Leakage Certification Made Simple

Aakash Chowdhury[1], Arnab Roy[4], Carlo Brunetta[2], and Elisabeth Oswald[1,3]

[1] University of Klagenfurt, Austria {aakash.chowdhury, elisabeth.oswald}@aau.at
[2] Independent Researcher, brunocarletta@gmail.com
[3] University of Birmingham, United Kingdom
[4] University of Innsbruck, Austria, arnab.roy@uibk.ac.at

Abstract. Side channel evaluations benefit from sound characterisations of adversarial leakage models, which are the determining factor for attack success. Two questions are of interest: can we define and estimate a quantity that captures the ideal adversary (who knows all the distributions that are involved in an attack), and can we define and estimate a quantity that captures a concrete adversary (represented by a given leakage model)?

Existing work has led to a proliferation of custom quantities to measure both types of adversaries, which can be data intensive to estimate in the ideal case, even for discrete side channels and especially when the number of dimensions in the side channel traces grows.

In this paper, we show how to define the mutual information between carefully chosen variables of interest and how to instantiate a recently suggested mutual information estimator for practical estimation. We apply our results to real-world data sets and are the first to provide a mutual information-based characterisation of ideal and concrete adversaries utilising up to 30 data points.

1 Introduction

Mutual information (MI) enables us to quantify the amount of information that we obtain about one random variable by observing another random variable. This is a useful concept in the context of side channels because it enables us to quantify how much information we get about a secret (key-dependent) device state by observing e.g. the device power consumption. As a consequence, the mutual information appears across various areas in side channel research, such as in proofs about the security of masking, e.g. Grosso et al. [17,12,33][5], in the context of side channel distinguishers, e.g. Heuser et al. [20], and in the context of reasoning about the quality of so-called leakage models, e.g. Durvaux et al. [13] — the latter applications are the focus of our work.

[5] Proofs show that the informativeness of the side channel decreases exponentially in the number of shares.

1.1 Evaluating Device Security via Leakage Certification

Leakage models are important ingredients in side channel attacks. Side channel attacks are highly configurable, but they always require the extraction of information of small portions of the secret key from some observed side channel traces following a divide-and-conquer principle. The extraction of key information from the observable side channel traces can be achieved with a wide range of statistical and machine learning tools, which use as inputs a (key-dependent) leakage model and the observed side channel traces. It is well known that the use of an accurate leakage model is necessary for optimal information extraction [8].

From an adversarial point of view, the best leakage model would evidently be equal to the distribution of the side channel that the device emits. We call an adversary *ideal* if they know this distribution. To understand the worst case security of a device, an evaluator, acting as a *concrete* adversary wishes to assess the strength of this ideal adversary. Thus, they wish to assess the amount of information they can extract with their (estimated) model where they may use the model as a predictor or classifier in an attack. In the context of physical side channels such as the power consumption, the EM emanation, or device timing characteristics, the exact distribution of the observable side channel is unknown—both adversaries and evaluators can only work with estimations.

State of the art. Side channel evaluations can take very different forms: "in-house" evaluations are often performed by software/hardware developers, and are based on a mix of leakage detection testing [16] and performing concrete attacks. Evaluations that are part of a formal certification scheme (e.g. FIPS 140-3 [21,22], and CC [9,38,39]) are typically structured and must follow scheme specific guidance. Informally speaking, an evaluation seeks to establish the "security level" of a device: in-house evaluations typically understand this by checking if a specific countermeasure has the desired effect; evaluations under certification schemes have complex rule books and application guidelines that define a "security level". Clearly, any evaluation seeks to produce evidence for how strongly a well resourced adversary can perform: we call any adversary that can be instantiated in practice a "concrete adversary". Any comprehensive evaluation will also try and understand how close such a concrete adversary is to the ideal adversary. Recently, Azouaoui et al. [2] advocated to do this in the context of "worst case" evaluation assumptions: they argue, that it is advantageous to give an evaluator as much control and information as possible in an evaluation, an idea that has been picked up in the latest guidelines by the German certifier BSI [7]. These guidelines provide best practices for evaluations under the Common Criteria umbrella; in their latest version, the guidelines also include MI estimation (for leakage quantification) based on a new approach by Gao et al. [14].

Leakage certification. An evaluator seeks to understand how good their (estimated) leakage model (representing a concrete adversary) is (both in comparison with other models and in relation to the ideal adversary), which is a task

that was described by Durvaux et al. [13] as *leakage certification*, drawing on the earlier work of Renauld et al. [36].

Essentially, leakage certification is about assessing the gap between the amount of information that can be extracted by the ideal adversary and the amount of information that can be extracted by a concrete adversary. Durvaux et al. [13] capture the strength of the ideal adversary via the mutual information (between a key-dependent intermediate value, and the observed leakage) and they put forward a notion for the concrete adversary called the Perceived Information (PI), i.e. a quantity initially introduced in Renauld et al. [36], which relates a concrete leakage model with the observed leakage. Renauld et al.'s approach [36] was qualitative and proposed a statistical test to check whether the model of the concrete adversary can be distinguished from the model of the ideal adversary, i.e. the PI significantly differs from the MI. Bronchain et al.'s follow-up [6] made a first attempt to make certification quantitative and to estimate and bound the gap between the MI and the PI. For this purpose, they suggest working with the empirical distribution of the observed trace data, and they showed that the empirical PI (ePI) is a lower bound of the MI and that the empirical Hypothetical Information (eHI), defined as the amount of information that would be extracted from a hypothetical device exactly following the empirical model distribution, is an upper bound of the MI. Unfortunately, the convergence of the eHI and ePI metrics was shown experimentally to be (extremely) slow, which was then formally confirmed/analysed by Masure et al. [28]. This last work further showed that the hypothetical information cannot be unbiased if not working with the empirical model and therefore focused on the restricted goal of upper bounding the information that can be extracted by a concrete adversary (i.e., the PI) with the notion of Training Information (TI). Cutting to the chase, the state of the art leaves us with limited tools to evaluate the ideal adversary quantitatively. Two problems of practical interest remain.

Problem 1. Physical side channels are typically not univariate. The estimation of the HI and PI becomes completely inefficient for multivariate traces, and some of the estimators suggested in the past behave badly: Masure et al. [28] show that the gHI (which is a specific estimator of the HI) is not guaranteed to be an upper bound for the PI when the PI is estimated via the gPI, and that the eHI suffers from very slow convergence especially in multivariate settings. We remark at this point, that non-parametric estimators such as the eHI and ePI become computationally infeasible as the number of dimensions increases.

Problem 2. Physical side channels are typically not discrete. Even though physical side channels such as power and EM are measured by *digital* oscilloscopes that use an analogue to digital converter, modern digital oscilloscopes offer sophisticated signal amplification and de-noising settings which produce continuous outputs: assuming that devices are only used in their most basic setting underestimates real-world adversaries. Secondly, implementations that use masking countermeasures are often analysed after further software processing, including

filtering, and mean-free product-combining [34], which again create continuous outputs.

The resulting disconnection between practice (working with continuous side channel traces) and theory (only considering MI, PI, HI, TI, LI for discrete traces) was already mentioned in literature [6,28] in the context of characterising the ideal adversary.

1.2 Contributions and Outline

Our main contribution is an approach, utilising the estimator by Gao et al. [14], for both tasks in leakage certification: we wish to highlight that our approach is the first to effectively deal with quantifying a multi-variate ideal adversary. Our approach is inspired by an alternative representation of the PI, that we develop in this paper. Doing so enables us to give novel definitions for model quality, considering both cases where models are used as classifiers and where models are used as predictors. We link the case where models are used as classifiers to the machine learning concept of conditional cross entropy and provide an MI based definition for model quality and comparison, therefore enabling the assessment of the ideal adversary as well as concrete adversaries. In the case where models are used as predictors we prove that in many practical cases, for a given intermediate step in a cryptographic algorithm, we can compute the MI that characterises the ideal adversary without the need to explicitly have access to the device leakage function. We provide our fast implementations of the Gao estimator, as well as scripts to replicate experiments via a public repository: `https://github.com/sca-research/Leakage-Certification-Made-Simple`.

Implications for practice.

Alternatives to full attacks. We explained before that current evaluations often require to perform complete attacks, i.e. a full instantiation of a concrete adversary. Studies exploring the distribution of attack outcomes (via the key rank, e.g. [26]) show that outcomes of identically configured attacks can have a huge variance, in other words, there can be both "lucky" and "unlucky" adversaries. To produce a statistically robust quantification of how well a concrete adversary performs, a large number of repetitions (at least 100, see [26]) of the same attack must be performed. This is entirely infeasible in practice where already a single attack can be extremely expensive because of the presence of well designed countermeasures. Consequently, alternative approaches to evaluating the success of full attacks with the aid of information theoretic quantities are valuable and previous work already established a sound link between the mutual information characterising the ideal adversary and the success rate [8].

Enabling a more informed profiling step. Concrete adversaries use leakage models, which must be estimated from side channel traces. The quality of leakage models needs to be assessed: of course, this can be done by performing a full

attack, which leads to the problem of requiring a large number of repetitions. Alternative quantities are thus useful in practice and the approach of leakage certification of models does just that. In addition to evaluating the quality of an estimated model, previous work also showed that observing the convergence properties of MI quantities expressing model quality can guide evaluators with regard to the question of how many traces should be used to estimate a good model.

Limitations. Whilst our contribution represents a meaningful development in the area of leakage certification, there remain challenges to be solved. For instance, our best implementation can currently work with multivariate data of up to 30 dimensions. This is often sufficient to reason about the model relating to a single intermediate value (in both software and hardware implementations), however, it is clear that a higher dimensionality would be desirable. An aspect of practical importance is that of imperfect measurement setups: our approach does not help to spot them or rectify them. Finally, whilst leakage models are a key factor for attack success, their quality interplays with that of the intermediate value that they relate to: our contribution does not address this interplay.

Outline. After introducing notation, reviewing the side channel setting, recapping mutual information estimation, and reviewing the state of the art in leakage certification in Sect. 2, we spell out our framework for mutual information based leakage certification, in Sect. 3, by providing the relevant definitions alongside practical considerations. We demonstrate the efficiency of our framework via one set of simulations, and real-world datasets in Sect. 4 and Sect. 5.

2 Preliminaries

Following convention, we represent random variables with upper case letters, and their realisations with the corresponding lower case letters and sets are denoted with calligraphic typefaces. For two functions g and h, $g \circ h$ denotes the composition of the functions.

We denote the probability density function (pdf) and cumulative distribution function (cdf) of a continuous random variable with f and F respectively. For a discrete random variable, p denotes its probability mass function (pmf); for a continuous random variable, P denotes its probability density function. Whenever necessary, in a pdf, cdf or pmf we will make the corresponding random variable explicit in the subscript (e.g. f_X or F_X). In particular $p_{(X,Y)}$ refers to the joint distribution (pmf in this case) of the variables X and Y.

For any random variable X, $\mathbb{E}(X)$ and resp. $\mathbb{E}_X$ denote the expectation of X. The conditional distribution of X given Y is $X|Y$. For simplicity, we denote the conditional expectation of random variable $X|Y = y$ by $\mathbb{E}_{X|y}$ and its distribution by $P_{X|Y}$. We refer to an estimated quantity by using the sample size n in the subscript, e.g. I_n refers to a mutual information estimate obtained from a sample with size n, $f_{X,n}$ or $p_{X,n}$ denote the estimated pdf or pmf corresponding to a random variable X using n samples.

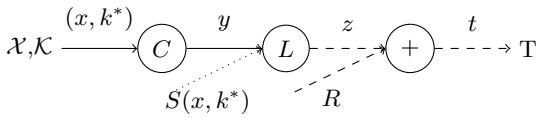


Fig. 1: Relationships between variables for the side channel scenario. The dashed lines indicate the random processes and variables, and the dotted line visualises that L might depend on some input and key-dependent randomness S.

The indicator function for a realisation x of X, is denoted as $\mathbb{I}_{X=x}$. We use $\mathcal{N}(\mu, \sigma)$ to denote the Gaussian distribution with mean μ and variance σ^2. We use $\mathcal{L}(0, \sigma)$ to denote a Laplacian distribution with mean 0 and variance $2\sigma^2$. For any d-dimensional vector $(x_1, \ldots, x_d) \in \mathbb{R}^d$ the ℓ_∞ or max norm is defined as $\max\{|x_i| : i = 1, \ldots, d\}$.

When working with functions, we overload notation and use the same variable for both the function, as well as the result of the function, and we may adapt the inputs to the context, e.g. $L(X, K)$ is a function, we also interpret L as a random variable, i.e. l is the realisation of L with some concrete inputs x, k.

2.1 The Side Channel Setting

In the side channel setting, we work with random variables that represent inputs/intermediates/outputs of cryptographic processes and leakage observations: we use $x \in \mathcal{X}$ for the input, which is mapped according to the cryptographic process via the application of some (cryptographic) target function(s) C and an (unknown) key $k^* \in \mathcal{K}$ to an intermediate $y \in \mathcal{Y}$. Implementations process cryptographic keys in "chunks", thus K and X have small support. The intermediate value is then mapped via a (noisy) device leakage function to the observable side channel trace t, see Fig. 1. A side channel trace t is a vector of leakage points. Each point corresponds to the physical processes that happen inside the device. Some of the physical processes depend on the input and key and we capture their contribution to the observable traces with the leakage function L and dependent noise S. Other processes are independent of the input and the key and we capture them via the independent noise variable R.

An adversary can observe (sometimes control) the inputs x, she knows the cryptographic function C and she observes traces t. An evaluator has the same knowledge plus the knowledge (and control) of the secret key k^*. Both do not know the device leakage function L, and thus use a so-called leakage model M.

Leakage functions. The leakage function L for a specific step in the execution of an algorithm can be simple. For instance, it can be determined by the number of bits changing within a register, or on a bus, in the case of a memory instruction, in which case it can be understood as a deterministic function. In other words, for a given input x and a fixed key k^*, it will always produce the same value

$L(x, k^*)$, and the distribution of $L(x, k^*)$ is completely determined by the current state (x, k^*).

However, the leakage function for a specific step in the execution of an algorithm can also be complex. For instance, in dedicated hardware, the power consumption depends on a complex interaction between many gates, which can result in data dependent glitches, cross-talk, etc. In this case, for a given input x and a fixed key k^*, we may see different values $L(x, k^*)$ upon repeat execution. The distribution of L thus depends on x, k^*, and some unknown randomness $S(x, k^*)$ that depends on x, k^*, but not on the independent noise R. Such a leakage function is probabilistic in (x, k^*). We provide Fig. 1 as a visual aid to understand the relationship between the variables, based on the functions that act on them.

We wish to emphasise the need to capture *all* types of leakage functions in the context of leakage certification because an evaluator does not know the leakage function(s) that a device exhibits and thus needs a methodology that always returns correct results.

For the rest of this paper, we have that T *should always be understood as a continuous multivariate variable (or a mixture with a continuous component)* to allow for the greatest flexibility. Whenever estimators require discrete inputs, we make this explicit by writing $[T]$ to indicate that discretisation of T must take place. Whenever the probabilistic nature of the leakage is not relevant, i.e. a statement holds irrespective of S and thus irrespective of whether L is deterministic or probabilistic, we drop S in the text for readability.

Leakage models. A leakage model is a function M that maps x, k under a target function C to $\mathbb{R}^d$. A model can be assumed based on device knowledge, or it can be estimated from real trace data. For example, a very popular standard leakage model is the Hamming Weight function, i.e. $M(x, k, C) = HW(C(x, k))$. In many practical cases, the leakage model is not known and must be estimated from the available trace data, typically by isolating some "points of interest" in each trace which are then used for model building (via statistical or machine learning methods).

Models can be used as predictors or as classifiers. In a predictive use, the adversary applies the model to the intermediate value (i.e. they compute $M(Y)$ and then uses a comparison based distinguisher [40]. Another use case for predictive models is in the context of leakage simulators [30]. When used as a classifier, the adversary applies the model to a new side channel observation to obtain a posterior distribution for the intermediate value, i.e. $\hat{Y} = (Y|T)_M$.

2.2 Measuring Dependency

The mutual information (MI) quantifies what we can learn about a variable X upon observing another variable Y. In other words, it quantifies a relationship between the distribution X and the distribution of $X|Y$. In the context of evaluating side channel security, we can use the MI to quantify how much we can

learn about a secret (key-dependent) device state upon observing the device's side channel (possibly by using a model; we call the adversary who knows the device's leakage characteristics the *ideal adversary*.

For general random variables X, Y (with marginal distributions P_X, P_Y and joint distribution P_{XY}), the MI is defined via the Radon-Nikodym derivative [41]:

$$I(X;Y) = \int_{\mathcal{X}\times\mathcal{Y}} \log \frac{dP_{XY}}{dP_X P_Y} dP_{XY}$$

The definition via the Radon-Nikodym derivative links the mutual information also with the Kullback-Leibler divergence D_{KL} as it can be expressed as $I(X;Y) = D_{KL}(dP_{XY}||dP_X dP_Y)$, see [41].

If either both variables are discrete, or both variables are continuous, then the MI can be expressed via the marginal and joint or conditional entropies[6], leading to the well known "2H" and "3H" expressions (owing to how many entropies are in the formulae) for MI, see Eq. (1).

$$I(X;Y) = H(X) - H(X|Y) \tag{1}$$
$$= H(X) + H(Y) - H(X,Y) \tag{2}$$

If one variable is discrete and one is continuous, or if one variable is a mixture, then the conditional density in the 2H formula, and the joint density in the 3H formula, may not be well defined unless the involved conditional distributions satisfy specific conditions [7], see [31]. Consequently, in situations where the distributions are unknown, and thus one cannot verify that the conditional/joint entropies are well defined, the conservative choice is to utilise an estimator that estimates the mutual information via the Radon-Nikodym derivative.

Estimating Mutual Information. The crucial property of any MI estimator is how well it "approximates" the true MI. This property is called the convergence of the estimator, and it describes the behaviour of the estimator when we supply it with increasing amounts of data. The weakest notion is convergence in probability A stronger notion is convergence in mean-square. Despite converging, an estimator can be biased. Bias in an estimator refers to the possibility that the estimator's expected value remains different from the true quantity being estimated. There are different approaches to estimating the MI (non-parametrically). One can either estimate the entropies in the 2H/3H formulas, or one can estimate the Radon-Nikodym derivative; we review them starting from the oldest techniques and leading up to the most recent advancements.

[6] We remind the reader that $H(X) = \mathbb{E}_X[-\log f(X)]$ if X continuous, and $H(X) = \mathbb{E}_X[-\log p(X)]$, if X is discrete; the definitions are extended naturally for conditional and joint distributions

[7] Observe that in such cases, we have a term that corresponds to a discrete entropy which is always positive, and a term that corresponds to a differential entropy which can be negative. Furthermore, the conditional distribution in the 2H formulae might not exist.

Entropy based MI estimation. Density based estimators directly estimate the densities in the 2H/3H formulae, whereas the k-NN based estimators estimate the distribution of the k-NN distance as a proxy for the density itself [25]. The previously mentioned limitations of 2H/3H estimators (i.e. both variables must either be discrete or continuous) initially applied to both approaches. A complementary approach based on using deep learning was published by Belghazi et al. [5] in 2018. It was suggested to be used for side channel tasks in Christiani et al. [10] but shortly thereafter McAllester and Stratos [29] highlighted problems with the approach. Antos and Kontoyiannis proved convergence for the plug-in estimator [1].

Integral estimators are well examined in the wider statistical literature, we refer to Györfi and van Meulen [18], Hall and Morton [19] (for the specific case of histogram density estimators), as seminal papers showing convergence results for low dimensions. No such guarantees can be given for higher dimensions because either the efficiency drops significantly or (to the best of our knowledge) no proof has been found.

In the side channel literature, based on the simplifying assumption of having discrete traces, the study of Batina et al. [3] uses an integral estimate [4]. Later, the notions of HI and eHI were developed as a means to bound the MI by Bronchain et al. [6], see Eq. (3 and 4) (the HI is thus a quantity to assess the ideal adversary).

$$\mathrm{HI}(X;[T];[M]) = \mathrm{H}(X) + \sum_{x\in\mathcal{X}} p_X(x) \sum_{t\in[\mathcal{T}]} p_{(X,[M])}(t|x) \log_2 p_{(X,[M])}(x|t) \tag{3}$$

$$\mathrm{eHI}_n(X;[T]) = \mathrm{H}(X) + \sum_{x\in\mathcal{X}} p_X(x) \sum_{t\in[\mathcal{T}]} \tilde{e}_n(t|[x]) \log_2 \tilde{e}_n(x|[t]) \tag{4}$$

The HI defines a quantity that measures the relationship between a variable X (which in [6] is set to be either the key variable, K or the intermediate $Y = C(X,K)$), and the observed (discrete or discretised) traces $[T]$ under a given model density $[M]$. It holds that $\mathrm{HI}(X;[T];[M]) = I(X;[M])$ and if $[M] = [T]$ then HI is equal to $I(X;T)$. The empirical HI (eHI) uses the empirical distribution $\tilde{e}_n(x,[t])$ in place of the model M, which can be estimated from the observed traces $[T]$. Bronchain et al. [6] show that with some assumptions the eHI converges in probability to the MI, thereby rediscovering the result by Antos and Kontoyiannis [1].

Nearest neighbour estimator for MI. Motivated by the need for a non-parametric MI estimator that applies even to high-dimensional/multivariate problems, Krasov et al. [24] introduced the idea of using a k-nearest neighbour (short k-NN) based estimator.

A recent contribution by Gao et al. [14] made a further significant step by estimating the Radon-Nikodym derivative requiring only **local** joint densities: their estimator does no longer require the existence of a joint density for the entire probability space. Their estimator essentially deals with two cases that

can occur for the joint distribution: either the sample (x, y) is discrete (this can be detected by checking the k-NN distance), then one can use the plug-in estimator for the Radon-Nikodym derivative; or the sample (x, y) is locally continuous, in which case they estimate the Radon-Nikodym derivative based on Eq. (5). They furthermore show that if either x or y are mixed, then the continuous case applies. Consequently, their estimator can deal with any form of mixture. The GKOV estimator is defined as given in Eq. (5).

$$I_n(X;Y) = \frac{1}{n}\sum_{i=1}^{n} \hat{I}_i = \log n + \frac{1}{n}\sum_{i=1}^{n}(\psi(\tilde{k}_i) - \log(n_{x,i}+1) - \log(n_{y,i}+1)) \quad (5)$$

where $\psi(u)$ is the digamma function $\psi(u) = \frac{d}{du}\log\Gamma(u) \approx \log u - \frac{1}{2u}$. The details of how to compute the quantities $n_{x,i}, n_{y,i}$ and $\tilde{k}_i$ can be found in Algorithm 1 in Appendix A.

With a suitable choice of k_n the GKOV estimator has the same convergence rate in the univariate setup as existing pmf/pdf based mutual information estimators, it provides strong convergence (convergence in mean-square, asymptotic unbiasedness) in all settings, and it can be generalised to multivariate variables. We refer to Appendix A for a discussion about choosing, k_n, as well as for the precise results of Gao et al. relating to convergence and bias.

2.3 Leakage Certification using HI and PI

We already defined the HI in Eq.(3). The PI, in Eq. (6)), was first introduced in Renauld et al. [36] as a measure for the quality of a leakage model.

$$\mathrm{PI}(Y;[T];[M]) = H(Y) + \sum_{y\in\mathcal{Y}} p_Y(y) \cdot \sum_{t\in[\mathcal{T}]} p_{(Y,[T])}(t|y)\log_2 p_{(Y,[M])}(y|t) \quad (6)$$

In the latest work, the process of comparing MI and PI, has been formalised via the concept of the *Regret* for a model M in Masure et al. [28, Definition 4]:

$$\mathsf{R}(M) = I(Y;T) - \mathrm{PI}(Y;T;M). \quad (7)$$

With the regret, the two tasks in leakage certification can be formalised as follows.

Definition 1 (PI based model quality). *Given a discrete model M and discrete traces T, we define the quality of the model as the regret $\mathsf{R}(M)$.*

Via a natural extension, we can also compare two different leakage models using the regret.

Definition 2 (PI based model comparison.). *Given two (discrete) leakage models $[M]_1$ and $[M]_2$, we say that $[M]_1$ is a better leakage model than $[M]_2$ for a (discrete) trace distribution $[T]$ if*

$$\mathsf{R}(M_1) < \mathsf{R}(M_2).$$

In practice, the process of leakage certification proceeds in the following way. The evaluator estimates $I(Y;T)$ via an estimator for the HI, which is taken to be an upper bound of the ideal adversary. The evaluator also estimates the $\mathrm{PI}(Y;T;M)$, which represents a concrete adversary and is thus taken to be a lower bound for the ideal adversary. We remark at this point that existing eHI and ePI require discrete data, whereas often side channel traces are representative of (multivariate) continuous variables. We offer a deeper discussion of the adverse impact of discretisation in Appendix B.

Considering the worst-case adversary. Azouaoui et al. [2] advocate that in real world evaluations, the evaluator is given as much power as technically feasible, e.g. control over device internal randomness, control over keys, knowledge of implementation details, etcetera. They argue that such an approach is likely to come close to an "optimal attack", and consequently they call the corresponding adversary the "worst case adversary".

The quantities to assess both the ideal adversary as well as any concrete adversary need to be estimated by an evaluator. In the spirit of Azouaoui et al., an evaluator may strive to do this in a "worst case" context. In our concrete experiments later on in this paper, we do utilise worst-case assumptions in the context of software implementations when defining a concrete adversary based on Gaussian templates (we assume access to randomness during Gaussian template building); we do not make this assumption for adversaries based on deep nets. We do not utilise any worst-case assumptions in the context of the hardware implementation.

3 Towards Simple Leakage Certification

3.1 PI and Regret Revisited

The PI is a quantity that has no correspondence to any known quantity in the wider machine learning community. We next develop an alternative representation of the PI, that will aid and motivate our simpler leakage certification approach.

Lemma 1. *The PI between the three variables $Y, [M], [T]$, and all distributions defined for all $y \in \mathcal{Y}$, can be written as:*

$$PI(Y;[T];[M]) = I(Y;[T]) - \mathbb{E}[D_{KL}((Y|[t],[T])||(Y|[t],[M]))]$$

Proof. To get the above equation, we use the substitution $p_{(Y,[M])}(y|t) = p_{(Y,[M])}(y|t)\frac{p_{(Y,[T])}(y|t)}{p_{(Y,[T])}(y|t)}$ in the expression of $PI(Y;[T];[M])$. The detailed derivation is provided in Appendix C. □

With this result, it is much clearer that for the PI to be well defined, we need that if $p_{(Y,[M])}(y|t) = 0$ then also $p_{(Y,[T])}(y|t) = 0$ otherwise we have $p_{(Y,[T])}(y|t)\log_2 0$, which is not well defined. This case may occur for models

that are bad representations of the unknown leakage, implying that the PI is not ideally suited to deal with models that are a poor approximation.

The above lemma makes apparent that the PI is indeed a quantity that is smaller or equal to $I(Y;[T])$, because the expected value of the KL divergence is positive. If $M = L$ then the expected value of the KL divergence is 0, and thus the PI equals to $I(Y;[T])$. If $M \neq L$ then the KL divergence is larger than zero and thus the PI measures, intuitively speaking, the amount of information lost on average if a specific model M is used.

Let us consider this alternative definition jointly with the PI based model quality Definition 1. This means the evaluator judges the model quality via the regret, i.e. they compute (with suitable estimators)

$$\begin{aligned}\mathsf{R}([M]) &= I(Y;[T]) - PI(Y;[T];[M]) \\ &= I(Y;[T]) - I(Y;[T]) + \mathbb{E}[D_{KL}((Y|[t],[T])||(Y|[t],[M]))] \\ &= \mathbb{E}[D_{KL}((Y|[t],[T])||(Y|[t],[M]))]. \end{aligned} \tag{8}$$

Equation 8 shows that the PI-based model quality only depends on the average KL divergence between the joint distributions $(Y|t,[M])$ and $(Y|t,[T])$. This motivates another definition for the regret, which is based on mutual information quantities that relate the intermediate and the resp. conditional probability.

3.2 MI based Model Quality (classifiers)

We mentioned before that leakage models can be used as predictors as well as classifiers. The PI-based model quality definition relates to the use case of classifiers: a model M is used to derive the posterior distribution $Y|T$; Masure et al. [28] make this explicit by the term "discriminative model". This fits well with the use case of deep learning models.

Reasoning about the quality of $Y|T$ using a model M is in fact a common machine learning problem, and it links with the concept of conditional cross-entropy, which in turn directly links to the mutual information, see McAllester and Stratos [29], by

$$\begin{aligned} I(T;Y) &= H(Y) - \inf_{P_{Y|T_M}} H(P_{Y|T}, P_{Y|T_M}) \\ \implies I(T;Y) &\geq H(Y) - H(P_{Y|T}, P_{Y|T_M}) \quad \forall P_{Y|T_M} \end{aligned} \tag{9}$$

If and only if $M = T$ then this equality can actually be achieved, otherwise the $I(T;Y)$ is an upper bound to the right hand side of Eq. 9.

Therefore, we suggest that the ideal adversary in the case where the model is used as a classifier should be defined as the left side of Eq. 9, and the quality of a given model should be defined via the right side of Eq. 9[8].

[8] We leave it as an open problem to qualitatively compare this notion with the notion of regret from previous work.

Definition 3 (MI-based classification-model quality). *Given any model M and traces T, we define the quality of the model as the difference:*

$$\delta(T, M) = I(T; Y) - \left(H(Y) - H(P_{Y|T}, P_{Y|T_M})\right).$$

Via a natural extension, we can also compare two different leakage models.

Definition 4 (MI-based classification-model comparison.). *Given two leakage models $[M]_1$ and $[M]_2$, we say that $[M]_1$ is a better leakage model than $[M]_2$ for a (discrete) trace distribution $[T]$ if*

$$\delta(T, M_1) < \delta(T, M_2).$$

These two definitions thus enable us to characterise concrete adversaries.

3.3 MI based Model Quality (predictors)

As explained before, leakage models can also be used as predictors, thus are used via application to Y as $M(Y)$. The best possible model would evidently be the case where $M = T$, which implies $M = L$ (because $T = L + R$, and R is independent of L).

We characterise the notion of the "best possible leakage model" via the *best MI*, referred to as I^b:

$$I^b = I(L(Y); T). \tag{10}$$

The challenge is that the evaluator does not know L, and that for different C, there will be different L. As a consequence, we wish to be able to estimate this quantity efficiently for many points (possibly jointly) in side channel traces.

An evaluator can however estimate the mutual information between the input and key and the observable traces, and we call this quantity I^{k}:

$$I^{\mathsf{k}} = I((X, K); T). \tag{11}$$

The connection between I^b and I^{k} is via the unknown leakage function L and the cryptographic target function C, which maps the key and input value to an intermediate value $Y = C(X, K)$. Using the data processing inequality [41], we know that $I^{\mathsf{k}} \leq I^b$ because the variables in Fig. 1 form a Markov chain.

From the data processing inequality, we can also infer that equality holds if $L \circ C$ is one-to-one, which we cannot expect to hold in practice. However, the data processing inequality is a very crude tool to reason about these two quantities, and we later show that equality holds under much more realistic conditions.

With this in mind, we define the MI-based quality of a predictive model.

Definition 5 (MI-based predictive-model quality). *Given any model M and traces T, we define the quality of the model as the difference:*

$$\Delta(T, M) = I^b - I(M(Y); T).$$

Definition 6 (MI-based predictive-model comparison). *Given any two models M_1 and M_2 and traces T, we say that M_1 is better than M_2 if*

$$\Delta(T, M_1) < \Delta(T, M_2).$$

Proof that $I^b = I^{\mathsf{k}}$ in many realistic scenarios We prove that $I^b = I^{\mathsf{k}}$ for a fixed cryptographic target C under some mild conditions, which we can expect to hold in many practical settings. This equality implies that in many practical cases, I^b can be obtained via estimating I^{k} and thus without the need to know or even estimate L. The full proof of this result can be found in Appendix D, and for the sake of readability, we only provide the proof outline in the following.

Based on the characteristics of the leakage functions (explained in Sect. 2.1), three cases must be considered in the proof:

- L is discrete and deterministically depends on the realisations of X and K. which means, $T = L \circ C(X, K) + R$.
- L is discrete and probabilistically depends on the values of X and K. i.e., $T = L(S, C(X, K)) + R$, where, S follows a discrete distribution.
- Lastly, L is continuous and probabilistically depends on the realisations of X and K. i.e., $T = L(S, C(X, K)) + R$, where, S follows a continuous distribution.

Now, consider $Z = L \circ C(X, K)$, when L is deterministic and $Z = L(C(X, K), S)$, when L is probabilistic, then the MI for the ideal adversary, I^b, can be represented as $I(T; Z) = H(T) - H(T|Z)$, while the MI between the random inputs and the observable trace can be written as:

- $I^{\mathsf{k}} = I(T; (X, K)) = H(T) - H(T|(X, K))$
- $I^{\mathsf{k}} = I(T; (X, K, S)) = H(T) - H(T|(X, K, S))$

Clearly, I^b and I^{k} only differ from each other in the conditional entropy term. Consequently, our proof argument is based on establishing the conditions under which these two conditional distributions are equal, in all three cases. A basic assumption is thus that the conditional entropy exists. A further assumption for the probabilistic leakage functions is that the entropy of the noise distribution is independent of a shift in location.

Remark 1 (Must we check the condition for the distribution of R?). We wish to point out that the distributional assumption (entropy is location independent) about the noise R holds for **all** the distributions that so far have been mentioned in the existing side channel literature, e.g. [20]. In particular, the entropy condition applies to distributions like Gaussian, Laplacian, Cauchy, Uniform, etc.

However, it is possible to check this assumption efficiently if this is desirable. One can evaluate the entropy criterion for a given set of traces and intermediate values by, for example, applying the Kolmogorov-Smirnov test [27] for goodness of fit on samples of the leakage partition $(T|Y = y)$ to identify which distribution they belong to.

3.4 Practical MI Estimation Using the GKOV Method

We now turn our attention to the practical aspects of estimating MI quantities, by leveraging the recent estimator proposed by Gao et al. [14]. For completeness, we briefly summarise our implementation choices in this section and show

some convergence results of the estimator in the higher dimensions: this aspect was theoretically discussed only in the original publication, while we provide simulation experiments specific to the side channel setting.

Fast implementation of Alg.1. Gao et al. [14] provide a Python implementation of their estimator[9]. However, we developed a much more efficient and generic implementation that works for high-dimensional data, which is important for side channel analysis. For our `C++` implementation, we used the popular machine learning library `mlpack`. The library offers several in-built distance metrics including the option of providing a custom distance metric. From the available options of efficient nearest neighbour search algorithms, we used `VPTree` and `BallTree`. For all experiments, we used an Intel(R) Core(TM) i7-8700 CPU 3.20GHz system having 6 CPU cores and Ubuntu operating system.

For calculating distances of each sample point from all other points which is necessary beyond the NN search, we have used OpenMP to parallelize the computation. Note that the OpenMP library can also be used by `mlpack` if it is available on the system. A particular observation on this part of our experiment is that for multidimensional leakage, computing the ℓ_∞ norm with an unrolled loop is more efficient than using the looped version or the `mlpack` library function for the same.

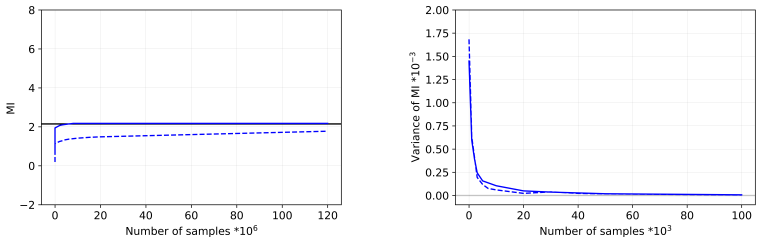
We investigated specific choices for the k_n function and settled for $k_n = \log n$. We provide experiments and a brief discussion in Appendix A.

Remark 2. A consistent property of the Gao et al. [14], that we noticed in our experiments, is that it approaches the theoretical MI value from below. This implies that for MI values close to zero, the estimator takes negative values.

Convergence in a multivariate setting. The GKOV estimator does elegantly generalise to multiple points because its only configuration parameter is the function k_n (which is based on the sample size n but no other feature of the data). For higher dimensional data, a parallel or vectorised tree search can be applicable for the efficient implementations of the k-NN search algorithm.

We provide representative experimental results that show how the dimensionality of data impacts on the convergence behaviour of the GKOV estimator. It is simple to calculate the closed form of the true MI for a multivariate Gaussian distribution, thus we are simulating the pair (X, Y) for X drawn from different dimensions $d_x = 5, 10$, while maintaining that Y is univariate $d_y = 1$. This choice reflects the nature of real-world side channel experiments, where the intermediate value is univariate, but the traces are multivariate. Figures 2a and 2b demonstrate that the estimator converges quickly, e.g., for $d_x = 10$: the variance approaches zero as, obviously, the number of traces increases.

[9] https://github.com/wgao9/mixed_KSG/blob/master/mixed.py



(a) Convergence of Bias (b) Convergence of variance

$(d_x = 5, d_y = 1)$: ——— , $(d_x = 10, d_y = 1)$: - - - , $I(X;Y)$: ———

Fig. 2: Convergence of GKOV MI in multivariate setup

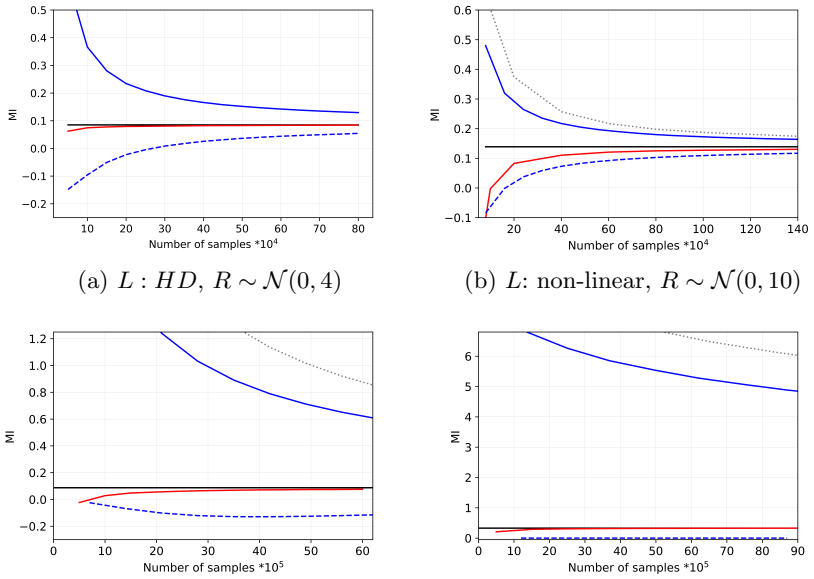
3.5 Characterising the Ideal Adversary using GKOV

Like in previous work, we use simulations to produce fully controlled experiments, so that the mutual information can both be calculated as well as estimated. All experiments are based on a single bijective target function, which is the AES SubBytes mapping, $y = C(x, k) = Sbox(x \oplus k)$; this keeps our experiments comparable with previous works. To compute the non-parametric MI estimates (eHI and ePI) from previous work, we use the scripts[10] that were provided by Bronchain et al. [6]. Note that the ePI and eHI are only defined for use with two discrete random variables, and the scripts include a step where traces are discretised.

In our simulations, we vary the device leakage function as well as the type and magnitude of the noise distribution, and we consider univariate and multivariate analyses. The considered leakage functions are the Hamming weight (HW), the Hamming distance (HD, between Y and the target $C(Y)$), the non-linear function given by the first DES S-box when applied to the 6 least significant bits of Y, and a linear function given as a weighted sum of the bits of Y. The noise R follows either a Gaussian ($\mathcal{N}(0, \sigma)$), a Laplacian ($\mathcal{L}(0, \sigma)$) or a discrete-Laplacian (discrete $\mathcal{L}(0, \sigma)$) distribution. In our experiments, we consider $\sigma \in [2.8, 10]$. In the multivariate simulations, the simulated trace points are either based on the HW or the HD leakage of some bits of Y. We give the exact specifications as part of the respective figures representing the experimental results.

We run a large number of simulated experiments and include a representative subset of outcomes in Figures 3a-3d. In all figures, the black line corresponds to the true MI, the blue lines, correspond to the ePI and eHI, and the red line shows GKOV. We also include the non-parametric estimator based on histogram-pdf estimation from Antos and Kontoyiannis [1] as a reference for a provably consistent estimator specific to discrete data. The four figures cover several different scenarios, whereby the two lower plots show a bi-variate experiment and a trivariate experiment.

[10] https://github.com/obronchain/Leakage_Certification_Revisited



(a) $L : HD$, $R \sim \mathcal{N}(0, 4)$ (b) L: non-linear, $R \sim \mathcal{N}(0, 10)$

(c) $L : (HW, HD)$, $R \sim \mathcal{N}(0, 4)$ (d) $L : (HW, HW, HD)$, $R \sim \mathcal{N}(0, 2)$

$I_n{}^{\text{hist}}$: ····· , eHI: —— , $I((X, K); T)$: —— , $I_n{}^{\text{GKOV}}$: —— , ePI: - - -

Fig. 3: Ideal Adversary: eHI-ePI vs I^{k}

Interpretation of results. The results show that the GKOV estimator (red line) converges to the true mutual information value (black line) faster than the other MI estimates. According to the figures, it is evident that as the dimension increases, the eHI convergence rate deteriorates more dramatically than GKOV. Gao et al. [14] already analyse the asymptotic complexity of their estimator, which shows that the trace complexity is not dependent on the number of dimensions of the data. We provide a summary of their reasoning in App. A.

In the results, we can observe that the regret (i.e. the difference between MI estimated via the eHI and the PI) increases when we move towards multivariate side channel observations. It is obvious from the experiments that a more trace-efficient estimator for the eHI and ePI is needed. One might question why the simulations here are limited to at most three dimensions (after all, we ran experiments for up to 10 dimensions before). This is only because we were unable to run ePI, eHI, and the histogram-based estimator for four or more shares. This is not a shortcoming of our implementation — because they need to explicitly create a multivariate pmf, making any higher order analysis highly computationally and memory intensive; they suffer much more from the "curse of dimensionality". Our observations further motivate interest in the recent results of Masure et al. [28].

4 Case study: LPC1313

In this section, we use a data set that was acquired from executing a two-shares masked AES SubBytes implementation (written in Thumb Assembly Language) on an ARM Cortex M3 processor core from NXP (LPC1313). This implies that no single point leaks information about the unshared intermediate value, further confirmed via leakage detection. We use a custom measurement board (Picoscope 5243D), which provides good measurements (at 250 MSa/s, and the working frequency is set to 2 MHz). We use our scope in a basic setting to avoid any trace processing (de-noising) and extract discrete traces ($n = 10^6$), where each point is represented by 8 bits.

The purpose of this case study is to show that the GKOV estimator correctly returns MI estimates, and therefore characterises the ideal adversary, in a real world setting.

4.1 Characterising the Ideal Adversary

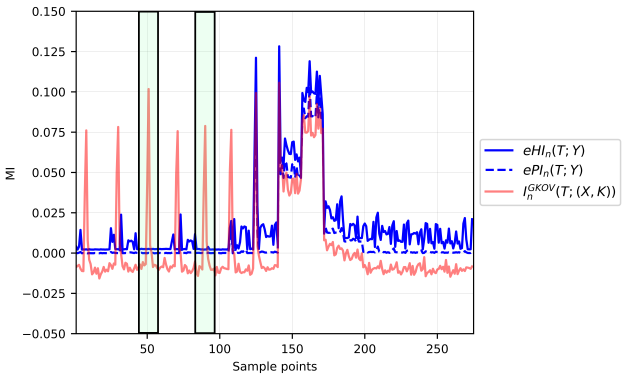
To assess the ideal adversary, a bivariate MI estimation must be carried out, which we do via the ePI-eHI as well as via estimating the MI via GKOV.

Figure 4a shows the results of this experiment. Each point in this trace is the result of applying the ePI, eHI, and GKOV estimator to two points of the power traces (to produce this picture we selected a small number of trace points from the dataset). We can see that the ePI is always lower than the eHI (as expected) and that GKOV returns negative values for trace points that have a very low MI (as expected).

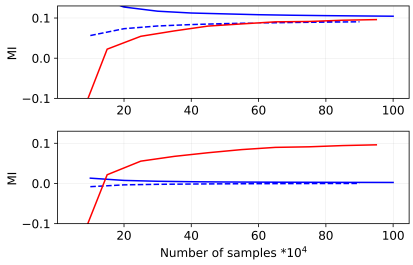
It is however striking that there are trace points that are highlighted by the GKOV estimate for $I^{\mathsf{k}} = I((X, K); T)$ as showing a high MI, which are missed by eHI − ePI (we highlight two of these with a rectangle).

Interpretation of the results. Our simulations in the previous section already demonstrated that the eHI and ePI return biased results in the case of multivariate data, which indicates that also in this bivariate experiment, they may, at times, return misleading results. However, we also must consider the possibility that with a larger number of traces, the eHI and ePI may eventually indicate a non-zero MI as well.

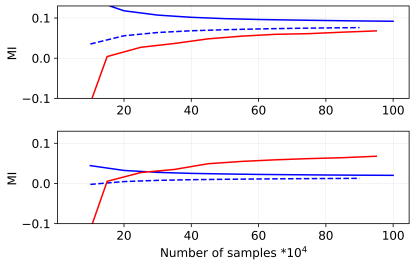
Thus, we run convergence experiments at two highlighted MI points (51 and 108) resulting in the bottom plots in Figures 4b and 4c. These outcomes clearly show that when increasing the number of traces, the ePI and eHI estimators even more strongly indicate a zero MI, and GKOV clearly keeps indicating a non-zero MI. Given that GKOV is asymptotically unbiased, we may conclude that it provides the correct result. As independent verification for this interpretation, we performed a further analysis: we estimate the MI *after applying the classical mean-free product combination function to the two points in the original dataset*, implying we do a univariate MI estimation: now all three estimators agree on the same non-zero MI. Thus, the discrepancy in the bivariate analysis is indeed caused by the eHI and ePI estimation.



(a) Bivariate discrete real device leakage



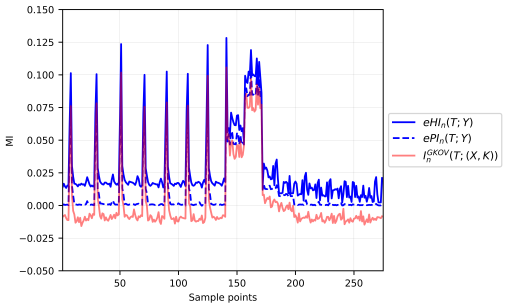
(b) Sample point = 51 (top: preprocessing, bottom: no preprocessing)



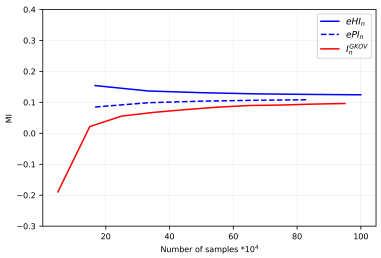
(c) Sample point = 108 (top: preprocessing, bottom: no preprocessing)

Fig. 4: Characterisation of the Ideal Adversary for masked software AES

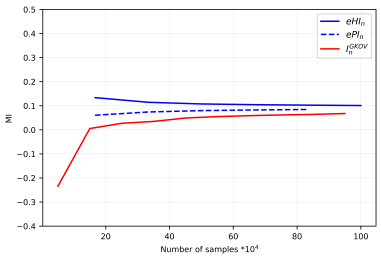
Investigating this issue further, we noticed that the distributions of the two trace points, which are used in the bivariate MI estimation, are markedly different to the distributions of points that the script by Bronchain was designed for. Therefore, we implemented our own adaptive binning strategy: this means that for each pair of data-points we now first analyse their distribution, and then specifically select a binning strategy for each pair. This is possible for this data set, where each trace point can at most take 256 values. With this adaptation, the eHI/ePI estimators return bounds indicating a non-zero MI, as clearly visible in Fig. 5a. We also provide convergence plots for the two trace points in Fig. 5b and Fig. 5b. With this binning strategy, the eHI/ePI bounds concur with the GKOV estimate.



(a) Bivariate discrete real device leakage with adaptive binning



(b) Sample point = 51 (with adaptive binning)



(c) Sample point = 108 (with adaptive binning)

Fig. 5: Correct eHI/ePI estimates via adaptive binning during estimation

5 Case Study: AES-HD

In this section, we use the AES-HD dataset[11]. This dataset was gathered using a Xilinx Virtex-5 FPGA implementation of AES-128. This unprotected implementation was first introduced by Picek et al. [32]. The 1250 time points that represent the target's electromagnetic emanations (EM) are used in the side channel measurements. In total, 500,000 traces were captured when the target encrypted 500,000 randomly generated plaintexts with a fixed key. From these 500,000 measurements, we select the first 450,000 as profiling traces and the last 50,000 as attack traces. The purpose of this case study is to demonstrate how our framework can be used to assess concrete adversaries, both relatively as well as in comparison to the ideal adversary. This time we also include uni-variate predictive models in our study.

We assess concrete adversaries that build leakage models for the last round's *Sbox* output as it overwrites a previous value in the corresponding register: thus the intermediate value is $Y^{(i)} = Sbox^{-1}[C_j^{(i)} \oplus k^*] \oplus C_{j'}^{(i)}$, where $C_j^{(i)}$ and $C_{j'}^{(i)}$ are

[11] https://github.com/AISyLab/AES_HD_Ext

two ciphertext bytes corresponding to the i-th trace, and k^* is the corresponding round key byte. The relation between j and j' is given by the ShiftRows operation of AES and we consider $j = 12$ and $j' = 8$. We utilise MI estimations using our GKOV implementation to determine 30 points of interest for model estimation, and we characterise the ideal adversary for these 30 points.

5.1 Comparing predictive models

We have examined three predictive uni-variate leakage models: a classical Gaussian template model (GT), a linear regression based on a model that includes linear terms only (LR), and a model that is based on the 6-least significant bits of the intermediate value. Figure 6a shows estimated MI values for the three values; the red line gives the best MI, $I_n(Y, T)$; we can see that the regression model and Gaussian template perform roughly equally well, while the 6LSB model is clearly not a good choice.

Interpretation of results. Both the regression model and Gaussian template model capture, in fact, the same information because we restrict both models to only capture the linear components of the leakage; thus they perform nearly identically across all data points. The 6LSB model ignores two bits of information and is therefore inappropriate. We can also observe that those data points with the highest MI values then give the best models.

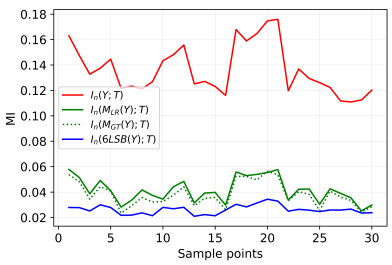
5.2 Comparing classification models

We now consider two deep net classifiers (M_1^*, M_2^*). For both networks, we use an MLP with four hidden layers (with 64, 32, 16 and 4 neurons, respectively) and one output layer. We train these for two different numbers of epochs such that the M_1^* one should be better than M_2^*. In Fig. 6b, we plot the MI estimators $\hat{I}_n(Y; Y|T_{M_i^*})$ for $i = 1$ and 2 , where $Y|T_{M_i^*}$ denotes the predicted label related to the conditional probability distribution $P_{Y|T_{M_i^*}}$ obtained from the classifier M_i^*. We also evaluate the multivariate MI estimator corresponding to 30 time points (plotted as a red line) and the intermediate, which represents the actual information leakage and finally compare with the MI estimators corresponding to two different classifiers. It is clear from the plot that M_1^* is better than M_2^*.

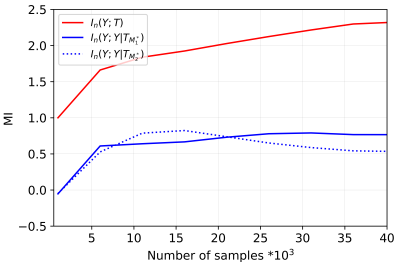
Interpretation of results. In contrast to the predictive models, our classification models now take advantage of the information of all 30 data points, they thus capture considerably more information as demonstrated by the higher MI values. We can observe that the model that we ran for more epochs performs asymptotically better, which is what we would expect.

5.3 Comparing profiling complexity

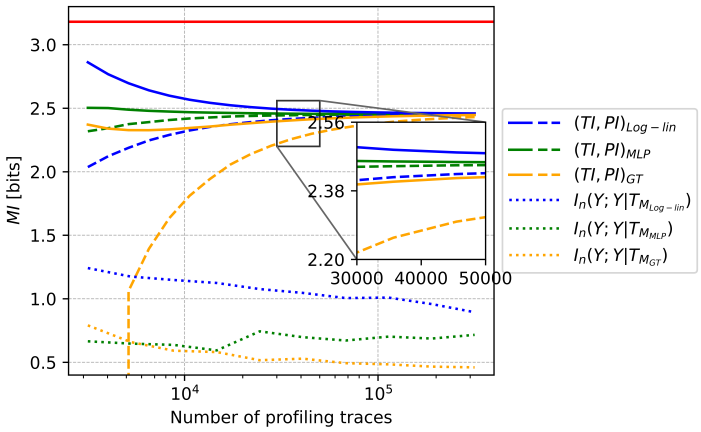
A question of practical relevance is often related to how much effort should be put into learning a leakage model. An evaluator will often start with several



(a) Predictive Models



(b) Classification Models



(c) Comparing profiling complexity

Fig. 6: Model Comparison, AES-HD Dataset

model building approaches (typically including classical statistical models and often a deep learning model) and then needs to decide at which point each of the resulting models is "as good as it will get", i.e. such that the estimation error is as small as the approach allows, and thus training can stop. To demonstrate the efficacy of our quantities and estimator, we now show an experiment where we fix the number of traces for validation of a model and vary the number of traces for training the model and plot MI estimates as well as PI and TI estimates (we draw on the work and implementation of Masure et al. [28]) in Fig. 6c. The green lines correspond to an MLP with four hidden layers (with 64, 32, 16 and 4 neurons, respectively) and one output layer. The blue and orange lines correspond to logistic regression and Gaussian templates (dotted lines are for comparing classification models, full lines are the TI estimates and dashed lines are the PI estimates). All learning approaches take 10 informative points (selected from the 30 points that we considered for comparing predictive

models). We also plot the best estimate that we have for these 10 points of the MI characterising the ideal adversary.

Interpretation of results. All estimated quantities conclude that for the selected learning methods (and trace points) the logistic regression leads to the best model; however, all models are very close and converge roughly at the same speed. Given that the MI characterising the ideal adversary is considerably higher than the MI quantities characterising the concrete adversaries, we can conclude that none of these models is particularly good. The classical learning approaches offer little room for improvement, but the MLP architecture should be improved to extract more information from the trace points.

Acknowledgments

Aakash Chowdhury has been supported in part by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement number 725042), and by the Austrian Science Fund (FWF) 10.55776/F85 (SFB SpyCode). Elisabeth Oswald has been supported in part by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement number 725042) and in part by the EU Horizon project (enCrypton, grant agreement number 101079319). Carlo Brunetta joined the author team whilst in Simula UiB (Bergen, Norway). All authors would like to thank the anonymous reviewers for their constructive comments.

A The GKOV estimator

Gao et al. [14] proved that for $\tilde{k}_i = k \ \forall i$, the bias of $I_n(X;Y)$ is equal to $O\left(\frac{(\log n)^{(1+\delta)\left(1+\frac{1}{d_x+d_y}\right)}}{n^{\frac{1}{d_x+d_y}}}\right)$, where, $\delta > 0$ is very small and d_x, d_y are the finite dimensions of random vectors X and Y, respectively (for details, see Theorem 4,5 of [15]). They also showed that the variance of fixed k-NN MI estimator is independent over the dimension of the data as it is equal to $O\big(\frac{(\log n)^2}{n}\big)$. The computation of $d_{i,xy}, n_{x,i}, n_{y,i}$ in Algorithm 1 can be done using any nearest neighbour binary tree search algorithm. Using then the efficient ball tree algorithm[12] the overall complexity of the GKOV MI estimator is $O((d_x+d_y)n\log n)$. To combat the inevitable "curse of dimensionality" in the k-NN algorithm (as described in [23]) one could further use a parallel ball tree construction (as proposed in [35]).

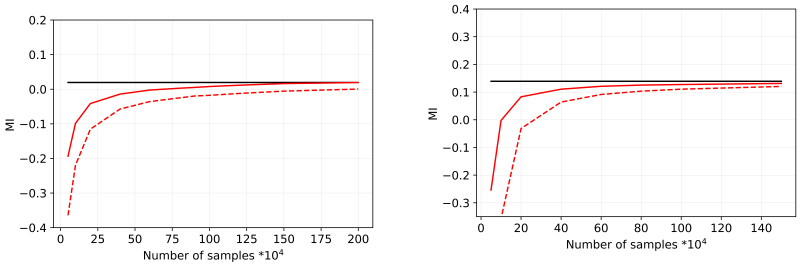
Choice of parameter k_n: according to Gao et al.'s Theorem 1 [14], the parameter k_n, should be chosen such that as $n \to \infty$, $k_n \to \infty$ and both $(k_n \log n)/n$,

[12] https://scikit-learn.org/stable/modules/neighbors.html#unsupervised-neighbors

Algorithm 1 Non-parametric $I(X;Y)$ estimation for mixed r.v.s (X,Y) [14]

Require: $\{x_i, y_i\}_{i=1}^{n}$ and $k_n = k$

for $i = 1, \ldots, n$ **do**
  $d_{i,xy} = k$-th smallest distance from$\{d_{ij} = \max\{\|x_j - x_i\|, \|y_j - y_i\|\} : i \neq j\}$
  if $d_{i,xy} = 0$ **then**
    $\tilde{k}_i = |\{j : d_{ij} = 0\}|$
  else
    $\tilde{k}_i = k$
  end if
  $n_{x,i} = |\{j : \|x_j - x_i\| \leq d_{i,xy}\}|$
  $n_{y,i} = |\{j : \|y_j - y_i\| \leq d_{i,xy}\}|$
  $\alpha_i = \psi(\tilde{k}_i) - \log(n_{x,i} + 1) - \log(n_{y,i} + 1)$
end for
return $\frac{1}{n}\sum_i \alpha_i + \log(n)$



(a) L: linear, $R \sim \mathcal{L}(0, 8)$ (b) L: non-linear, $R \sim \mathcal{N}(0, 10)$

$k_n = \log(n)$: —— , $k_n = \log_{10}^2(n)$: - - -, $I((X,K);T)$: ——

Fig. 7: Convergence experiments for different choices of k_n.

$(k_n \log n)^2/n$ converge to zero. In our experiments, we consider k_n using this criterion. It is important to note that unlike a plug-in (histogram) estimator, which requires data-dependent parameter tuning, the choice of the parameter k_n can be pre-determined based only on the sample size n. Moreover, the choice of k_n only affects the rate of convergence, i.e. the efficiency of the estimation, unlike histogram based estimators, where a wrong choice can lead to bias.

A.1 Establishing Practical Choices for k_n

For simulated side channel experiments, we select k_n equal to $\log n$ and $\log_{10}^2 n$ for comparison. Figure 7 shows some representative experimental results for the GKOV estimator as implemented (via Alg. 1) in different situations. To create these plots, we performed a number of simulations where we varied both device leakage functions and noise distributions.

The results in Fig. 7 illustrate that the convergence rate for $k_n = \log n$ has a smaller advantage over the other one. In the remaining practical experiments, we will thus show results for $k_n = \log n$.

An observation is that the GKOV estimator approaches the true MI from below. There is no formal proof for this in [14], but in all our experiments we observed this behaviour. This implies that if an MI quantity is close to zero, then the GKOV estimator will take negative values until enough samples are available and it crosses the zero line and is positive. This behaviour is not a sign of bias (note that [14] shows the asymptotic unbiasedness of their estimator).

B Considering Discretisation

B.1 The Impact of Discretisation

The convergence guarantee for the eHI towards the MI requires the assumption that the traces are discrete (but typically they are not). Discretisation divides the range of a continuous random variable X into possibly an infinite or finite number of intervals. Drawing on Darbellay and Vajda's results [11, cf. Proposition 1], we now provide a concrete mathematical characterisation for the MI between a discrete and a discretised continuous random variable.

Darbellay and Vajda [11] consider two (continuous) random variables X, Y and the use of a simple partitioning of the space $X \times Y$ into rectangles. Typically, such a partitioning $\mathcal{P}$ is a product partitioning i.e. $\mathcal{P} = \mathcal{I} \times \mathcal{J}$ where $\mathcal{I}$ and $\mathcal{J}$ are partitioning of X and Y respectively [13]. We denote the discretised random variables obtained from such partitioning as $X^{\mathcal{I}}, Y^{\mathcal{J}}$.

We can now show that the MI which is based on the discretised leakage is smaller or equal to the MI based on the non-discretised leakage. This implies that an evaluator who discretises traces for the estimation of mutual information will underestimate the strength of an adversary who works with the non-discretised traces.

Proposition 1. *Let X, Y be two random variables with pmf p_X and pdf f_Y respectively. Let $\mathcal{P} = \mathcal{I} \times \mathcal{J}$ be the product partitioning of $X \times Y$ as described above (the partitioning $\mathcal{I}$ is defined by the discrete X). Then $I(X;Y) \geq I(X^{\mathcal{I}}; Y^{\mathcal{J}})$.*

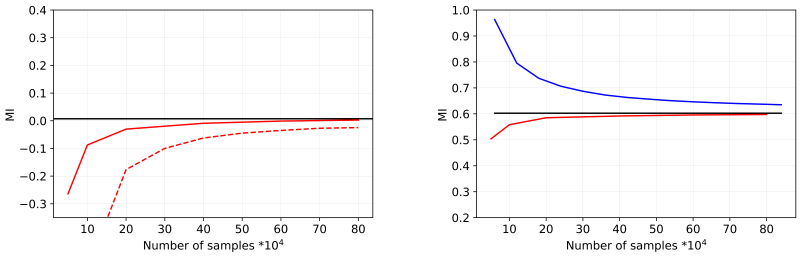
Proof. We assume that the joint distribution exists. As explained by Darbellay and Vajda [11, Section II], for the product partition $\mathcal{P}$ we can write that

$$I(X;Y) = I(X^{\mathcal{I}}; Y^{\mathcal{J}}) + D_{\mathcal{P}}(X;Y)$$

where $D_{\mathcal{P}}(X;Y)$ is the residual divergence, see [11, cf. Proposition 1] for the definition.

From their argument, we observe that the residual divergence $D_{\mathcal{P}}(X;Y) \geq 0$ for any partition $\mathcal{P}$(including the specific partition that is given by a discrete X). Thus, the result follows. □

[13] In the side channel community, a similar method is often implemented by partitioning the leakage into a countably finite number of intervals, which then define the bins for histogram based estimation techniques—this is also the method used by Bronchain et al.[6] for the eHI.



(a) L: linear, $R \sim \mathcal{N}(0, 10)$ (b) L: non-linear, $R \sim \mathcal{N}(0, 4)$

${I_n}^{\text{GKOV}}((X, K); T)$: —— , ${I_n}^{\text{GKOV}}((X, K); [T])$: - - -, $I((X, K); T)$: ——, eHI: ——

Fig. 8: The impact of discretisation on the MI and its estimation via the eHI.

Proposition 2 of Darbellay and Vajda [11, Section II] goes on to develop that the residual divergence converges to zero asymptotically for increasingly finer product partitions. Consequently, in practice when we set the number of partitions to finite, the residual divergence is strictly larger than zero, and thus we always lose information upon discretisation:

$$I(X; T) > I(X; [T]) = \lim_{n \to \infty} \mathbb{E}[\text{eHI}_n(X; [T])] \tag{12}$$

In the next section, we provide practical experiments that show the effect of Prop. 1 in action. Proposition 1 also implies that the eHI is not necessarily an upper bound to the MI in the context of any arbitrary continuous traces (it also depends on the bias that it has, which is different in different settings).

B.2 Practical Demonstration

Before moving away from simulations, we briefly demonstrate the information loss that is incurred by the discretisation of traces.

We simulate traces with Gaussian noise and a linear leakage function, as well as a non-linear leakage function, applied to an intermediate value resulting from the SubBytes function. Because all distributions are known, we can compute the MI theoretically.

Figure 8a shows what happens when we use GKOV to estimate the mutual information $I_n((X, K); T)$ and $I_n((X, K); [T])$. The theoretical MI value for $I((X, K); T)$ is also provided. We can clearly observe that the mutual information estimate for the discretised traces is considerably lower than the estimate that uses the traces "as they are".

Figure 8b shows a plot that includes the eHI: because it approaches the true value from above, and since it is biased, it remains, in this case, an upper bound for the true MI.

C Proof of Lemma 1

Result 1. The PI between the three variables $Y, [M], [T]$, and all distributions defined for all $y \in \mathcal{Y}$, can be written as:

$$PI(Y; [T]; [M]) = I(Y; [T]) - \mathbb{E}[D_{KL}((Y|t, T)||(Y|t, M))]$$

Proof. From Eq. 6 (see Page 10), we have

$$\mathrm{PI}(Y; [T]; [M]) = H(Y) + \sum_{y \in \mathcal{Y}} p_Y(y) \cdot \sum_{t \in [\mathcal{T}]} p_{(Y,[T])}(t|y) \log_2 p_{(Y,[M])}(y|t)$$

We substitute $p_{(Y,[M])}(y|t) = p_{(Y,[M])}(y|t)\frac{p_{(Y,[T])}(y|t)}{p_{(Y,[T])}(y|t)}$ and finally can write:

$$\begin{aligned}
&= H(Y) + \sum_{y \in \mathcal{Y}} p_Y(y) \cdot \sum_{t \in [\mathcal{T}]} p_{(Y,[T])}(t|y) \log_2 \left(p_{(Y,[M])}(y|t) \frac{p_{(Y,[T])}(y|t)}{p_{(Y,[T])}(y|t)} \right) \\
&= H(Y) + \sum_{y \in \mathcal{Y}} p_Y(y) \cdot \sum_{t \in [\mathcal{T}]} p_{(Y,[T])}(t|y) \left(\log_2 p_{(Y,[T])}(y|t) + \log_2 \frac{p_{(Y,[M])}(y|t)}{p_{(Y,[T])}(y|t)} \right) \\
&= H(Y) + \sum_{y \in \mathcal{Y}} p_Y(y) \cdot \sum_{t \in [\mathcal{T}]} p_{(Y,[T])}(t|y) \log_2 p_{(Y,[T])}(y|t) \\
&\quad + \sum_{t \in [\mathcal{T}]} p_{[T]}(t) \cdot \sum_{y \in \mathcal{Y}} p_{(Y,[T])}(y|t) \log_2 \frac{p_{(Y,[M])}(y|t)}{p_{(Y,[T])}(y|t)} \\
&= I(Y; [T]) - \mathbb{E}[D_{KL}((Y|t, [T])||(Y|t, [M]))]
\end{aligned}$$

□

D Relating I^b to I^k

D.1 Equality of I^b and I^k when L is discrete

Characterising the conditional distributions. We first study the conditional distribution of $T|Z$. It is easy to see that this conditional distribution is completely defined by the distribution of R:

$$\begin{aligned}
F_{T|Z}(t|z) &= P(T \leq t | Z = z) \\
&= P(Z + R \leq t | Z = z) \\
&= P(z + R \leq t) \text{ (as, } Z \text{ is independent of } R) \\
&= F_R(t - z) \ \forall t \in \mathbb{R}
\end{aligned} \tag{13}$$

Consequently, the pdf $f_{T|Z}$ of the conditional variable $T|Z$ is given by the pdf of R.

We now consider the conditional distribution of $T|(X,K)$ when L is deterministic (i.e. depends only on X, K and the cryptographic function C).

$$\begin{aligned} F_{T|(X,K)}(t|(x,k)) &= P(T \leq t|(X,K) = (x,k)) \\ &= P(L \circ C(X,K) + R \leq t|(X,K) = (x,k)) \\ &= F_R(t - L \circ C(x,k)) \ \forall t \in \mathbb{R} \end{aligned} \tag{14}$$

It follows again that the pdf of $T|(X,K)$ is given by the pdf of R. This observation has been formalised before [20, Corollary 3.]. Note that, by using the same technique as above it is also obvious that when L is discrete and probabilistic,

$$F_{T|(X,K,S)}(t|(x,k,s)) = F_R(t - L(s, C(x,k))) \ \forall t \in \mathbb{R} \tag{15}$$

Now, with these properties of conditional distributions, we show that I^b is equal to I^{k} for both cases when L is deterministic and probabilistic, respectively.

Proposition 2. *If L is discrete and $T = L \circ C(X,K) + R$, then for any well-defined [14] function $C(\cdot)$, the following equality will hold*

$$I^b = I(T;Z) = I(T;(X,K)) = I^{\mathsf{k}}$$

Proof. We recall that $Z = L \circ C(X,K)$, and suppose it has m realisations. It is clear that the probability of $Z = z_i$ is given by the number of pairs (x,k) that map to z_i. Thus, we have

$$\begin{aligned} p_Z(z_i) = P(Z = z_i) &= P\{(X,K) = (x,k) : L(C(x,k)) = z_i\} \\ &= \sum_{\substack{(x,k): \\ L(C(x,k))=z_i}} p_{(X,K)}(x,k) \quad \text{for } i = 1,2,..,m \end{aligned} \tag{16}$$

[14] An assignment of values y to elements $x \in \mathcal{X}$ is said to be a *well-defined* function $f: X \to Y$ if it satisfies the following three properties:

- Totality: For every $x \in \mathcal{X}$, $\exists\ y$ such that $f(x) = y$.
- Existence: For every $x \in \mathcal{X}$, $f(x) \in \mathcal{Y}$.
- Uniqueness: For every $x \in \mathcal{X}$, there is only $y \in \mathcal{Y}$ such that $f(x) = y$.

(Here, every pair (x,k) maps to exactly one z_i, because C is well defined). We use this observation to rewrite I^b :

$$\begin{aligned}
I^{\mathsf{k}} &= I(T;(X,K)) \\
&= H(T) - \sum_{(x,k)\in(\mathcal{X}\times\mathcal{K})} p_{(X,K)}(x,k)\mathbb{E}_{T|(x,k)}\left[-\log_2(f_{T|(X,K)}(t|(x,k)))\right] \\
&\overset{\text{Eq.(14)}}{=} H(T) - \sum_{i=1}^{m} \sum_{\substack{(x,k):\\ L(C(x,k))=z_i}} p_{(X,K)}(x,k)\mathbb{E}_R\left[-\log_2(f_R(t-L\circ C(x,k)))\right] \\
&\overset{\text{Eq.(16)}}{=} H(T) - \sum_{i=1}^{m} p_Z(z_i)\mathbb{E}_R\left[-\log_2(f_R(t-z_i))\right] \\
&\overset{\text{Eq.(13)}}{=} H(T) - \sum_{i=1}^{m} p_Z(z_i)\mathbb{E}_{T|z_i}\left[-\log_2(f_{T|Z}(t|z_i))\right] = I^b \square
\end{aligned}$$

Proposition 3. *Suppose, R follows a distribution with the location and scaling parameters μ and σ (>0) respectively. Let X, K denote the discrete plaintext and uniformly drawn discrete key (both are independently distributed), and the leakage function L is discrete such that $T = L(S, C(X,K))+R$. If, the differential entropy of R is independent of location shift*[15] *(i.e., $H(R) = \phi(\sigma)$, where ϕ depends only on the pdf f_R), then the following equality holds:*

$$I^b = I(T;Z) = I(T;(X,K,S)) = I^{\mathsf{k}}$$

Proof. First, we compute I^b:

$$\begin{aligned}
I^b = I(T;Z) &= H(T) - H(T|Z) \\
&= H(T) - \sum_{z\in\mathcal{Z}} p_Z(z)\mathbb{E}_{T|z}\left[-\log_2(f_{T|Z}(t|z))\right] \\
&\overset{\text{Eq.(13)}}{=} H(T) - \sum_{z\in\mathcal{Z}} p_Z(z)\mathbb{E}_R\left[-\log_2(f_R(t-z))\right]
\end{aligned}$$

Second, we derive I^{k}:

$$\begin{aligned}
I^{\mathsf{k}} &= I(T;(X,K,S)) \\
&= H(T) - \sum_{x,k,s} p_{(X,K,S)}(x,k,s)H(T|(x,k,s)) \\
&= H(T) - \sum_{x,k,s} p_{(X,K,S)}(x,k,s)\mathbb{E}_{T|(x,k,s)}\left[-\log_2(f_{T|(X,K,S)}(t|(x,k,s)))\right] \\
&\overset{\text{Eq.(15)}}{=} H(T) - \sum_{x,k,s} p_{(X,K,S)}(x,k,s)\mathbb{E}_R\left[-\log_2(f_R(t-L(s,C(x,k))))\right]
\end{aligned}$$

[15] An illustration of location independent entropy:
Suppose, X_1,X_2 follow univariate normal distribution with different means μ_1 and μ_2, respectively but have same variance σ^2. Then, $H(X_1) = H(X_2) = \frac{1}{2}\log_2(2\pi e\sigma^2)$

Clearly, we already know from the entropy condition that $H(R) = \phi(\sigma)$, when $R \sim f_R(t-z)$ or when $R \sim f_R(t - L(s, C(x,k)))$. Hence, we can say that $\mathbb{E}_R[-\log_2(f_R(t-z))]$ is equal to $\mathbb{E}_R[-\log_2(f_R(t - L(s, C(x,k))))]$, which implies $I^b = I^k$. □

D.2 Equality of I^b and I^{k} when L is Continuous

Characterising the conditional distributions. The continuity of L is due to some randomness of the continuous variable S that depends on X, K and the target function C but importantly we still have the independence between $Z = L(Y)$ and R. To derive the distribution of $T|Z$ and later $T|(x,k,s)$, we need a little bit more machinery than before because L is continuous. This scenario was not covered by Heuser et al. [20, Corollary 3.].

Given their joint distribution, the distribution of a function of two random variables can be derived by a technique that is known as "change of variables"[37]. The trick works as follows, given two variables (X_1, X_2) and two functions u_1 and u_2 such that $Y_1 = u_1(X_1, X_2)$ and $Y_2 = u_2(X_1, X_2)$, with inverses $X_1 = v_1(Y_1, Y_2)$ and $X_2 = v_2(Y_1, Y_2)$; the joint pdf of (Y_1, Y_2) is given by $f_{(Y_1,Y_2)}(y_1, y_2) = |J| \cdot f_{(X_1,X_2)}(x_1, x_2)\big|_{\{x_1 = v_1(y_1,y_2), x_2 = v_2(y_1,y_2)\}}$. The value $|J|$ is the absolute value of the Jacobian $J = \left|\frac{\partial(x_1,x_2)}{\partial(y_1,y_2)}\right| = \begin{vmatrix} \frac{\partial(x_1)}{\partial(y_1)} & \frac{\partial(x_1)}{\partial(y_2)} \\ \frac{\partial(x_2)}{\partial(y_1)} & \frac{\partial(x_2)}{\partial(y_2)} \end{vmatrix}$. Knowledge of the joint distribution (Y_1, Y_2) enables to derive the distributions of Y_1 (and Y_2 respectively) by marginalisation.

We first derive the distribution of $T|Z$. Hence we apply the change of variables technique to derive the distribution of $T = Z + R$, Z, and choose $Y_1 = Z + R$, $Y_2 = Z$. Hence $|J| = 1$, and this gives

$$
\begin{aligned}
& f_{T,Z}(t,z) = 1 \cdot f_{R,Z}(t-z, z) = 1 \cdot f_Z(z) \cdot f_R(t-z) = f_Z(z) \cdot f_R(t-z) \\
& \Rightarrow f_{T|Z}(t,z) = \frac{f_{T,Z}(t,z)}{f_Z(z)} = \frac{f_Z(z) \cdot f_R(t-z)}{f_Z(z)} = f_R(t-z) \qquad (17)
\end{aligned}
$$

Using the same trick, we can also derive the pdf of $T|(x,k,s)$, which will give us $f_R(t - L(s, C(x,k)))$. To achieve this, we have to consider the following change of variables for each pair $(x,k) \in (\mathcal{X}, \mathcal{K})$:

$$
(R, S) \to (T, S) : T = L(S, C(x,k)) + R
$$

And the Jacobian of the transformation $J = \frac{1}{\left|\frac{\partial(t,s)}{\partial(r,s)}\right|} = 1$ under the condition that the mapping $L : S \to L(S, C(x,k))$ is one-to-one, which is a criterion for the existence of the partial derivative $\frac{\partial(t)}{\partial(s)}$ (for details see [37]).

Using this property of conditional distribution we now proof the equality between I^b and I^{k} exactly as same as we did in Proposition 3.

Proposition 4. *Suppose, R follows a distribution with the location and scaling parameters μ and σ (> 0) respectively. Let X, K denote the discrete plaintext*

and uniformly drawn discrete key (both are independently distributed), and the leakage function L is continuous such that $T = L(S, C(X, K)) + R$. If, the differential entropy of R is independent of location shift (i.e., $H(R) = \phi(\sigma)$, where ϕ depends only on the pdf f_R), then the following equality holds:

$$I^b = I(T; Z) = I(T; (X, K, S)) = I^{\mathsf{k}}$$

Proof. We prove this proposition simply by using the characterisation of conditional distribution (discussed in Section D.2 on Page 30) in exactly the same way as in Proposition 3. We start with the derivation of I^b:

$$\begin{aligned} I^b = I(T; Z) &= H(T) - H(T|Z) \\ &= H(T) - \int_z f_Z(z) H(T|z) dz \\ &= H(T) - \int_z f_Z(z) \mathbb{E}_{T|z} \left[-\log_2(f_{T|Z}(t|z)) \right] dz \\ &= H(T) - \int_z f_Z(z) \mathbb{E}_R \left[-\log_2(f_R(t-z)) \right] dz \end{aligned}$$

We now derive I^{k} as in the following:

$$\begin{aligned} I^{\mathsf{k}} &= I(T; (X, K, S)) \\ &= H(T) - \sum_{x,k} \int_s f_{(X,K,S)}(x, k, s) H(T|x, k, s) ds \\ &= H(T) - \sum_{x,k} \int_s f_{(X,K,S)}(x, k, s) \mathbb{E}_{T|(x,k,s)} \left[-\log_2(f_{T|(X,K,S)}(t|x, k, s)) \right] ds \\ &= H(T) - \sum_{x,k} \int_s f_{(X,K,S)}(x, k, s) \mathbb{E}_R \left[-\log_2(f_R(t - L(s, C(x, k)))) \right] ds \end{aligned}$$

Based on the entropy criteria of R it is known that $H(R) = \phi(\sigma)$ irrespective of whether $R \sim f_R(t - z)$ or $R \sim f_R(t - L(s, C(x, k)))$. Therefore, we have

$$\begin{aligned} &\mathbb{E}_R \left[-\log_2(f_R(t - z)) \right] = \phi(\sigma) = \mathbb{E}_R \left[-\log_2(f_R(t - L(s, C(x, k)))) \right] \\ &\qquad \implies I^b = H(T) - \phi(\sigma) = I^{\mathsf{k}} \end{aligned}$$

□

References

1. Antos, A., Kontoyiannis, I.: Convergence properties of functional estimates for discrete distributions. Random Structures & Algorithms **19**, 163 – 193 (10 2001). https://doi.org/10.1002/rsa.10019

2. Azouaoui, M., Bellizia, D., Buhan, I., Debande, N., Duval, S., Giraud, C., Jaulmes, É., Koeune, F., Oswald, E., Standaert, F., Whitnall, C.: A systematic appraisal of side channel evaluation strategies. In: van der Merwe, T., Mitchell, C.J., Mehrnezhad, M. (eds.) Security Standardisation Research - 6th International Conference, SSR 2020, London, UK, November 30 - December 1, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12529, pp. 46–66. Springer (2020)
3. Batina, L., Gierlichs, B., Prouff, E., Rivain, M., Standaert, F.X., Veyrat-Charvillon, N.: Mutual Information Analysis: a Comprehensive Study. J. Cryptology **24**(2), 269–291 (2011). https://doi.org/10.1007/s00145-010-9084-8
4. Beirlant, J., Dudewicz, E.J., Györfi, L., Dénes, I.: Nonparametric entropy estimation. an overview. International Journal of Mathematical and Statistical Sciences **6**(1), 17–39 (1997), https://eprints.sztaki.hu/1417/
5. Belghazi, I., Rajeswar, S., Baratin, A., Hjelm, R.D., Courville, A.C.: MINE: mutual information neural estimation. CoRR **abs/1801.04062** (2018), http://arxiv.org/abs/1801.04062
6. Bronchain, O., Hendrickx, J.M., Massart, C., Olshevsky, A., Standaert, F.: Leakage certification revisited: Bounding model errors in side-channel security evaluations. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11692, pp. 713–737. Springer (2019)
7. Bundesamt für Sicherheit in der Informationstechnik:
8. de Chérisey, E., Guilley, S., Rioul, O., Piantanida, P.: Best information is most successful mutual information and success rate in side-channel analysis. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**(2), 49–79 (2019). https://doi.org/10.13154/tches.v2019.i2.49-79
9. Common Criteria: Common criteria v3.1 release 4 (2020), http://www.commoncriteriaportal.org/cc/
10. Cristiani, V., Lecomte, M., Maurine, P.: Leakage assessment through neural estimation of the mutual information. In: Zhou, J., Conti, M., Ahmed, C.M., Au, M.H., Batina, L., Li, Z., Lin, J., Losiouk, E., Luo, B., Majumdar, S., Meng, W., Ochoa, M., Picek, S., Portokalidis, G., Wang, C., Zhang, K. (eds.) Applied Cryptography and Network Security Workshops - ACNS 2020 Satellite Workshops, AIBlock, AIHWS, AIoTS, Cloud S&P, SCI, SecMT, and SiMLA, Rome, Italy, October 19-22, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12418, pp. 144–162. Springer (2020). https://doi.org/10.1007/978-3-030-61638-0_9, https://doi.org/10.1007/978-3-030-61638-0_9
11. Darbellay, G.A., Vajda, I.: Estimation of the information by an adaptive partitioning of the observation space. IEEE Trans. Inf. Theory **45**(4), 1315–1321 (1999). https://doi.org/10.1109/18.761290, https://doi.org/10.1109/18.761290
12. Duc, A., Faust, S., Standaert, F.: Making masking security proofs concrete (or how to evaluate the security of any leaking device), extended version. J. Cryptol. **32**(4), 1263–1297 (2019)
13. Durvaux, F., Standaert, F., Veyrat-Charvillon, N.: How to certify the leakage of a chip? In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8441, pp. 459–476. Springer (2014)
14. Gao, W., Kannan, S., Oh, S., Viswanath, P.: Estimating mutual information for discrete-continuous mixtures. In: Proceedings of the 31st International Conference

on Neural Information Processing Systems. p. 5988–5999. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
15. Gao, W., Oh, S., Viswanath, P.: Demystifying fixed k-nearest neighbor information estimators. CoRR **abs/1604.03006** (2016), http://arxiv.org/abs/1604.03006
16. Gilbert Goodwill, B.J., Jaffe, J., Rohatgi, P., et al.: A testing methodology for side-channel resistance validation. In: NIST non-invasive attack testing workshop. vol. 7, pp. 115–136 (2011)
17. Grosso, V., Standaert, F.: Masking proofs are tight and how to exploit it in security evaluations. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2018. vol. 10821, pp. 385–412. Springer (2018)
18. Györfi, L., van der Meulen, E.C.: Density-free convergence properties of various estimators of entropy. Computational Statistics and Data Analysis **5**(4), 425–436 (1987). https://doi.org/10.1016/0167-9473(87)90065-X
19. Hall, P., Morton, S.: On the estimation of entropy. Annals of the Institute of Statistical Mathematics **45**, 69–88 (02 1993). https://doi.org/10.1007/BF00773669
20. Heuser, A., Rioul, O., Guilley, S.: Good is not good enough - deriving optimal distinguishers from communication theory. In: Batina, L., Robshaw, M. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8731, pp. 55–74. Springer (2014)
21. Information Technology Laboratory,NIST: Security Requirements for Cryptographic Modules. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf
22. ISO/IEC: Testing methods for the mitigation of non-invasive attack classes against cryptographic modules. https://www.iso.org/obp/ui/#iso:std:iso-iec:17825:ed-1:v1:en (2016)
23. Kouiroukidis, N., Evangelidis, G.: The effects of dimensionality curse in high dimensional knn search. In: 2011 15th Panhellenic Conference on Informatics. pp. 41–45. IEEE (2011)
24. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating mutual information. Phys Rev E Stat Nonlin Soft Matter Phys **69**, pp. 066138 (07 2004). https://doi.org/10.1103/PhysRevE.69.066138
25. L. F. Kozachenko, N.N.L.: Sample estimate of the entropy of a random vector. Problems in Information Transmission **23** (1987)
26. Martin, D.P., Mather, L., Oswald, E., Stam, M.: Characterisation and estimation of the key rank distribution in the context of side channel evaluations. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 548–572 (2016)
27. Massey Jr, F.J.: The kolmogorov-smirnov test for goodness of fit. Journal of the American statistical Association **46**(253), 68–78 (1951)
28. Masure, L., Cassiers, G., Hendrickx, J., Standaert, F.X.: Information bounds and convergence rates for side-channel security evaluators. Cryptology ePrint Archive, Paper 2022/490 (2022), https://eprint.iacr.org/2022/490
29. McAllester, D., Stratos, K.: Formal limitations on the measurement of mutual information. In: Chiappa, S., Calandra, R. (eds.) The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]. Proceedings of Machine Learning Research, vol. 108, pp.

875–884. PMLR (2020), http://proceedings.mlr.press/v108/mcallester20a.html
30. McCann, D., Oswald, E., Whitnall, C.: Towards practical tools for side channel aware software engineering: 'grey box' modelling for instruction leakages. In: Kirda, E., Ristenpart, T. (eds.) 26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017. pp. 199–216. USENIX Association (2017)
31. Nair, C., Prabhakar, B., Shah, D.: On entropy for mixtures of discrete and continuous variables. arXiv preprint cs/0607075 (2006)
32. Picek, S., Heuser, A., Jovic, A., Bhasin, S., Regazzoni, F.: The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 209–237 (2019)
33. 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)
34. Prouff, E., Rivain, M., Bevan, R.: Statistical analysis of second order differential power analysis. IEEE Trans. Computers **58**(6), 799–811 (2009). https://doi.org/10.1109/TC.2009.15
35. Rajani, N., McArdle, K., Dhillon, I.S.: Parallel k nearest neighbor graph construction using tree-based data structures. In: 1st high performance graph mining workshop, sydney, 10 august 2015. Barcelona Supercomputing Center (2015)
36. Renauld, M., Standaert, F.X., Veyrat-Charvillon, N., Kamel, D., Flandre, D.: A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In: EUROCRYPT. pp. 109–128 (2011)
37. Roussas, G.G.: Chapter 6 - transformation of random variables. In: Roussas, G.G. (ed.) An Introduction to Probability and Statistical Inference (Second Edition), pp. 207–243. Academic Press, Boston, second edition edn. (2015). https://doi.org/10.1016/B978-0-12-800114-1.00006-8
38. SOG-IS: Application of attack potential to smartcards and similar devices (2019), https://www.sogis.eu/documents/cc/domains/sc/JIL-Application-of-Attack-Potential-to-Smartcards-v3-0.pdf
39. SOG-IS: Attack methods for smartcards and similar devices (2020)
40. Standaert, F., Gierlichs, B., Verbauwhede, I.: Partition vs. comparison side-channel distinguishers: An empirical evaluation of statistical tests for univariate side-channel attacks against two unprotected CMOS devices. In: Lee, P.J., Cheon, J.H. (eds.) Information Security and Cryptology - ICISC 2008, 11th International Conference, Seoul, Korea, December 3-5, 2008, Revised Selected Papers. Lecture Notes in Computer Science, vol. 5461, pp. 253–267. Springer (2008)
41. Thomas M. Cover, J.A.T.: Elements of Information Theory. Wiley (2005)