the coefficients of the first equation in $\mathbf{P}_n$. For this equation, we can write

$$\begin{aligned}&\langle I, \mathbf{P}_n(1, 1 \text{ to } T(n)+1) \times [V, X_1, X_2, ..., X_{T(n)}]^\dagger\rangle = \\ &\quad \mathbf{P}_n(1,1)\langle I, X_1\rangle \oplus \mathbf{P}_n(1,2)\langle I, X_2\rangle \oplus \ldots \oplus \mathbf{P}_n(1, T(n)+1)\langle I, X_{T(n)}\rangle = 0.\end{aligned} \tag{53}$$

Putting together all the $d(n)$ equations, we have

$$\langle I, \mathbf{P}_n \times [V, X_1, X_2, ..., X_{T(n)}]^\dagger\rangle = \mathbf{P}_n \times [\langle I, X_1\rangle, \langle I, X_2\rangle, \ldots, \langle I, X_{T(n)}\rangle] = \mathbf{0}. \tag{54}$$

This is what we wanted to show.

Appendix C Proof of Lemma 3 and 4

Assume X is a k-bit variable, with bits as $[X^1, \ldots, X^k]$. For a fixed I, there is at least one set of $k-1$ bits of X that collectively with $\langle I, X\rangle$ determines X uniquely. For each I, we fix a such $k-1$ bits and show it by IX.

So, we decompose each k-bit variable X as $[\langle I, X\rangle, IX]$. Next, with lemma 2, we can show that the parity-check relations over $\{\langle I, X_1\rangle, \langle I, X_2\rangle, \ldots, \langle I, X_{T(n)}\rangle\}$ is independent of constraints over $\{IX_1, IX_2, \ldots, IX_{T(n)}\}$. In fact, the two following systems are separate.

$$\begin{aligned} &\mathbf{P}_n \times [\langle I, X_1\rangle, \langle I, X_2\rangle, \ldots, \langle I, X_{T(n)}]^\dagger = \mathbf{0} \\ &\mathbf{P}_n \times [IX_1, IX_2, \ldots, IX_{T(n)}]^\dagger = \mathbf{0} \end{aligned} \tag{55}$$

Members of $\mathcal{S}_n$ are the Cartesian product of solution sets of the above two systems. We have already named the solution set of the first set by $\mathcal{S}_n^I$. Here, for ease of referencing, we denote the solution set of the second system with $\overline{\mathcal{S}}_n^I$. So, we have

$$\mathcal{S}_n = \mathcal{S}_n^I \times \overline{\mathcal{S}}_n^I \tag{56}$$

With the following interpretation. Each $(T(n)+1)$-length vector in $\mathcal{S}_n$ is composed of one $(T(n)+1)$-length vector from $\mathcal{S}_n^I$ merged with one $(T(n)+1)$-length vector from $\overline{\mathcal{S}}_n^I$.

Before starting the main proof we need to mention another point. Assume variable V is randomly selected from $\mathbb{F}_q$, such that it satisfies the constrain $\langle I, V\rangle = \alpha$ for some fixed I. The bit α is the value of a uniform binary random variable A. Also, assume $L \leftarrow \nu(V)$. We can write

$$\Pr(L=\beta) = \Pr(\zeta=\beta|\langle I, V\rangle = \alpha) = \frac{\sum_{v\in\{0,1\}^k, \langle I,v\rangle=\alpha} \mathbb{P}_{V,\zeta}(v,\beta)}{\sum_{v\in\{0,1\}^k, \langle I,v\rangle=\alpha} \mathbb{P}_V(v)} = \mathbb{P}_{\zeta_I|A=\alpha}(\beta) \tag{57}$$

Where the last equality follows from the definition given by relation (34). Now, if L_I is sampled based on the probability distribution $\mathbb{P}_{\zeta_I|A=a}$, we can write

$$\Pr(L=\beta) = \Pr(L_I=\beta).$$

As defined in the text, the function $\lambda_I(\alpha)$ at each invocation outputs a random value based on distribution $\mathbb{P}_{\zeta_I|A=a}$.

With these preambles, we are ready to turn back to the MAP problem. Starting with lemma 3, we can write

$$
\begin{aligned}
&arg \max_{\alpha\in\{0,1\}} \Pr(\langle I,V\rangle=\alpha|\mathbf{L},\mathcal{S}_n)\\
&\quad=_a arg \max_{\alpha\in\{0,1\}} \Pr(\mathbf{L}|\langle I,V\rangle=\alpha,\mathcal{S}_n)\Pr(\langle I,V\rangle=\alpha,\mathcal{S}_n)\\
&\quad=_b arg \max_{\alpha\in\{0,1\}} p(\alpha)\Pr(\mathbf{L}|\langle I,V\rangle=\alpha,\mathcal{S}_n)\\
&\quad=_c arg \max_{\alpha\in\{0,1\}} p(\alpha)\Pr(\mathbf{L}|\langle I,V\rangle=\alpha,\mathcal{S}_n^I\times\overline{\mathcal{S}}_n^I)\\
&\quad=_d arg \max_{\alpha\in\{0,1\}} p(\alpha)\sum_{\mathbf{s}^I\in\mathcal{S}_n^I,s_1^I=\alpha}\sum_{\overline{\mathbf{s}}^I\in\overline{\mathcal{S}}_n^I}\Pr(\mathbf{L}|[\mathbf{s}^I,\overline{\mathbf{s}}^I])\\
&\quad=_e arg \max_{\alpha\in\{0,1\}} p(\alpha)\sum_{\mathbf{s}^I\in\mathcal{S}_n^I,s_1^I=\alpha}\Pr(\mathbf{L}|\mathbf{s}^I)\\
&\quad=_f arg \max_{\alpha\in\{0,1\}} p(\alpha)\sum_{\mathbf{s}^I\in\mathcal{S}_n^I,s_1^I=\alpha}\prod_{i=1}^{T(n)+1}\Pr(\mathbf{L}_i|s_i^I)\\
&\quad=_g arg \max_{\alpha\in\{0,1\}} p(\alpha)\sum_{\mathbf{s}_n^I\in\mathcal{S}_n^I,s_1^I=\alpha}\prod_{i=1}^{T(n)+1}\Pr(\mathbf{L}_{I,i}|s_i^I)\\
&\quad=_h arg \max_{\alpha\in\{0,1\}} p(\alpha)\sum_{\mathbf{s}_n^I\in\mathcal{S}_n^I,s_1^I=\alpha}\Pr(\mathbf{L}_I|\mathbf{s}^I)\\
&\quad=_i arg \max_{\alpha\in\{0,1\}} p(\alpha)\Pr(\mathbf{L}_I|\langle I,V\rangle=\alpha,\mathcal{S}_n^I)\\
&\quad=_j arg \max_{\alpha\in\{0,1\}} \Pr(\langle I,V\rangle=\alpha|\mathbf{L}_I,\mathcal{S}_n^I)
\end{aligned}
\tag{58}
$$

Where (a) is by the Bayesian relation and ignoring the probabilities that are independent of α.
(b) is by noting that

$$\Pr(\langle I,V\rangle=\alpha,\mathcal{S}_n)=\Pr(\langle I,V\rangle=\alpha,\mathcal{S}_n^I\times\overline{\mathcal{S}}_n^I)=\Pr(\langle I,V\rangle=\alpha,\mathcal{S}_n^I)\triangleq p(\alpha)$$

Which follows by independence of set $\overline{\mathcal{S}}_n^I$ and values that $\langle I,V\rangle$ can take.
(c) is by replacing the solution set $\mathcal{S}_n$ with its decomposed format as in (56).
(d) is by expanding the conditional probability and omitting the size constants.
(e) also follows by the rule of conditional probability. Note that all possible cases of $\overline{\mathcal{S}}_n^I$ are summed up. At this point, proof suffices for what is conjectured in lemma 3.
(f) is by independent leakage assumption. i.e., each variable leaks independently. In this equation, the ith member of leakage vector is represented by $\mathbf{L}_i$.
In (g), random value of $\mathbf{L}_i$ is replaced by a new variable that has the same probability distribution. See discussion around relation (57) and note that how $\mathbf{L}_{I,i}$ is sampled based on the realized value. Refer to (35) for the definition of the sampling function.

(h) is by converting back the decomposed probability expression.
(i) is by definition of set $\mathcal{S}_n^I$.
Finally, (j) is by the Bayesian rule. This completes the proof of lemma 4.

Appendix D Proof of Lemma 5

With a binary uniform random variable A, at a fixed I, given probabilistic function λ_I, the random variable ζ_I is defined as $\zeta_I = \lambda_I(A)$. For the evaluation of δ_I, we can write

$$\begin{aligned}
\delta_I =& \mathsf{SD}(A; A|\lambda_I(A)) =_a \frac{1}{2}\int_{\beta\in\Omega(\zeta_I)} |\mathbb{P}_{A,\zeta_I}(0,\beta) - \frac{1}{2}\mathbb{P}_{\zeta_I}(\beta)|d\beta \;\; + \\
& \qquad\qquad \frac{1}{2}\int_{\beta\in\Omega(\zeta_I)} |\mathbb{P}_{A,\zeta_I}(1,\beta) - \frac{1}{2}\mathbb{P}_{\zeta_I}(\beta)|d\beta \\
=_b & \frac{1}{2}\int_{\beta\in\Omega(\zeta_I)} |\mathbb{P}_{A,\zeta_I}(1,\beta) - \mathbb{P}_{A,\zeta_I}(0,\beta)|d\beta \\
=_c & \frac{1}{2}\int_{\beta\in\Omega(\zeta_I)} [\max_{\alpha\in\{0,1\}} \mathbb{P}_{A,\zeta_I}(\alpha,\beta) - \min_{\alpha\in\{0,1\}} \mathbb{P}_{A,\zeta_I}(\alpha,\beta)]d\beta \\
=_d & \int_{\beta\in\Omega(\zeta_I)} [\frac{1}{2} - \min_{\alpha\in\{0,1\}} \mathbb{P}_{A,\zeta_I}(\alpha,\beta)]d\beta \\
=_e & \frac{1}{2} - \frac{1}{2}\int_{\beta\in\Omega(\zeta_I)} \min_{\alpha\in\{0,1\}} [\mathbb{P}_{\zeta_I|A=\alpha}(\beta)]d\beta =_f \frac{1}{2}\epsilon_{I,min}
\end{aligned} \tag{59}$$

Where (a) is by definition given in (10), and the fact that $\mathbb{P}_A(0) = \mathbb{P}_A(1) = 1/2$.
(b) is by substitution of the value of $\mathbb{P}_{\zeta_I}(\beta)$ as $\mathbb{P}_{\zeta_I}(\beta) = \mathbb{P}_{A,\zeta_I}(1,\beta) + \mathbb{P}_{A,\zeta_I}(0,\beta)$.
(c) is by definition of the absolute value.
(d) is obtained by following identity.

$$\int_{\beta\in\Omega(\zeta_I)} [\max_{\alpha\in\{0,1\}} \mathbb{P}_{A,\zeta_I}(\alpha,\beta) + \min_{\alpha\in\{0,1\}} \mathbb{P}_{A,\zeta_I}(\alpha,\beta)]d\beta = 1$$

(e) is by conditional probability as

$$\min_{\alpha\in\{0,1\}} \mathbb{P}_{A,\zeta_I}(\alpha,\beta) = \min_{\alpha\in\{0,1\}} [\mathbb{P}_A(\alpha)\mathbb{P}_{\zeta_I|A=\alpha}(\beta)] = \frac{1}{2}\min_{\alpha\in\{0,1\}} \mathbb{P}_{\zeta_I|A=\alpha}(\beta).$$

Finally, (f) is by definition of $\epsilon_{I,min}$.

Appendix E Proof of Lemma 6

Let random variables $V \in \mathbb{F}_q$ and $\zeta = \nu(V)$ be as in the text. Starting with the definition of δ, for a fixed $I \in [1, 2^k - 1]$, we can write

$$
\begin{aligned}
\delta = \mathsf{SD}(V;V|\zeta) &= \frac{1}{2}\sum_{v\in\mathbb{F}_q}\int_{\beta\in\Omega(\zeta)} |\mathbb{P}_{V,\zeta}(v,\beta) - \mathbb{P}_V(v)\mathbb{P}_\zeta(\beta)|\ d\beta \\
&=_a \frac{1}{2}\int_{\beta\in\Omega(\zeta)} \sum_{v\in\{0,1\}^k,\langle I,v\rangle=0} |\mathbb{P}_{V,\zeta}(v,\beta) - \mathbb{P}_V(v)\mathbb{P}_\zeta(\beta)|\ d\beta \\
&\quad + \frac{1}{2}\int_{\beta\in\Omega(\zeta)} \sum_{v\in\{0,1\}^k,\langle I,v\rangle=1} |\mathbb{P}_{V,\zeta}(v,\beta) - \mathbb{P}_V(v)\mathbb{P}_\zeta(\beta)|\ d\beta \\
&\geq_b \frac{1}{2}\int_{\beta\in\Omega(\zeta)} |\sum_{v\in\{0,1\}^k,\langle I,v\rangle=0} [\mathbb{P}_{V,\zeta}(v,\beta) - \mathbb{P}_V(v)\mathbb{P}_\zeta(\beta)]|\ d\beta \\
&\quad + \frac{1}{2}\int_{\beta\in\Omega(\zeta)} |\sum_{v\in\{0,1\}^k,\langle I,v\rangle=1} [\mathbb{P}_{V,\zeta}(v,\beta) - \mathbb{P}_V(v)\mathbb{P}_\zeta(\beta)]|\ d\beta \\
&=_c \frac{1}{2}\int_{\beta\in\Omega(\zeta_I)} |\mathbb{P}_{A,\zeta_I}(0,\beta) - \mathbb{P}_A(0)\mathbb{P}_{\zeta_I}(\beta)|\ d\beta \\
&\quad + \frac{1}{2}\int_{\beta\in\Omega(\zeta_I)} |\mathbb{P}_{A,\zeta_I}(1,\beta) - \mathbb{P}_A(1)\mathbb{P}_{\zeta_I}(\beta)|\ d\beta \\
&=_d \frac{1}{2}\sum_{\alpha\in\{0,1\}}\int_{\beta\in\Omega(\zeta_I)} |\mathbb{P}_{A,\zeta}(\alpha,\beta) - \mathbb{P}_A(\alpha)\mathbb{P}_{\zeta_I}(\beta)|\ d\beta =_e \delta_I
\end{aligned}
\tag{60}
$$

Where in (a) order of the integration and the summation are exchanged. Also, the summation over $v \in \{0,1\}^k$ is partitioned into two parts, namely $[v \in \{0,1\}^k, \langle I, v\rangle = 0]$ and $[v \in \{0,1\}^k, \langle I, v\rangle = 1]$. Note that, each variable $v \in \mathbb{F}_q$ has a unique k-bit representation.
For (b), we have changed order of summation and the absolute value calculation and used the so-called triangle inequality.
(c) is by definition given in (32) and relations proposed in (33).
(d) is by merging the two integrals by an auxiliary variable $\alpha \in \{0,1\}$.
(e) is by definition of δ_I in the lemma. At this point, the proof is completed.