IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2023, No. 2, pp. 310–357. DOI:10.46586/tches.v2023.i2.310-357

Conditional Variational AutoEncoder based on Stochastic Attacks

Gabriel Zaid[1], Lilian Bossuet[2], Mathieu Carbone[1], Amaury Habrard[2,3] and Alexandre Venelli[4]

[1] Thales ITSEF, Toulouse, France, firstname.lastname@thalesgroup.com
[2] Univ Lyon, UJM-Saint-Etienne, CNRS Laboratoire Hubert Curien UMR 5516 F-42023, Saint-Etienne, France, firstname.lastname@univ-st-etienne.fr
[3] Institut Universitaire de France (IUF), Paris, France
[4] NXP Semiconductors, France, firstname.lastname@nxp.com

Abstract. Over the recent years, the cryptanalysis community leveraged the potential of research on Deep Learning to enhance attacks. In particular, several studies have recently highlighted the benefits of Deep Learning based Side-Channel Attacks (DLSCA) to target real-world cryptographic implementations. While this new research area on applied cryptography provides impressive result to recover a secret key even when countermeasures are implemented (e.g. desynchronization, masking schemes), the lack of theoretical results make the construction of appropriate and powerful models a notoriously hard problem. This can be problematic during an evaluation process where a security bound is required. In this work, we propose the first solution that bridges DL and SCA in order to get this security bound. Based on theoretical results, we develop the first Machine Learning generative model, called *Conditional Variational AutoEncoder based on Stochastic Attacks* (cVAE-SA), designed from the well-known *Stochastic Attacks*, that have been introduced by Schindler *et al.* in 2005. This model reduces the black-box property of DL and eases the architecture design for every real-world crypto-system as we define theoretical complexity bounds which only depend on the dimension of the (reduced) trace and the targeting variable over $\mathbb{F}_2^n$. We validate our theoretical proposition through simulations and public datasets on a wide range of use cases, including multi-task learning, curse of dimensionality and masking scheme.

Keywords: Side-Channel Attacks · Deep Learning · Generative Models · Discriminative Models · Stochastic Attacks · Variational AutoEncoder

1 Introduction

Context. Side-Channel Analysis (SCA) is a class of cryptographic attack in which an evaluator tries to exploit the vulnerabilities of a real-word crypto-system for key recovery by analyzing its physical characteristics via side-channel traces like power consumption or electromagnetic emissions. During the execution of an algorithm into a crypto-system, side-channel traces record the intermediate variable being processed. A *sensitive value* denotes an intermediate variable that depends on small pieces of the secret key, namely *subkeys*. Side-channel observables related to sensitive variables are referred to as *traces* and can be exploited in order to recover secret subkeys. One of the most powerful types of SCA, referred to as *profiled* SCA, is defined as a two-stage process. The underlying problem solved by this type of SCA relies on the classification task based on estimation of conditional (class-oriented) *probability distributions* related to each secret subkey. The first profiled SCA so-called *template attacks* was introduced by [CRR03] considering that

Received: 2022-10-15 Accepted: 2022-12-15 Published: 2023-03-06

a real leakage model coincides exactly with the deterministic part of the leakage and Gaussian noise assumption. Then, Schindler *et al.* proposed the so-called *stochastic attacks* that refine the approximation of real leakage model [SLP05] characterizing the leakage as a pseudo-boolean function over a monomial basis. Both approaches constitute the classical profiled SCA in the current state-of-the-art and are based on the construction of a *generative model*.

Two separated worlds. Widely developed in the Machine Learning field, a *generative* model has the particularity of generating new data from estimated *Probability Density Functions* (PDFs) (which are close to the real unknown one) by a sampling approach. Contrary to the classical profiled SCA, the state-of-the-art of Deep Learning based Side-Channel Analysis (DLSCA) only considers the *discriminative* models for key recovery. This solution differs from the generative model as it learns a direct map from traces to classes while the classical profiled SCA model the PDF of individual classes. Following [Vap98], the discriminative approach "*should solve the classification problem directly and never solve a more general problem as an intermediate step.*". Hence, generative models solve a more general problem than just guessing which subkey leaks the most. However, while the classical profiled SCA (*i.e.* generative approach) can be easily justified from a theoretical point of view, the discriminative DLSCA models, provided by the state-of-the-art, are very difficult to design and interpret. While such problematic might not be of interest from an adversary's point of view, the lack of interpretability and explainability is a real challenge for the security/evaluation laboratories. Indeed, to provide a relevant security assessment, the evaluator has to justify all the choices she/he has made to assess the robustness of a target of evaluation. Those choices lead the evaluator to define if the targeted device reaches a given security level. Therefore, finding a Machine Learning model that can be easily designed to provide a security bound is crucial. In [MCHS22], Masure *et al.* reduce the gap between the discriminative Logistic Regression models and the (pooled) Gaussian templates. However, there is a lack of intuitions about the decision-making process, even against unprotected implementations. In particular, one may wonder how a (non-)linear neural network should be designed in order to correctly estimate a suited decision boundary from a given set of traces. Indeed, even if a wide-range of architectures have been studied in DLSCA context (e.g. fully-connected neural networks [PCP20, PHJ+18], ResNets [JZHY20], transformer neural network [HSAM22], Support Vector Machine (SVM) [PHJ+18], Random Forest [PHJ+18]), there are no practical rules to construct them due to a lack of understanding between DL and SCA paradigms. Consequently, the discriminative models force the evaluator to consider the DLSCA approach as black-box tools which is problematic from an evaluation perspective. Thus, bridging the gap between DL and SCA is essential to define the limitations of the DLSCA and provide clear improvements in this field. From this new starting point, an evaluator can bring additional DL features that reduce the limitations provided by the classical profiled SCA approaches while preserving the interpretability and the explainability provided by the generative models.

Contributions. This paper falls into the class of works on machine-learning based cryptanalysis targeting real-world crypto-systems. In particular, we bridge DL and SCA paradigms by proposing the very first generative architecture which is based on theoretical results provided by stochastic attacks [SLP05]. This new model clarifies the links between DL and classical SCA issues (*i.e.* dimensionality reduction, needs of synchronization, higher-order attacks, multi-task learning). It represents a seminal contribution for further investigations and developments, in particular to get a general security bound of a targeted device. The contributions of our work can be summarized as follows:

- We establish the first link between DLSCA models and classical profiled SCA. By bridging both techniques, each field can benefit from the advantages of the other.

Explainability of classical SCA can be transferred to DLSCA that was considered more or less as a black-box tool[a].

- We propose the Conditional Variational AutoEncoder based Stochastic Attacks (cVAE-SA) as a new neural network architecture that lies between stochastic attacks and DLSCA. Our models benefit from the theoretical aspects of stochastic attacks, as well as their ability to estimate and reconstruct the targeted leakage models. This analogy is helpful to ease the construction of the neural network as well as its interpretation.

- We propose a full contextualization of the cVAE optimization process in the SCA field.

- Thanks to its analogy with the stochastic attacks, we define some theoretical bounds related to the neural network complexity. It suggests that shallow neural networks can be sufficient to exploit the sensitive information induced in a trace. This result is in accordance with the Universal Approximation Theorem [Pin99].

- We develop a new key recovery strategy based on similarity measure that allows an evaluator to specifically choose which samples the model should target to retrieve the sensitive information. This results in a more flexible solution than classical profiled side-channel attacks.

- We validate all our theoretical results through a wide range of use cases including the following challenges in SCA context namely multi-task learning, curse of dimensionality, targeting masking scheme.

- Through a detailed experimental comparison of our cVAE-SA proposition with classical profiled attacks (*i.e.* template and stochastic attacks) as well as multiple DLSCA models, we highlight the benefits and the drawbacks of cVAE-SA. This results in a perspective about a new typology of models specific to the SCA context.

This proposition opens-up further research directions where improvements from both fields could be further combined for enhancing the attack efficiency as well as the explainability of the results. All these experiments can be reproduced through a GitHub repository[b].

Paper Organization. This work is organized as follows: Sec.2 contrasts the related works in DLSCA, which is based on discriminative approach, with the generative approach we introduce in this paper. This section is then concluded by a general overview of the main results of this work. In Sec.3, a new neural network architecture based on stochastic attacks is proposed and, a detailed description of the optimization process as well as the key recovery phase is provided. Then, Sec.4 investigates the benefits of the cVAE-SA from an interpretability and explicability perspective while validating all the theoretical observations. Sec.5 illustrates the benefits and the limitations of the cVAE-SA in comparison with traditional approaches through experimental results. Finally, Sec.6 discusses about the benefits and the limitations of the contribution while introduces some new perspectives to consider as future works.

[a] Some works reduce this gap from an optimization perspective [MDP19, ZZN+20, ZBD+20, ISUH21, IUH22].

[b] https://github.com/gabzai/Conditional-Variational-Autoencoder-based-Stochastic-Attacks

2 Preliminaries

2.1 Notation & terminology

Basics on probability theory. Let calligraphic letters $\mathcal{X}$ denote sets such that, if $\mathcal{X}$ is finite, its cardinality, denoted $|\mathcal{X}|$, defines its number of elements. The corresponding capital letters X (resp. bold capital letters) denote random variables (resp. random vectors $\mathbf{T}$). The lowercase x (resp. $\mathbf{t}$) denote the realization of X (resp. $\mathbf{T}$). The i^{th} entry of a vector $\mathbf{T}$ is defined as $\mathbf{T}[i]$.

The probability of observing an event X is denoted by $\mathsf{Pr}[X]$ such that a conditional probability of observing an event X knowing an event Y is denoted $\mathsf{Pr}[X|Y]$. In the rest of this paper, a conditional probability, which approximates the true unknown $\mathsf{Pr}[X|Y]$, will be denoted as a δ-parametric conditional probability $\mathsf{Pr}[X|Y,\delta]$. The moments of a random variable X are quantities providing information about the shape and location of its *Probability Mass Function* (discrete case) or its *Probability Density Function* (continuous case). $\mathbb{E}[X]$ is used to denote the *expected value* of a random variable X and $\mathbb{E}_{X\sim\mathcal{D}}$ defines under which probability distribution it is computed. In addition, the second central moment, also known as the *variance*, of a random variable X is defined as $\mathbb{V}[X]$. The symbol $\mathbb{V}_X[f(X)]$ (resp. $\mathbb{E}_X[f(X)]$) denotes the variance (resp. expected value) of a function f related to the random variable X, over the distribution of X. The *standard deviation* of a random variable X, denoted σ_X, is defined as the square root of its variance.

A side-channel measurement will be constructed as a random vector $\mathbf{T} \in \mathbb{R}^D$ where D defines the dimension of the related trace. The targeted sensitive variable, denoted $Y = f(X, k^*)$, depends on a cryptographic primitive $f : \mathcal{X} \times \mathcal{K} \to \mathbb{F}_2^n$, a public variable $X \in \mathcal{X}$ (e.g. plaintext or ciphertext) and a part of the secret key $k^* \in \mathcal{K}$ (e.g. byte) that the evaluator tries to retrieve. We define $\mathbf{X} \sim \mathcal{N}_D(\boldsymbol{\mu}, \Sigma)$ to denote a D-dimensional random vector $\mathbf{X}$ that follows a multivariate Gaussian distribution of parameters $\boldsymbol{\mu} \in \mathbb{R}^D$ and $\Sigma \in \mathcal{M}_{D,D}(\mathbb{R})$. In the rest of this paper, $\Sigma_{\mathbf{X}} = Cov(\mathbf{X}, \mathbf{X})$ (resp. $\boldsymbol{\mu}_{\mathbf{X}}$) denotes the covariance matrix (resp. mean) of a random variable $\mathbf{X}$. Given p_X and q_X two probability distributions on $\mathcal{X}$, the Kullback-Leibler (KL) divergence measures how p_X differs from q_X such that:

$$\mathcal{D}_{\text{KL}}\left(p_X||q_X\right) = \sum_{x\in\mathcal{X}} p_X\left[x\right] \log\left(\frac{p_X\left[x\right]}{q_X\left[x\right]}\right).$$

Usually, p_X denotes the measured probability distribution while q_X defines the theoretical model. The KL-divergence is always non-negative and equals zero if and only if $p_X = q_X$.

SCA terminology. SCA usually apply a divide-and-conquer strategy which consists in separately recovering different parts of the N-bit (global) secret key $k^* = \|_{i=1}^{\frac{N}{n}} k_i^*$ considering n-bit subkeys $k_i^* \in \mathcal{K}$. For the rest of this paper, we will consider only attacking a subkey (*i.e.* $n = 8$), hence using k^* instead of k_i^* and referencing subkey as key in the rest of the paper. Given a variable Y and an independent noise $\mathbf{Z}$, a trace $\mathbf{T}$ is a D-dimensional random vector $\{\mathbf{T}[0], \ldots, \mathbf{T}[D-1]\}$ where $\mathbf{T}[i]$ represents the leakage of time sample i (for $0 \leq i < D$) satisfying the Gaussian Indepedent Noise Assumption[c]:

$$\mathbf{T} = \psi(Y) + \mathbf{Z}, \tag{1}$$

where $\psi : \mathbb{F}_2^n \to \mathbb{R}$ is a pseudo-boolean function [Car10] mapping a n-bit intermediate value Y which is generated from a cryptographic primitive $f : \mathcal{X} \times \mathcal{K} \to \mathbb{F}_2^n$. The latter corresponds to the deterministic part of the trace. Let $\mathbf{Z} \sim \mathcal{N}_D(\boldsymbol{\mu}, \Sigma)$ correspond to the noise which is characterized by a multivariate Gaussian distribution parameterized by an unknown pair $(\boldsymbol{\mu}, \Sigma)$.

[c] This means that the Gaussian noise $\mathbf{Z}$ is independent of the variable Y.

Explainability & Interpretability. In this work, the interpretation refers to the ability of the evaluator to clearly identify each operation induced by the generative/discriminative model's layers during the decision-making process. For example, following the classical profiled SCA scenarios (*i.e.* stochastic attacks, template attacks), the evaluator may wonder which leakage model is extracted for each point of interest, how does the noisy part of a trace is characterized, how does the dimensionality reduction is processed or even, how does the statistical model should be designed. This paper tackles those problems in a DLSCA perspective in order to ease the use of those techniques in side-channel context.

2.2 Related works & limitations of discriminative models

Related works. Typically, to perform a DLSCA attack the evaluator considers a discriminative approach which models the conditional posterior probabilities $\mathsf{Pr}[Y|\mathbf{T}]$ in order to discriminate and pick the most likely hypothetical candidate Y (*i.e.* sensitive information) given a trace $\mathbf{T}$. A discriminative model estimates a ϕ-parametric probability conditional distribution $\mathsf{Pr}[Y|\mathbf{T},\phi]$ that is as similar as possible to the true unknown joint probability distribution $\mathsf{Pr}[Y|\mathbf{T}]$. This approach is beneficial for directly solving a classification problem without modeling unnecessary information and thus, mitigating the impact of some countermeasures such as the desynchronization effect [CDP17a, ZBHV19, Mag19, HSAM22].This reason leads the side-channel community to investigate the DL approaches to improve the profiled SCA [CDP17a, KPH+19, ZBHV19, BPS+20] relying on discriminative approach. While [WPP22] combines a DL dimensionality reduction method with template attacks as an alternative to the *Principal Component Analysis* [APSQ06], the *Linear Discriminant Analysis* [BGH+15] or the *Kernel Discriminant Analysis* [CDP17b], all the end-to-end DLSCA models proposed in the state-of-the-art are based on the discriminative approach (e.g. fully-connected neural networks [MZ13, MHM14, Wei20], ResNets [ZS19, JZHY20, GJS20, MS21], RNNs [LLY+20], transformer neural network [HSAM22], attention mechanisms [LZC+21]). However, due to the lack of theoretical results, the discriminative models can be seen as black-box tools, and the design of models can be a real challenge even against unprotected cryptographic implementations. To reduce this issue, some solutions which automatically tune model hyperparameters have been investigated [MPP16, BPS+20, WPP20, PRA20, RWPP21, YAGF21] but the related process is time-consuming, and the range of the hyperparameters' values is randomly bounded such that a poor design of the model can be highly impacted by underfitting/overfitting issues. This paper reduces this issue by providing the first DL model based on SCA theoretical result in order to make the construction phase easier.

Generative approach. An alternative solution consists in considering a probabilistic generative approach which captures the interactions between all the variables considered by the resulting learning algorithm. To comply with this technical specification, this strategy builds a model that estimates the probability distribution of the traces. To fit with SCA context[d], the conditional probability distribution, $\mathsf{Pr}[\mathbf{T}|Y]$, has to be estimated such that, afterwards, the Bayes' theorem can be computed in order to retrieve the conditional posterior probabilities $\mathsf{Pr}[Y|\mathbf{T}]$ and pick the most likely label Y. More concretely, a generative model can be viewed as an estimation of a Θ-parametric conditional distribution $\mathsf{Pr}[\mathbf{T}|Y,\Theta]$ that is as similar as possible to the true unknown conditional distribution $\mathsf{Pr}[\mathbf{T}|Y]$. The classical profiled SCAs, such as the stochastic attacks [SLP05], follow this approach by building a model, *i.e. leakage model*, that estimates the class-conditional probability distributions (*i.e.* $\mathsf{Pr}[\mathbf{T}|Y]$) for each possible value of a sensitive intermediate

[d] In profiled SCA, the profiling phase suggests the estimation of the PDFs $\mathsf{Pr}[\mathbf{T}|Y] = \epsilon \cdot \mathsf{Pr}[Y|\mathbf{T}]$ where ϵ denotes a constant independent of the secret key. Thus, in SCA, we can hope that discriminative and generative models perform almost similarly. This observation is supported by the experimental results provided in Tab.4.

variable. One benefit of this method is the ability to explain and interpret the result provided by the model. The following section summarizes the stochastic attacks [SLP05] introduced by Schindler *et al.* as well as our contribution that we detail in Sec.3.

2.3 General description of this work

A short description of stochastic attacks [SLP05]. Given a trace $\mathbf{T}$ such that its i^{th} time sample can be defined as $\mathbf{T}[i] = \psi_i(f(X, k^*)) + \mathbf{Z}[i]$, the goal of the stochastic attack is to find an approximation of the leakage model, denoted $\hat{\psi}_i$, as close as possible to the true unknown ψ_i. As ψ_i is assumed to be a pseudo-boolean function, ψ_i can be viewed as a linear combination of monomial basis' vectors $u \in \mathbb{F}_2^n$ [Car10]. Hence, there exists a set of real coefficients $(\alpha_u)_{u\in\mathbb{F}_2^n}$ such that, for a sensitive intermediate value $Y \in \mathbb{F}_2^n$, the *leakage model* (see Eq.1) is redefined as:

$$\hat{\psi}_{i,\alpha}(Y) = \sum_{u=(u[0],\ldots,u[n-1])\in\mathbb{F}_2^n} \alpha_u[i] \cdot Y^u, \tag{2}$$

where Y^u denotes the *monomial basis* and characterizes the conjunction of all bits of Y such that $Y^u = \prod_{j=0}^{n-1} Y[j]^{u[j]}$ where $Y[j] \in \mathbb{F}_2$ defines the j^{th} bit of Y and the power notation is simply $Y[j]^0 = 1$ and $Y[j]^1 = Y[j]$. In other words, ψ_i can be approximated as a multivariate polynomial in the bit-coordinate $Y[j]$ with coefficients in $\mathbb{R}$. The degree d (s.t. $d \leq n$) of such monomial is defined as the maximal number of bits' interaction induced in $\hat{\psi}_{i,\alpha}(Y)$. In particular, this degree d can be viewed as logical operators (e.g. AND or XOR). The related subspace is denoted by $\mathcal{F}_{d+1}$. For the profiling phase, the stochastic attack mechanism consists firstly in choosing the degree d of the pseudo-boolean function $\hat{\psi}_\alpha$, and then in estimating the leakage model related to the targeted device. Given a set of N_p labeled traces $\mathcal{I}_p = \{(\mathbf{t}_0, y_0), \ldots, (\mathbf{t}_{N_p-1}, y_{N_p-1})\}$, the evaluator estimates the leakage model $(\hat{\psi}_{i,\alpha}(Y))_{Y\in\mathbb{F}_2^n}$ by finding the best set of coefficients $(\alpha_u[i])_{u\in\mathbb{F}_2^n}$ through the application of the *ordinary least squares* (OLS) method. The set of coefficients $(\alpha_u[i])_{u\in\mathbb{F}_2^n}$ which minimizes the OLS are called the *OLS estimator* for ψ. More details on how to practically implement stochastic attacks can be found in [CK15]. While the basis choice is essential for efficient profiling phase [MOW17], *i.e.* having a good approximation of the *leakage model*, the application of *gradient descent* method for minimizing the OLS method is an interesting alternative to the classical approach (see [SLP05, Eq.13]) and will be explored in next sections. This leads to get a better intuition into how a DL model should be designed in order to extract the sensitive information and results in a more flexible solution during the exploitation phase.

A new generative strategy in DLSCA. In SCA context, we want to explicitly compute an approximation of the true unknown conditional probability distribution $\mathsf{Pr}[\mathbf{T}|Y]$ in order to retrieve the secret key that is manipulated by the targeted real-world crypto-system. In 2014, Kingma and Welling introduced the *Variational AutoEncoder* (VAE) [KW14] as a solution to this issue outside of the SCA context. Ever since the seminal work has been widely applied in various fields (e.g. face generation [KW14, KWKT15], handwritten digits [KW14], objects [KWKT15]), we propose to contextualize conditional variational autoencoder into side-channel analysis in order to give a new perspective for generative models. In this paper, we develop a new usage of variational autoencoders for DLSCA and we present our main contribution: the *Conditional Variational AutoEncoder based on Stochastic Attacks* (cVAE-SA). This work can be decomposed into three parts (see Fig.1):

1. First, a description of the cVAE-SA structure is proposed. In particular, a theoretical link is highlighted with the stochastic attacks in order to model a Θ-parametric

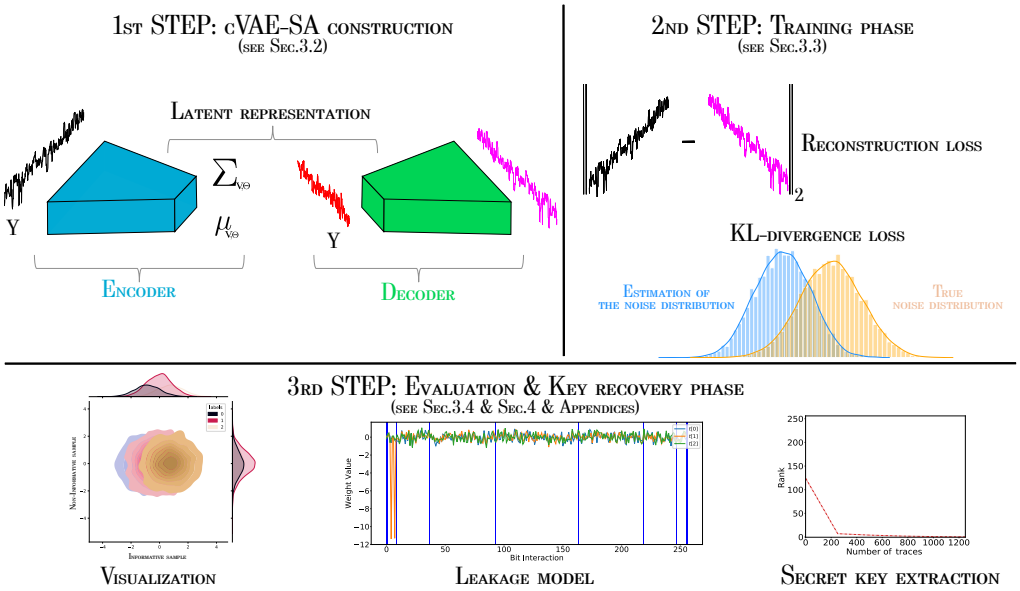


Figure 1: Overview of this work.

conditional distribution $\mathsf{Pr}[\mathbf{T}|Y,\Theta]$ through the design of two distinct parts referred to the *encoder* and *decoder*. In outline, the encoder approximates the parameters $\boldsymbol{\mu}$ and Σ which characterize the noise $\mathbf{Z}$ included in a trace (see Eq.1).Then, the decoder is defined to generate a synthetic trace from a variable which follows $\mathcal{N}_D(\boldsymbol{\mu},\Sigma)$, and an approximation of the deterministic part $\psi(Y)$ defined in Eq.1. The description of these entities is detailed in Sec.3.2. This part is helpful from an evaluation perspective to reduce the explainability/interpretability issues mentioned in Sec.1. In particular, it clarifies the operations induced in each layer of the cVAE-SA model.

2. Once the cVAE-SA is designed, it is automatically configured over a set of training traces in order to estimate the Θ-parametric conditional distribution $\mathsf{Pr}[\mathbf{T}|Y,\Theta]$ that should be as similar as possible to the true unknown conditional distribution $\mathsf{Pr}[\mathbf{T}|Y]$. To obtain such model, a combination of a reconstruction and a KL-divergence losses is conducted in order to find the trainable parameters that fit the most with the true unknown solution. While the reconstruction loss is used to measure the similarity error (in term of Euclidean distance) between a synthetic and a real trace, the KL-divergence loss penalizes the cVAE-SA if the parameters $\boldsymbol{\mu}$ and Σ do not fit with the expected noise distribution. The combination of those losses is widely known as the ELBO loss [KW14]. The justification about the use of these losses is provided in Sec.3.3.

3. Finally, based on this configured and trained model, the evaluator can compute the maximum likelihood over a set of attack traces in order to retrieve the most likely subkey candidate over $\mathbb{F}_2^n$ such that $n = 8$. A detailed modus operandi is provided in Sec.3.4. In addition, multiple visualization techniques can be considered in order to better understand the extracted leakage model as well as the latent representation. Those visualization tools are introduced in Sec.4 in order to validate the stated theoretical results. Further investigations have also been conducted in App.A and App.B to verify the theoretical statements.

3 Conditional Variational AutoEncoder based on Stochastic Attacks

Through this section, we explain the link between generative DL models and classical profiled SCA by building a new type of VAE. Sec.3.1 introduces the problem we want to solve and proposes a first link with SCA. Sec.3.2 explains our architecture and the theoretical link with the work provided by Schindler *et al.* [SLP05], known as the stochastic attacks. Then, Sec.3.3 describes the training process of cVAE-SA and the relation with similarity measures. Finally, Sec.3.4 describes the attack phase and the theoretical architecture complexity bounds.

3.1 Generative latent variable models

After a general introduction of the Conditional VAE (cVAE) [SLY15], we contextualize this solution into SCA in order to give a new perspective for DL generative models. Supported by theoretical aspects of stochastic attacks, this new approach can be considered as an alternative to classical discriminative models often used in DLSCA.

Problem statement. The cVAE aims at modeling a Θ-parametric conditional distribution $\mathsf{Pr}[\mathbf{T}|Y,\Theta]$ from two random variables $\mathbf{T} \in \mathbb{R}^D$ and $Y \in \mathbb{F}_2^n$. Suppose that a trace $\mathbf{T} \in \mathbb{R}^D$ is acquired by assuming that all the time samples are sequentially generated such that its assigned label only depends on a small set of time samples (*i.e.* PoIs). As the cVAE is a *latent variable model*, which suggests that the variability in the traces given a label Y can be captured by a small finite set of time samples, its applicability in the SCA context fits well. By designing such models for performing SCA, we thus want to capture the interactions between the time samples *via* the characterization of a latent space $\mathcal{V}$. In particular, a Θ-parametric latent variable model F_Θ, providing a Θ-parametric conditional distribution $\mathsf{Pr}[\mathbf{T}|Y,\Theta]$, is representative of the true unknown conditional distribution $\mathsf{Pr}[\mathbf{T}|Y]$, for every trace $\mathbf{T}$ and every given sensitive variable Y, if there is a representation of compressed data $\mathbf{V} \in \mathcal{V}$, also known as latent space representation, such that the marginal distribution is given by:

$$\mathsf{Pr}\left[\mathbf{T}|Y,\Theta\right] = \int_{\mathbf{v}\in\mathcal{V}} \mathsf{Pr}\left[\mathbf{T}|Y,\mathbf{v},\Theta\right] \mathsf{Pr}\left[\mathbf{v}|Y\right] d\mathbf{v}, \tag{3}$$

where $\mathbf{v}$ is the realization of a random variable $\mathbf{V}$ in a $D^{'}$-dimensional space $\mathcal{V}$, with a probability $\mathsf{Pr}[\mathbf{V}=\mathbf{v}]$ defined over $\mathcal{V}$, and $\mathsf{Pr}[\mathbf{V}=\mathbf{v}|Y]$ denotes the probability of observing $\mathbf{v}$ over the latent space $\mathcal{V}$ knowing Y.

Intractability. However, Eq.3 is unfortunately intractable as it should be computed for every latent representation induced by the latent space $\mathcal{V}$. Thus, the following part of the section proposes solutions to circumvent this issue. Hopefully, $\mathsf{Pr}[\mathbf{T}|Y,\Theta]$ may still be efficiently approximated thanks to the *Monte-Carlo* method. Hence, for a large number of samples $\{\mathbf{v}_0,\ldots,\mathbf{v}_{N_v}\}$, a trace $\mathbf{T} \in \mathbb{R}^D$ and a label $Y \in \mathbb{F}_2^n$, we can compute an estimation of $\mathsf{Pr}[\mathbf{T}|Y]$. As a consequence, for a given label Y and a latent variable $\mathbf{V} \in \mathbb{R}^{D^{'}}$, we can build a neural network that computes $\mathsf{Pr}[\mathbf{T}|Y,\mathbf{V},\Theta]$. This model, denoted $F_\Theta^{(dec)} : \mathbb{R}^{D^{'}} \times \mathbb{F}_2^n \rightarrow \mathbb{R}^D$, is named *the decoder*.

Given a latent variable $\mathbf{V}$ and a sensitive variable $Y \in \mathbb{F}_2^n$, the decoder generates a new trace $\tilde{\mathbf{T}}$ as close as possible to the real observed trace $\mathbf{T}$. However, to perfectly construct a new set of D-dimensional traces from $F_\Theta^{(dec)}$, we have to compute latent space samples that are representative of the observed trace. To this end, we estimate the latent space $\mathcal{V}$ by approximating the following probability distribution $\mathsf{Pr}[\mathbf{V}|\mathbf{T},Y]$.

This probability is defined as $\mathsf{Pr}[\mathbf{V}|\mathbf{T},Y] = \frac{\mathsf{Pr}[\mathbf{T}|\mathbf{V},Y]\cdot\mathsf{Pr}[\mathbf{V}]}{\mathsf{Pr}[\mathbf{T}|Y]}$ and it is also intractable due to Eq.3. Consequently, a solution is to find a parametric model that approximates the true unknown posterior $\mathsf{Pr}[\mathbf{V}|\mathbf{T},Y]$. In statistics, the variational inference techniques can approximate such complex distributions. Given a trace $\mathbf{T} \in \mathbb{R}^D$ and a label $Y \in \mathbb{F}_2^n$, a Θ-parametric model can be constructed to estimate the latent space $\mathcal{V}$ such that the KL-divergence between the approximation and the targeted probability distribution $\mathsf{Pr}[\mathbf{V}|\mathbf{T},Y]$ is minimized. The Θ-parametric model, denoted $F_\Theta^{(enc)} : \mathbb{R}^D \times \mathbb{F}_2^n \to \mathbb{R}^{D'} \times \mathcal{M}_{D',D'}(\mathbb{R})$, is called *the encoder*. In the rest of this paper, we denote as $F_{\Theta,\phi}$ the resulted cVAE-SA such that, for a given trace $\mathbf{T}$, a given label Y and a function $g : \mathbb{R}^{D'} \times \mathcal{M}_{D',D'}(\mathbb{R}) \to \mathbb{R}^{D'}$, $F_{\Theta,\phi}(\mathbf{T},Y) = F_\phi^{(dec)} \circ (g(F_\Theta^{(enc)}(\mathbf{T},Y)),Y)$. Furthermore, as the aim of this paper is to bridge the DL and the classical profiled SCA, no particular focus will be proposed on dimensionality reduction techniques. Thus, the following part assumes that $D' = D$.

3.2 Latent space estimation and instances' generation

Through the description of the stochastic attack (see Sec.2.3), the evaluator can construct a conditional variational autoencoder adapted for the SCA context.

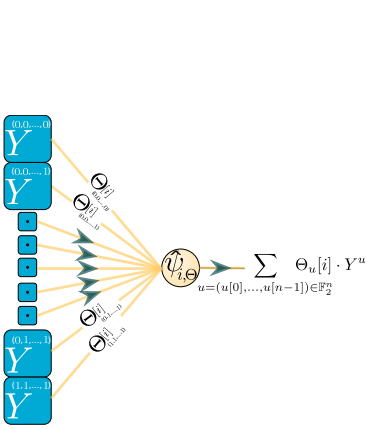
Encoder. As mentioned in Sec.3.1, the encoder models a neural network $F_\Theta^{(enc)} : \mathbb{R}^D \times \mathbb{F}_2^n \to \mathbb{R}^D \times \mathcal{M}_{D,D}(\mathbb{R})$ which returns an element in the latent space $\mathcal{V}$ of dimension D. This element should describe the behavior of the targeted crypto-system. In this regard, we design the encoder $F_\Theta^{(enc)}$ such that it characterizes the leakage model $\psi(Y)$ and the random part $\mathbf{Z}$ of a trace $\mathbf{T}$ in order to fit with the stochastic attack process. To build a suited encoder, the related neural network should follow the structure defined in Sec.2.3 in order to extract the maximum amount of relevant information from $\mathbf{T}$. First, the evaluator has to estimate the deterministic part of a trace $\mathbf{T}$ (*i.e.* leakage model ψ) that is defined by Eq.2. This modeling can be estimated by a fully-connected layer of D neurons such that each of them is linked with all elements of the monomial basis $(Y^u)_{u\in\mathbb{F}_2^n}$. Let $Y \in \mathbb{F}_2^n$ and $(Y^u)_{u\in\mathbb{F}_2^n}$ (resp. $\hat{\psi}_{i,\Theta}(Y)$) be the input (resp. output) of the i^{th} neuron such that:

$$\hat{\psi}_{i,\Theta}(Y) = \varrho\left(\sum_{u=(u[0],\ldots,u[n-1])\in\mathbb{F}_2^n} \Theta_u[i]\cdot Y^u\right),$$

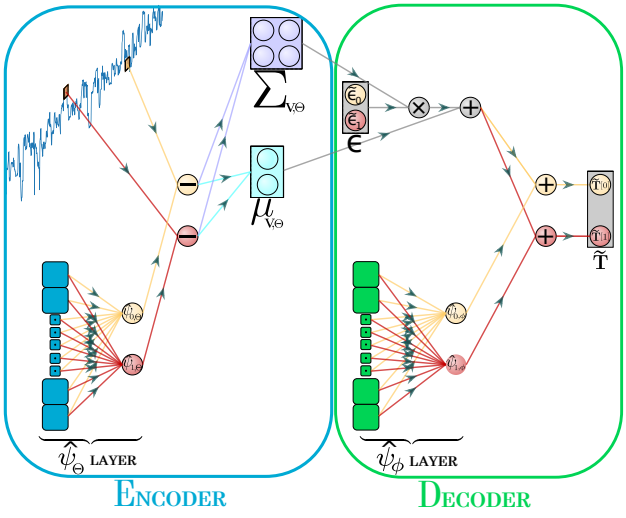
where $\varrho(.)$ is a function (linear or non-linear) and $\Theta \in \mathcal{M}_{1+\sum_{i=0}^{d}\binom{n}{i},D}(\mathbb{R})$ denotes the set of trainable parameters for a given degree d that characterizes the space $\mathcal{F}_{d+1}$. While the goal of our work is to reduce the gap between deep learning and classical profiled SCA, we define $\varrho(.)$ as the identity function[e] in order to satisfy $\hat{\psi}_\Theta[i] = \hat{\psi}_\alpha[i]$ and consider that the deterministic part of a trace at time sample i can be approximated by a single neuron (see Fig.2a). In the rest of this paper, this layer will be denoted as $\hat{\psi}_\Theta$ (see Fig.2b).

Once the noise-free part $\hat{\psi}_\Theta$ is estimated, the next step is to deeply characterize the noise part $\mathbf{Z}$ using traces and the neurons of $\hat{\psi}_\Theta$ layer. In the cVAE-SA, we choose to deliberately force the subtraction of the traces at time sample i and the i^{th} neuron of $\hat{\psi}_\Theta$ layer in order to fit with Eq.1. Then, the encoder $F_\Theta^{(enc)}$ is trained to return a Θ-parametric mean vector $\boldsymbol{\mu}_{\mathbf{V},\Theta} \in \mathbb{R}^D$, and a Θ-parametric covariance matrix $\Sigma_{\mathbf{V},\Theta} \in \mathcal{M}_{D,D}(\mathbb{R})$ that describes the multivariate Gaussian noise for a given trace $\mathbf{T}$. Those approximations respectively estimate $\boldsymbol{\mu}_\mathbf{V}$ and $\Sigma_\mathbf{V}$ that characterize the latent space $\mathcal{V}$. Thus, from these

[e] As this paper bridges DL with SCA, no further investigation has been conducted to assess the impact of non-linear functions due to the lack of interpretability it brings. In addition, as the combination of linear functions is a linear function, adding layers is not adapted to our context.



(a) Example of a $\hat{\psi}_\Theta$ layer with a single neuron such that the input is defined as the monomial basis $(Y^u)_{u\in\mathbb{F}_2^n}$ in $\mathcal{F}_9$.



(b) Overall architecture.

Figure 2: cVAE-SA structure.

parameters, the evaluator can compute[f] $\mathsf{Pr}[\mathbf{V}|\mathbf{T}, Y, \Theta]$. That is, $F_\Theta^{(enc)}$ extracts all the information induced in a trace such that the latent variable $\mathbf{V} \in \mathcal{V}$ is characterized by its noise part $\mathbf{Z}$.

Decoder. Once the encoder is constructed, the evaluator can capture the parameters $\boldsymbol{\mu}_{\mathbf{V},\Theta}$ and $\Sigma_{\mathbf{V},\Theta}$ which are needed to design the related Gaussian noise distribution $\mathcal{N}_D(\boldsymbol{\mu}_{\mathbf{V},\Theta}, \Sigma_{\mathbf{V},\Theta})$. From a new sample $\mathbf{V} \sim \mathcal{N}_D(\boldsymbol{\mu}_{\mathbf{V},\Theta}, \Sigma_{\mathbf{V},\Theta})$, the evaluator designs a decoder $F_\phi^{(dec)} : \mathbb{R}^D \times \mathbb{F}_2^n \to \mathbb{R}^D$ (see on the right part of Fig.2b) such that given a trace $\mathbf{T}$ and the subspace $\mathcal{F}_{d+1}$, containing all the pseudo-boolean functions of degree lower or equal to d, he wants to maximize the conditional probability distribution $\mathsf{Pr}[\mathbf{T}|Y, \mathbf{V}, \phi]$, *i.e.* building a new trace $\tilde{\mathbf{T}} \in \mathbb{R}^D$ as similar as possible to the related real trace $\mathbf{T}$ and defined as follows:

$$\tilde{\mathbf{T}}[i] = \underbrace{\sum_{u=(u[0],\ldots,u[n-1])\in\mathbb{F}_2^n} \phi_u[i] \cdot Y^u}_{\hat{\psi}_{i,\phi}} + \mathbf{V}[i]. \tag{4}$$

Note that a latent variable $\mathbf{V} \in \mathbb{R}^D$ is initially sampled from the prior distribution $\mathsf{Pr}[\mathbf{V}]$ such that the dimension of $\mathbf{V}$ should correspond with the dimension of the latent space estimated by the encoder. However, performing the training process in such configuration can be arduous. Indeed, during the training process, the backpropagation cannot be performed because the evaluator has to compute the gradient of the loss function with respect to samples (*i.e.* latent variable $\mathbf{V} \in \mathcal{V}$), which is inherently non-differentiable. To circumvent this issue, the reparametrization trick [KW14] proposes to rewrite $\mathbf{V}$ such that the derivative can be computed with respect to the parametric distributions (*i.e.* $\boldsymbol{\mu}_{\mathbf{V},\Theta}$ and $\Sigma_{\mathbf{V},\Theta}$) that are differentiable. Instead of generating samples from $\mathcal{N}_D(\boldsymbol{\mu}_{\mathbf{V},\Theta}, \Sigma_{\mathbf{V},\Theta})$, sampling is performed from $\epsilon \sim \mathcal{N}_D(\mathbf{0}, \mathbf{I}_D)$, followed by the computation of $\mathbf{V} = \boldsymbol{\mu}_{\mathbf{V},\Theta} + \Sigma_{\mathbf{V},\Theta}^{\frac{1}{2}} \times \epsilon$. This process is defined by the function $g : \mathbb{R}^{D'} \times \mathcal{M}_{D',D'}(\mathbb{R}) \to \mathbb{R}^{D'}$ in Sec.3.1.

[f] While the latent space characterizes the noise distribution defined in Eq.1, we can easily assume that $\mathsf{Pr}[\mathbf{V}|Y] = \mathsf{Pr}[\mathbf{V}]$ because $\mathbf{V}$ is independent of the label Y. Therefore, Eq.3 can be simplified and $\mathsf{Pr}[\mathbf{V}]$ follows a multivariate Gaussian distribution of parameters $(\boldsymbol{\mu}, \Sigma)$.

Once $\mathbf{V}$ is constructed, the evaluator has to approximate the deterministic part of the leakage model, namely $\hat{\psi}_\phi$. As already mentioned for the encoder, its estimation can be made with a fully-connected layer such that the input of size D is characterized by $(Y^u)_{u\in\mathbb{F}_2^n}$ for a given $Y \in \mathbb{F}_2^n$. Because the evaluator wants to characterize all the input time samples, the number of nodes in the $\hat{\psi}_\phi$ layer depends on the dimensionality of the latent space, *i.e.* dimension of $\mathcal{V}$ (see Fig.2b). Based on $\hat{\psi}_\phi$ and $\mathbf{V}$, the evaluator can then build a new trace $\tilde{\mathbf{T}}$ following Eq.4.

A discussion and some visualization methods are proposed in App.A in order to ease the understanding of the encoder and the decoder. Then, to adequately find the trainable parameters Θ and ϕ, the evaluator has to consider some learning metrics that aims at approximating $\mathsf{Pr}[\mathbf{T}|Y]$.

3.3 Similarity maximization

This section describes the optimization process from a side-channel perspective and introduces some simplifications that can be conducted thanks to the side-channel literature.

Introduction of the optimization problem. As defined in Sec.3.2, our generative model has to optimize a set of parameters ϕ and Θ in order to maximize the marginal log-likelihood $\log(\mathsf{Pr}[\mathbf{T}|Y,\phi])$.

Theorem 1. *For any choice of encoder $F_\Theta^{(enc)}$ and trainable parameters Θ, the conditional marginal log likelihood $\log(\mathsf{Pr}[\boldsymbol{T}|Y,\phi])$ can be defined as:*

$$\begin{aligned}\log\left(\mathsf{Pr}\left[\boldsymbol{T}|Y,\phi\right]\right) = \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\mathsf{Pr}\left[\boldsymbol{T},\mathbf{v}|Y,\phi\right]\right) - \log\left(\mathsf{Pr}\left[\mathbf{v}|\boldsymbol{T},Y,\Theta\right]\right)\right] \\ + \mathcal{D}_{KL}\left(\mathsf{Pr}\left[\boldsymbol{V}|\boldsymbol{T},Y,\Theta\right]||\mathsf{Pr}\left[\boldsymbol{V}|\boldsymbol{T},Y,\phi\right]\right).\end{aligned} \tag{5}$$

Proof. From Eq.3 and the work provided by Kingma and Welling [KW19, Sec.2.2], we can extend their result to the conditional marginal log likelihood $\log\left(\mathsf{Pr}[\mathbf{T}|Y,\phi]\right)$ as follows:

$$\begin{aligned}\log\left(\mathsf{Pr}\left[\mathbf{T}|Y,\phi\right]\right) &= \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\mathsf{Pr}\left[\mathbf{T}|Y,\mathbf{v},\phi\right]\right)\right] \\ &= \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\frac{\mathsf{Pr}\left[\mathbf{T},\mathbf{v}|Y,\phi\right]}{\mathsf{Pr}\left[\mathbf{v}|\mathbf{T},Y,\phi\right]}\right)\right] \\ &= \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\frac{\mathsf{Pr}\left[\mathbf{T},\mathbf{v}|Y,\phi\right]}{\mathsf{Pr}\left[\mathbf{v}|\mathbf{T},Y,\Theta\right]}\cdot\frac{\mathsf{Pr}\left[\mathbf{v}|\mathbf{T},Y,\Theta\right]}{\mathsf{Pr}\left[\mathbf{v}|\mathbf{T},Y,\phi\right]}\right)\right] \\ &= \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\frac{\mathsf{Pr}\left[\mathbf{T},\mathbf{v}|Y,\phi\right]}{\mathsf{Pr}\left[\mathbf{v}|\mathbf{T},Y,\Theta\right]}\right)\right] + \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\frac{\mathsf{Pr}\left[\mathbf{v}|\mathbf{T},Y,\Theta\right]}{\mathsf{Pr}\left[\mathbf{v}|\mathbf{T},Y,\phi\right]}\right)\right] \\ &= \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\mathsf{Pr}\left[\mathbf{T},\mathbf{v}|Y,\phi\right]\right) - \log\left(\mathsf{Pr}\left[\mathbf{v}|\mathbf{T},Y,\Theta\right]\right)\right] \\ &\quad + \mathcal{D}_{\mathrm{KL}}\left(\mathsf{Pr}\left[\mathbf{V}|\mathbf{T},Y,\Theta\right]||\mathsf{Pr}\left[\mathbf{V}|\mathbf{T},Y,\phi\right]\right).\end{aligned}$$

□

Unfortunately, due to the intractability of $\mathsf{Pr}[\mathbf{V}|\mathbf{T},Y,\phi]$ (see Sec.3.1), Eq.5 cannot be solved in practice. Hence, we have to define a function such that $\log(\mathsf{Pr}[\mathbf{T}|Y,\phi])$ can be approximated through an optimization algorithm. In [KW14], Kingma and Welling propose a variational lower bound on the marginal likelihood which was generalized by Sohn *et al.* on conditional marginal likelihood [SLY15].

Theorem 2. *[SLY15] For any choice of encoder $F_\Theta^{(enc)}$ and trainable parameters Θ, the variational lower bound of* $\log(\mathsf{Pr}[\boldsymbol{T}|Y,\phi])$ *is defined as:*

$$\begin{aligned}\log\left(\mathsf{Pr}\left[\boldsymbol{T}|Y,\phi\right]\right) \geq & -\mathcal{D}_{KL}\left(\mathsf{Pr}\left[\boldsymbol{V}|\boldsymbol{T},Y,\Theta\right]||\mathsf{Pr}\left[\boldsymbol{V}\right]\right) \\ & + \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\mathsf{Pr}\left[\boldsymbol{T}|Y,\mathbf{v},\phi\right]\right)\right].\end{aligned} \tag{6}$$

Proof. [SLY15]

$$\begin{aligned}\log\left(\mathsf{Pr}\left[\mathbf{T}|Y,\phi\right]\right) &= \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\mathsf{Pr}\left[\mathbf{T},\mathbf{v}|Y,\phi\right]\right) - \log\left(\mathsf{Pr}\left[\mathbf{v}|\mathbf{T},Y,\Theta\right]\right)\right] \\ &\quad + \mathcal{D}_{\mathrm{KL}}\left(\mathsf{Pr}\left[\mathbf{V}|\mathbf{T},Y,\Theta\right]||\mathsf{Pr}\left[\mathbf{V}|\mathbf{T},Y,\phi\right]\right) \\ &\geq \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\mathsf{Pr}\left[\mathbf{T},\mathbf{v}|Y,\phi\right]\right) - \log\left(\mathsf{Pr}\left[\mathbf{v}|\mathbf{T},Y,\Theta\right]\right)\right] \\ &= \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\mathsf{Pr}\left[\mathbf{v}|Y,\phi\right]\right) - \log\left(\mathsf{Pr}\left[\mathbf{v}|\mathbf{T},Y,\Theta\right]\right)\right] \\ &\quad + \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\mathsf{Pr}\left[\mathbf{T}|Y,\mathbf{v},\phi\right]\right)\right] \\ &= -\mathcal{D}_{\mathrm{KL}}\left(\mathsf{Pr}\left[\mathbf{V}|\mathbf{T},Y,\Theta\right]||\mathsf{Pr}\left[\mathbf{V}|Y,\phi\right]\right) + \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\log\left(\mathsf{Pr}\left[\mathbf{T}|Y,\mathbf{v},\phi\right]\right)\right].\end{aligned}$$

As mentioned in Sec.3.2, the prior distribution $\mathsf{Pr}\left[\mathbf{V}|Y,\phi\right]$ can be reduced to $\mathsf{Pr}\left[\mathbf{V}\right]$ because $\mathbf{V}$ is independent from the label Y and ϕ. □

The equality between Eq.5 and Eq.6 holds if and only if the encoder $F_\Theta^{(enc)}$, which approximates the parameters $\boldsymbol{\mu}_{\mathbf{V},\Theta}$ and $\Sigma_{\mathbf{V},\Theta}$ that are needed to compute $\mathsf{Pr}[\mathbf{V}|\mathbf{T},Y,\Theta]$, is able to perfectly predict $\mathsf{Pr}[\mathbf{V}|\mathbf{T},Y,\phi]$. In such configuration, the latent space exactly captures the random part induced in a trace $\mathbf{T}$. Based on Eq.6, we define the empirical risk that we minimize to train the cVAE-SA.

Definition 1 (Empirical risk combined with Evidence Lower BOund (ELBO) Loss)**.** [KW14] Given a latent space $\mathcal{V}$, a set of N_p labeled traces $\mathcal{I}_p = \{(\mathbf{t}_0,y_0),\ldots,(\mathbf{t}_{N_p-1},y_{N_p-1})\}$, we define the empirical risk optimizing $F_{\Theta,\phi}$, that approximates the generative distribution $\mathsf{Pr}[\mathbf{T}|Y]$, as follows:

$$\begin{aligned}\hat{\mathcal{R}}\left(\mathcal{L}_{ELBO},F_{\Theta,\phi}\right) = \frac{1}{N_p}\sum_{i=0}^{N_p-1} & \underbrace{\mathcal{D}_{\mathrm{KL}}\left(\mathsf{Pr}\left[\mathbf{V}|\mathbf{t}_i,y_i,\Theta\right]||\mathsf{Pr}\left[\mathbf{V}\right]\right)}_{\text{KL-Divergence Loss}} \\ & \underbrace{-\,\mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\log\left(\mathsf{Pr}[\mathbf{t}_i|y_i,\mathbf{v},\phi]\right)}_{\text{Reconstruction Loss}},\end{aligned}$$

such that $(\mathsf{Pr}[\mathbf{V}|\mathbf{t}_i,y_i,\Theta])_{0\leq i<N_p}$ is computed from $\boldsymbol{\mu}_{\mathbf{V},\Theta}$ and $\Sigma_{\mathbf{V},\Theta}$ provided by the encoder $(F_\Theta^{(enc)}(\mathbf{t}_i))_{0\leq i<N_p}$ and $(\mathsf{Pr}[\mathbf{t}_i|y_i,\mathbf{v},\phi])_{0\leq i<N_p}$ is obtained from $F_\phi^{(dec)}(\mathbf{v})$.

Sampling $\mathbf{v}$ from the learned posterior $\mathsf{Pr}[\mathbf{V}|\mathbf{T},Y,\Theta]$ knowing the trace $\mathbf{T}$, the related label Y and the multivariate Gaussian distribution $\mathcal{N}_D(\boldsymbol{\mu}_{\mathbf{V},\Theta},\Sigma_{\mathbf{V},\Theta})$, can be seen as encoding $\mathbf{T}$ into $\mathbf{v}$, while $F_\phi^{(dec)}$ seeks to reconstruct $\mathbf{T}$ from $\mathbf{v}$. Classically used for training a variational autoencoder [KW14, KWKT15, GDG+15, SSB17], the loss function defined in Def.1 can be decomposed into two terms: the *reconstruction* and the *KL-divergence* terms. From a general perspective, to minimize the reconstruction loss, the embedding means $\boldsymbol{\mu}_{\mathbf{V},\Theta}$, for various Y, are pushed far away from each other and embedding standard deviations $\Sigma_{\mathbf{V},\Theta}$ are pulled toward zero. On the other hand, to get smaller $\mathcal{D}_{\mathrm{KL}}(\mathsf{Pr}[\mathbf{V}|\mathbf{T},Y,\Theta]||\mathsf{Pr}[\mathbf{V}])$, the embedding means are pulled toward zero and the embedding standard deviations are increased. While the KL-divergence term is opposed to the reconstruction loss, it can be seen as a regularization term. Indeed, putting a lot of information about $\mathbf{T}$ in $\mathbf{V}$ makes reconstruction trivial, but the penalization induced by the regularization term is

non-negligible. Therefore, the regularization term acts as an information bottleneck, so a balance between both terms must be found in order to only keep the informative and generic features. If necessary, the KL-divergence loss can be monitored by a hyperparameter β. In the state-of-the-art, these models are called β-*Variational AutoEncoders* [HMP+17]. However, as this paper bridges the stochastic attacks with the cVAE model, the impact of the β-parameter on the resulted learning algorithm is considered as out of the scope of this paper.

Remark 1. The readers might notice that the minimization optimization process is conducted on the empirical risk combined with the ELBO loss. Therefore, the reconstruction and the KL-divergence losses are simultaneously computed to train the encoder and the decoder of the cVAE-SA.

Reconstruction loss. This term, denoted by $\mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}} \log(\mathsf{Pr}[\mathbf{T}|Y,\mathbf{v},\phi])$, is linked with the decoder $F_\phi^{(dec)}$ introduced in Sec.3.2. It defines the probability of constructing $\mathbf{T} \in \mathbb{R}^D$ given the label $Y \in \mathbb{F}_2^n$ and a sample $\mathbf{v} \in \mathbb{R}^D$ of the latent space $\mathcal{V}$. Hence, the reconstruction loss tends to maximize the log likelihood in order to construct traces that are correlated with the true unknown leakage model ψ and the noise $\mathbf{Z}$ related to $\mathbf{T}$. Thus, it encourages the decoder to learn how a trace can be reconstructed from a given noise representation defined by a latent variable $\mathbf{V} \sim \mathcal{N}_D(\boldsymbol{\mu}_{\mathbf{V},\Theta}, \Sigma_{\mathbf{V},\Theta})$ (see Eq.4). The reconstruction loss optimizes the parameters ϕ to retrieve the correct coefficients associated with each vector of the monomial basis $(Y^u)_{u\in\mathbb{F}_2^n}$.Typically, if we only consider the case where no interaction between the time samples of $\mathbf{T}$ occurs, then, the covariance matrix $\Sigma_{\mathbf{V},\Theta}$ can be simplified to a diagonal matrix such that its vector representation can be described as $\sigma^2_{\mathbf{V},\Theta} = [\Sigma_{\mathbf{V},\Theta}[0,0], \Sigma_{\mathbf{V},\Theta}[1,1], \ldots, \Sigma_{\mathbf{V},\Theta}[D,D]]$. In such configuration, we do not expect to capture the time samples' interaction related to the constructed trace $\tilde{\mathbf{T}} \in \mathbb{R}^D$. Thus, the reconstruction loss can be computed as follows:

$$\mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}} - \log\left(\mathsf{Pr}\left[\mathbf{T}|Y,\mathbf{v},\phi\right]\right) = \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}}\left[\sum_{i=0}^{D-1} \frac{1}{2}\log\left(2\pi\sigma^2_{\tilde{\mathbf{T}}}[i]\right) + \frac{(\mathbf{T}[i]-\boldsymbol{\mu}_{\tilde{\mathbf{T}}}[i])^2}{2\sigma^2_{\tilde{\mathbf{T}}}[i]}\right], \quad (7)$$

where $\mu_{\tilde{\mathbf{T}}}[i]$ (resp. $\sigma^2_{\tilde{\mathbf{T}}}[i]$) indicates the i^{th} element of the mean (resp. variance) vector of generated traces $\tilde{\mathbf{T}}$ given a set of latent representations and a deterministic part $\hat{\psi}_\phi$ which depends on $Y = f(X, k^*)$ (see Eq.4). However, assuming that $\Sigma_{\mathbf{V},\Theta}$ can be simplified to a diagonal matrix affects the ability of the generated trace $\tilde{\mathbf{T}}$ to capture the interaction between the time samples of $\mathbf{T}$. While this choice can be problematic from a performance perspective[g], the computation gain is non-negligible as the matrix inversion does not have to be computed in order to process the reconstruction loss.

Then, we assume that the output distribution of the conditional variational autoencoder is an isotropic Gaussian[h] (*i.e.* for all $\mathbf{v} \sim \mathcal{N}_D(\boldsymbol{\mu}_{\mathbf{V},\Theta}, \mathrm{diag}(\Sigma_{\mathbf{V},\Theta}))$, we can define $\Sigma_{\tilde{\mathbf{T}}} = \sigma^2 \cdot \mathbf{I}_D$ where σ^2 is a scalar). While the *Mean Squared Error* (MSE) loss function can be written as $\mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}} ||\mathbf{T} - \boldsymbol{\mu}_{\tilde{\mathbf{T}}}||_2$, Eq.7 can be simplified as follows:

$$\mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}} - \log\left(\mathsf{Pr}\left[\mathbf{T}|Y,\mathbf{v},\phi\right]\right) = \frac{1}{2}\log\left(2\pi\sigma^2\right) + \frac{\mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}} ||\mathbf{T} - \boldsymbol{\mu}_{\tilde{\mathbf{T}}}||_2}{2\sigma^2}. \quad (8)$$

Note that this solution is minimized if the scalar $\sigma^2 = \mathbb{E}_{\mathbf{v}\sim F_\Theta^{(enc)}} ||\mathbf{T} - \boldsymbol{\mu}_{\tilde{\mathbf{T}}}||_2 = \mathrm{MSE}(\mathbf{T}, \boldsymbol{\mu}_{\tilde{\mathbf{T}}})$ [Yu20].

[g] To nuance this issue, Bruneau *et al.* [BGH+15, Fig.3] illustrate that the information induced by the covariance matrix is mainly brought by its diagonal.

[h] This choice can be justified in this paper as all the manipulated traces are standardized (*i.e.* zero mean, unit variance).

This loss is approximated via *Monte-Carlo* sampling, however, due to computation constraints, we consider only one sample $\mathbf{v}$ for computing Eq.8 during the training process. Consequently, for an estimated trace $\tilde{\mathbf{T}}$, we minimize its L^2-norm from the related true trace $\mathbf{T}$ in order to find the best parameters ϕ. In other words, through this solution, we attempt to find an estimated trace $\tilde{\mathbf{T}}$ as similar as the real one $\mathbf{T}$. Thus, the decoder $F_{\phi}^{(dec)}$ is only affected by the reconstruction loss and seeks to suitably reconstruct $\tilde{\mathbf{T}}$ based on a latent representation $\mathbf{V}$ and a deterministic part $\hat{\psi}_{\phi}$.

KL-divergence loss. However, to reduce the overfitting issue, a *regularization* term is added. In addition to the optimization of ϕ, the cVAE concurrently optimizes Θ to minimize the KL-divergence of the approximation $\mathsf{Pr}[\mathbf{V}|\mathbf{T}, Y, \Theta]$ from $\mathsf{Pr}[\mathbf{V}]$. As a remainder, the better $\mathsf{Pr}[\mathbf{V}|\mathbf{T}, Y, \Theta]$ approximates the true posterior distribution $\mathsf{Pr}[\mathbf{V}|\mathbf{T}, Y, \phi]$, in terms of the KL divergence, the smaller the gap between $\hat{\mathcal{R}}(\mathcal{L}_{ELBO}, F_{\Theta,\phi})$ and the marginal log-likelihood $\log(\mathsf{Pr}[\mathbf{T}|Y, \phi])$. Both $\mathsf{Pr}[\mathbf{V}|\mathbf{T}, Y, \Theta]$ and $\mathsf{Pr}[\mathbf{V}]$ are assumed to be Gaussian, specifically, $\mathsf{Pr}[\mathbf{V}|\mathbf{T}, Y, \Theta]$ follows $\mathcal{N}_D(\boldsymbol{\mu}_{\mathbf{V},\Theta}, \Sigma_{\mathbf{V},\Theta})$ and $\mathsf{Pr}[\mathbf{V}]$ follows $\mathcal{N}_D(0, \mathbf{I}_D)$. The latter distribution is assumed as the traces are standardized, *i.e.* zero mean and unit variance, and such that no interactions are captured between the time samples. As $\mathsf{Pr}[\mathbf{V}]$ characterizes the random part of $\tilde{\mathbf{T}}$ (see Eq.4), it has to follow the same distribution as the random part of the real trace $\mathbf{T}$ which is $\mathcal{N}(0, 1)$ for each non-informative time sample. Through this configuration, the KL-divergence can be computed as follows:

$$
\begin{aligned}
D_{\mathrm{KL}}(\mathsf{Pr}\left[\mathbf{V}|\mathbf{T}, Y, \Theta\right]||\mathsf{Pr}\left[\mathbf{V}\right]) &= \mathbb{E}_{\mathbf{v}\sim F_{\Theta}^{(enc)}}\left[\log(\mathsf{Pr}\left[\mathbf{v}|\mathbf{T}, Y, \Theta\right]) - \log(\mathsf{Pr}\left[\mathbf{v}\right])\right] \\
&= \mathbb{E}_{\mathbf{v}\sim F_{\Theta}^{(enc)}}\left[\frac{1}{2}\log\left(\frac{|\mathbf{I}_D|}{|\Sigma_{\mathbf{V},\Theta}|}\right) - \frac{1}{2}(\mathbf{v}-\boldsymbol{\mu}_{\mathbf{V},\Theta})^T\Sigma_{\mathbf{V},\Theta}^{-1}(\mathbf{v}-\boldsymbol{\mu}_{\mathbf{V},\Theta})\right. \\
&\quad \left. + \frac{1}{2}(\mathbf{v}-0)^T\mathbf{I}_D^{-1}(\mathbf{v}-0)\right] \\
&= -\frac{1}{2}\log\left(|\Sigma_{\mathbf{V},\Theta}|\right) - \frac{1}{2}\mathbb{E}_{\mathbf{v}\sim F_{\Theta}^{(enc)}}\left[(\mathbf{v}-\boldsymbol{\mu}_{\mathbf{V},\Theta})^T\Sigma_{\mathbf{V},\Theta}^{-1}(\mathbf{v}-\boldsymbol{\mu}_{\mathbf{V},\Theta})\right] \\
&\quad + \frac{1}{2}\mathbb{E}_{\mathbf{v}\sim F_{\Theta}^{(enc)}}\left[\mathbf{v}^T\mathbf{I}_D^{-1}\mathbf{v}\right] \\
&= -\frac{1}{2}\log\left(|\Sigma_{\mathbf{V},\Theta}|\right) - \frac{1}{2}\mathrm{tr}(\Sigma_{\mathbf{V},\Theta}^{-1}\Sigma_{\mathbf{V},\Theta}) + \frac{1}{2}\left(\boldsymbol{\mu}_{\mathbf{V},\Theta}^T\mathbf{I}_D^{-1}\boldsymbol{\mu}_{\mathbf{V},\Theta} + \mathrm{tr}(\mathbf{I}_D^{-1}\Sigma_{\mathbf{V},\Theta})\right) \\
&= \frac{1}{2}\left(-\log\left(|\Sigma_{\mathbf{V},\Theta}|\right) - tr(\mathbf{I}_D) + \boldsymbol{\mu}_{\mathbf{V},\Theta}^T\cdot\boldsymbol{\mu}_{\mathbf{V},\Theta} + \mathrm{tr}(\Sigma_{\mathbf{V},\Theta})\right). \qquad (9)
\end{aligned}
$$

As $\Sigma_{\mathbf{V},\Theta}$ can be rewritten as a vector $\sigma^2_{\mathbf{V},\Theta}$ such that each element of $(\sigma^2_{\mathbf{V},\Theta}[i])_{0\leq i<D}$ defines the i^{th} diagonal of $\Sigma_{\mathbf{V},\Theta}$, then, Eq.9 can be expressed as follows:

$$
\begin{aligned}
\mathcal{D}_{\mathrm{KL}}\left(\mathsf{Pr}\left[\mathbf{V}|\mathbf{T}, Y, \Theta\right]||\mathsf{Pr}\left[\mathbf{V}\right]\right) &= -\frac{1}{2}\left(\log\prod_{i=0}^{D-1}\sigma^2_{\mathbf{V},\Theta}[i] + \sum_{i=0}^{D-1}1 - \sum_{i=0}^{D-1}\boldsymbol{\mu}^2_{\mathbf{V},\Theta}[i] - \sum_{i=0}^{D-1}\sigma^2_{\mathbf{V},\Theta}[i]\right) \\
&= -\frac{1}{2}\sum_{i=0}^{D-1}\left(1 + \log\left(\sigma^2_{\mathbf{V},\Theta}[i]\right) - \boldsymbol{\mu}^2_{\mathbf{V},\Theta}[i] - \sigma^2_{\mathbf{V},\Theta}[i]\right).
\end{aligned}
$$

As a remainder, for correctly dealing with the stochastic attack scenario, the deterministic part (*i.e.* $\psi(f(X, k^*))$ as well as the random part (*i.e.* $\mathbf{Z}$) should be correctly characterized by the cVAE-SA model. While the deterministic part is approximated by the $\hat{\psi}$ layer, the random part is modeled by the latent space $\mathcal{V}$ (see Sec.3.2). Therefore, a well-trained cVAE-SA should provide a latent space that is representative of the random part $\mathbf{Z}$. Through the use of the KL-divergence loss, we force a latent variable $\mathbf{V}$ to follow

$\mathcal{N}_D(0, \mathbf{I}_D)$. To clearly explain the impact of the KL-divergence loss on the trainable parameters Θ, let us denote $\mathbf{T}$ a D-dimensional trace that has been standardized at each sample. Let $\{l_0, \ldots, l_{s-1}\}$ define a set of indices where the sensitive information leaks (*i.e.* PoIs) such that,

$$\mathbf{T}[i] = \begin{cases} \psi(Y)[i] + \mathbf{Z}[i] \sim \mathcal{N}(0,1) & \text{if} \quad i \in \{l_0, \ldots, l_{s-1}\}, \\ \mathbf{Z}[i] \sim \mathcal{N}(0,1) & \text{otherwise.} \end{cases}$$

In this setting, we assume that the interactions between trace samples are negligible. If the i^{th} time sample of $\mathbf{T}$ has no deterministic part (*i.e.* $i \notin \{l_0, \ldots, l_{s-1}\}$), the related element of the latent variable $\mathbf{V}[i] = \mathbf{Z}[i] - \hat{\psi}_\Theta(Y)[i]$, which follows $\mathcal{N}(0,1)$, induces that $\hat{\psi}_\Theta[i]$, and thus the trainable parameters Θ, are negligible. Consequently, if $\psi(Y) = 0$, the related sample of the latent variable follows the same distribution as $\mathbf{Z}[i]$. In such scenario, the KL-divergence loss is negligible as $\mathsf{Pr}[\mathbf{V}|\mathbf{T}, Y, \Theta] = \mathsf{Pr}[\mathbf{V}]$. Thus, considering these non-informative time samples do not affect the regularization term and are unsuitable for the decision process. This result encourages the evaluator to only consider the time samples with a non-negligible deterministic part (*i.e.* the points of interest).

On the other hand, if $i \in \{l_0, \ldots, l_{s-1}\}$, then $\mathbf{V}[i] = \psi(Y)[i] + \mathbf{Z}[i] - \hat{\psi}_\Theta(Y)[i]$ follows $\mathcal{N}(0,1)$. Thus, it suggests that $\mathbf{Z}[i] \sim \mathcal{N}(\mathbb{E}[\hat{\psi}_\Theta(Y)[i] - \psi(Y)[i]], \mathbb{V}[\psi(Y)[i]] + \mathbb{V}[\hat{\psi}_\Theta(Y)[i]] + \mathbb{V}[\mathbf{V}[i]] - 2 \cdot Cov[\psi[i], \hat{\psi}_\Theta[i]] - 2 \cdot Cov[\psi[i], \mathbf{V}[i]] + 2 \cdot Cov[\hat{\psi}_\Theta[i], \mathbf{V}[i]])$. However, due to the KL-divergence loss function involved during the training process, we force the latent variable $\mathbf{V}$ to follow $\mathcal{N}_D(0, \mathbf{I}_D)$. As defined in Sec.3.2, this latent variable characterizes an estimation of the noise $\mathbf{Z}$ induced in the trace $\mathbf{T}$. Thus, during the training process of the cVAE-SA, we penalize the model to tend $\mathbb{E}[\hat{\psi}_\Theta(Y)[i] - \psi(Y)[i]]$ towards 0 and $\mathbb{V}[\psi(Y)[i]] + \mathbb{V}[\hat{\psi}_\Theta(Y)[i]] + \mathbb{V}[\mathbf{V}[i]] - 2 \cdot Cov[\psi[i], \hat{\psi}_\Theta[i]] - 2 \cdot Cov[\psi[i], \mathbf{V}[i]] + 2 \cdot Cov[\hat{\psi}_\Theta[i], \mathbf{V}[i]]$ towards 1 such that this solution is reached if and only if $\hat{\psi}_\Theta = \psi$. Consequently, when the KL-divergence loss is computed, the cVAE-SA optimizes the trainable parameters Θ of the encoder $F_\Theta^{(enc)}$ such that the regularization term equals 0 if and only if Θ is optimal. In addition, to fully assess the suitability of the training process, the evaluator can visualize the trainable parameters Θ such that, if the correct leakage model appears, therefore, the cVAE-SA model is well trained. A discussion related to this visualization technique is provided in Sec.4.2.

This justification suggests that the latent space should be only composed by PoIs. When the input traces are standardized (*i.e.* zero mean, unit variance), considering the KL-divergence loss is helpful to reduce the impact of irrelevant time samples. However, when the Gaussian noise increases, the dependence between $\mathbf{T}[i]$ and $\psi[i]$ decreases. In this configuration, differentiating the sensitive information from the noise can be difficult as $\mathbf{Z}[i]$ approximately follows $\mathcal{N}(0,1)$ regardless of the information included in the time sample i. This observation confirms the benefits of the noise to reduce the efficiency of DLSCA approach. This observation will be confirmed in Sec.4 and in App.B.

From a practical perspective, even if the ELBO loss function is composed of two sub-losses, namely reconstruction and KL-divergence losses, a single optimization process is performed in order to minimize the ELBO loss. Once the generative model $F_{\Theta,\phi}$ is trained, the evaluator has to make a decision following the approximation of $\mathsf{Pr}[\mathbf{T}|Y]$ in order to fit with the stochastic attack approach. The following section describes this strategy.

3.4 Decision rule & network complexity

Typically, in the Machine Learning community, the inference phase of cVAE consists in generating a new set of data based on an input and a conditional known label. In SCA context, our goal is different and tends to find the conditional unknown label Y that fits

best for a given trace $\mathbf{T}$. The following part describes a new solution to retrieve the secret key k^* from the model previously defined.

Key recovery phase. During the training phase, we defined a function $F_{\Theta,\phi}$ that approximates $\log(\mathsf{Pr}[\mathbf{T}|Y,\phi])$ through an optimization algorithm (*i.e.* gradient descent-based algorithms) such that the generated trace $\tilde{\mathbf{T}}$, defined by the output of the decoder $F_{\phi}^{(dec)}$, is close to the real one $\mathbf{T}$ captured for a given label. Once the encoder and the decoder are successfully trained simultaneously to optimize $F_{\Theta,\phi}$, the evaluator can dissociate them in order to extract the unknown secret key from a targeted device. As mentioned in Sec.3.1, the encoder is defined by $F_{\Theta}^{(enc)} : \mathbb{R}^D \times \mathbb{F}_2^n \to \mathbb{R}^D \times \mathcal{M}_{D,D}(\mathbb{R})$, while the decoder is denoted by $F_{\Theta}^{(dec)} : \mathbb{R}^D \times \mathbb{F}_2^n \to \mathbb{R}^D$. The key recovery phase will use these functions independently in order to retrieve the targeted secret key. To fully understand this strategy, a *modus operandi* is suggested for a given key hypothesis $k \in \mathcal{K}$:

1. First, the evaluator generates a new set of traces from the targeted device[i] with a fixed unknown secret key k^*. Let $\mathcal{I}_a$ be the set of N_a attack traces such that $\mathcal{I}_a = \{\mathbf{t}_0, \ldots, \mathbf{t}_{N_a-1}\}$.
2. For each trace $\mathbf{t}$ in $\mathcal{I}_a$,
   (a) The evaluator computes the label $Y = f(X,k)$ related to $\mathbf{t}$ by mixing the known plaintexts $X \in \mathcal{X}$ and the key hypothesis k.
   (b) Then, he estimates the parameters $\boldsymbol{\mu}_{\mathbf{V},\Theta} \in \mathbb{R}^D$ and $\Sigma_{\mathbf{V},\Theta} \in \mathcal{M}_{D,D}(\mathbb{R})$ of the multivariate Gaussian distribution $\mathcal{N}_D(\boldsymbol{\mu}_{\mathbf{V},\Theta}, \Sigma_{\mathbf{V},\Theta})$ through the computation of $F_{\Theta}^{(enc)}(\mathbf{t}, Y)$.
   (c) Given $\boldsymbol{\mu}_{\mathbf{V},\Theta}$ and $\Sigma_{\mathbf{V},\Theta}$, the evaluator generates a set of N_v latent samples $\{\mathbf{v}_0, \ldots, \mathbf{v}_{N_v-1}\}$ such that $(\mathbf{v}_i \sim \mathcal{N}_D(\boldsymbol{\mu}_{\mathbf{V},\Theta}, \Sigma_{\mathbf{V},\Theta}))_{i\in\{0,\ldots,N_v-1\}}$.
   (d) Thanks to the decoder $F_{\Theta}^{(dec)} : \mathbb{R}^D \times \mathbb{F}_2^n \to \mathbb{R}^D$, the evaluator constructs a set of N_v synthetic traces $\tilde{\mathbf{t}} \in \mathbb{R}^D$ such that $\left(\tilde{\mathbf{t}}_i = F_{\Theta}^{(dec)}(\mathbf{v}_i, Y)\right)_{i\in\{0,\ldots,N_v-1\}}$.
   (e) Finally, based on the N_v synthetic traces, he can estimate $\mathsf{Pr}[\mathbf{T}|Y,\phi]$ through the computation of an approximation of the marginal log-likelihood:

$$\log\left(\mathsf{Pr}\left[\mathbf{t}_i|y_i,\phi\right]\right) \approx -\mathcal{D}_{\mathrm{KL}}\left(\mathsf{Pr}\left[\mathbf{V}|\mathbf{t}_i,y_i,\Theta\right]||\mathsf{Pr}\left[\mathbf{V}\right]\right) - \frac{1}{2}\log\left(2\pi\cdot\sum_{j=0}^{D-1}\left(\mathbf{t}_i[j] - \frac{1}{N_v}\sum_{h=0}^{N_v-1}\tilde{\mathbf{t}}_h[j]\right)^2\right) - \frac{1}{2}, \tag{10}$$

   where $\tilde{\mathbf{t}}_h = \hat{\psi}_\phi(y_i) + \mathbf{v}_h$ is the h^{th} generated trace constructed from $\mathsf{Pr}[\mathbf{t}_i|y_i,\mathbf{v}_h,\phi]$ (see Eq.8).

When the inferred posterior $\mathsf{Pr}[\mathbf{V}|\mathbf{T},Y,\Theta]$ deviates from the true unknown posterior $\mathsf{Pr}[\mathbf{V}|\mathbf{T},Y,\phi]$, the number of samples N_v increases in order to obtain an accurate approximation of $\mathsf{Pr}[\mathbf{T}|Y,\phi]$. If the profiling phase has been performed successfully, then $(\mathbf{t}_i - \frac{1}{N_v}\sum_{j=0}^{N_v-1}\tilde{\mathbf{t}}_j)^2$ should be minimized when $k = k^*$. Hence, the most likely candidate is defined through the maximum likelihood rule:

$$\hat{k} = \underset{k\in\mathcal{K}}{\arg\max}\left(\sum_{i=0}^{N_a-1}\log\left(\mathsf{Pr}\left[\mathbf{t}_i|f(x_i,k),\phi\right]\right)\right).$$

[i] As this paper is dedicated to the profiled attack scenario, the readers must consider the targeted device as identical to the open device used during the training process.

Following Eq.4, $\tilde{\mathbf{T}} \sim \mathcal{N}_D(\boldsymbol{\mu}_{\tilde{\mathbf{T}}}, \Sigma_{\tilde{\mathbf{T}}})$ such that, $\boldsymbol{\mu}_{\tilde{\mathbf{T}}} = \hat{\psi}_\phi$ and $\Sigma_{\tilde{\mathbf{T}}} = \Sigma_{\mathbf{V},\Theta}$. As a consequence, through this key recovery phase, the evaluator aims at identifying the hypothetical leakage model $\hat{\psi}_\phi(f(X,k))$ which fits the most with T. Consequently, this process exploits the first order moment to recover information about the secret key. This observation is confirmed in App.A.

To enhance the key extraction phase, the evaluator can precisely define the PoIs' indexes *via* a leakage assessment once the profiling phase is performed. Indeed, if Θ and ϕ are correctly learned, the evaluator can visualize them in order to properly select the PoIs (see Sec.4.2). Thus, during the attack phase, instead of parsing all the samples j, the evaluator can only compute Eq.10 on the samples that are considered relevant for the cVAE-SA.

Theoretical network complexity bounds. Based on the previous sections, we can efficiently find an architecture for a given implementation. Consequently, some theoretical network complexity bounds can be expressed following the evaluator's knowledge. Indeed, our generative neural network (*i.e.* cVAE-SA) can be easily built for a given $Y \in \mathbb{F}_2^n$, a degree d of bits' interaction and a D-dimensional trace $\mathbf{T} \in \mathbb{R}^D$.

First, for estimating $(\hat{\psi}_\Theta[i])_{0\leq i<D}$ (resp. $(\hat{\psi}_\phi[i])_{0\leq i<D}$), the encoder (resp. decoder) needs to optimize Θ (resp. ϕ) in order to retrieve the correct leakage model. Hence, for a given $Y \in \mathbb{F}_2^n$, the number of weights that have to be optimized are $((1+\sum_{i=0}^{d}\binom{n}{i})\cdot D)$ in both cases (s.t. $d \leq n$). Then, for estimating $(\mathbf{V}[i])_{0\leq i<D}$ (resp. $(\tilde{\mathbf{T}}[i])_{0\leq i<D}$), we have to link the i^{th} sample of the trace $\mathbf{T}$ (resp. the latent variable $\mathbf{V}$) with the related $\hat{\psi}_\Theta[i]$ (resp. $\hat{\psi}_\phi[i]$). Here, we decide to follow the classical stochastic attacks in order to easily extract the related noise. Hence, no weights are needed for this operation. Finally, to approximate $\boldsymbol{\mu}_{\mathbf{V},\Theta}$ (resp. $\Sigma_{\mathbf{V},\Theta}$), we need $(D\cdot(D+1))$ (resp. $D^2\cdot(D+1)$) neurons. For the simplified diagonal case, $\Sigma_{\mathbf{V},\Theta}$ can be reduced to $\sigma^2_{\mathbf{V},\Theta}$, thus, only $D\cdot(D+1)$ neurons are needed in this configuration. To sum up the complexity metrics, the evaluator needs to construct a generative model with $(D\cdot((D+1)^2+2\cdot(1+\sum_{i=0}^{d}\binom{n}{i})))$ weights (resp. $(2D\cdot((D+1)+1+\sum_{i=0}^{d}\binom{n}{i}))$ weights if $\Sigma_{\mathbf{V},\Theta}$ is reduced to $\sigma^2_{\mathbf{V},\Theta}$). Following those metrics, it can be noticed that the trace dimension D influences the most of the network complexity.

However, a solution can be considered to improve the network complexity without altering the cVAE-SA performance. Indeed, following Sec.3.3, if the evaluator detects s PoIs, he can construct a vector $\{l_0,\ldots,l_{s-1}\}$ of s indices such that l_i denotes the index related to the i^{th} point of interest. Based on this knowledge, he can build a cVAE-SA with lower complexity such that most of the relevant information, dedicated to the s PoIs, can be extracted from a trace. Instead of considering all the samples of the D-dimensional trace (s.t. $D \gg s$), he can construct a neural network with $(s\cdot((s+1)^2+2\cdot(1+\sum_{i=0}^{d}\binom{n}{i})))$ weights (resp. $(2s\cdot((s+1)+1+\sum_{i=0}^{d}\binom{n}{i}))$ weights if $\Sigma_{\mathbf{V},\Theta}$ is reduced to $\sigma^2_{\mathbf{V},\Theta}$). As a consequence, we drastically reduce the network complexity without altering the ability of the generative model to retrieve the secret key as suggested in Sec.3.3. For example, the network complexity of Fig.2b is about $1,040$ weights if all bits' interactions are considered (*i.e.* $d=8$, $s=2$ and $n=8$). When black-box models (e.g. discriminative models) are considered, finding such complexity bounds is known as an arduous task as no correlations are provided with classical profiled SCA.

One of the main benefits of the proposed variational autoencoder is its explainability and its interpretability regarding the side-channel context. In addition, our theoretical results suggest that its width does not have to be large no matter the dimension of the traces. This result is faithful with the Universal Approximation Theorem [Pin99]. Through the following section, we validate these properties and broaden the attacks' spectrum on protected implementations considering the boolean making scheme.

4 Empirical investigations on cVAE-SA

4.1 Settings

Hyperparameter selection. While classical DLSCA models need to tune a lot of hyperparameters (e.g. type of neural network, number of layers, number of nodes per layer, activation function, optimizer algorithms, learning rate, number of epochs, batch size), the configuration of the proposed cVAE-SA only deals with the optimizer algorithm, the batch size, the learning rate and the number of epochs. In this section, optimization is done using the *Adam* optimizer on batch size $\{8, 16, 32, 64, 128\}$ and the learning rate is set to $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. We construct each model with a maximum number of epochs of 40 and select the hyperparameters that provide the best ranking value. Finally, in this section, Nt_{rank} denotes the number of attack traces that are needed to reach a constant rank of 1. These traces are randomly shuffled and picked up from a set of attack traces $\mathcal{I}_a$ which is characterized by simulations that are described in the following. For a good estimation of Nt_{rank}, an average over 10 simulations, denoted $\bar{Nt}_{\text{rank}}$, is computed.

Simulations. To verify the benefits of the cVAE-SA, we simulate D-dimensional traces from a 8-bit sensitive variable Y. In this section, the simulated traces are built following two scenarios:

- **Scenario 1** – We assume the leakage model induces the maximum amount of interactions between bits (*i.e.* $\mathcal{F}_9$), such that all bits influencing the leakage model have the same weights. Hence, the i^{th} time sample of the simulated trace $\mathbf{T}$ is defined as follows:

$$\mathbf{T}[i] = \begin{cases} 1 \cdot Y[1] + 1 \cdot Y[3] + 1 \cdot Y[6] \\ + 1 \cdot \oplus_{b=0}^{1} Y[b] + 1 \cdot \oplus_{b=0}^{2} Y[b] + 1 \cdot \oplus_{b=0}^{3} Y[b] \\ + 1 \cdot \oplus_{b=0}^{4} Y[b] + 1 \cdot \oplus_{b=0}^{5} Y[b] + 1 \cdot \oplus_{b=0}^{6} Y[b] \\ + 1 \cdot \oplus_{b=0}^{7} Y[b] + \mathbf{Z}[i] & \text{if } i \in \{l_0, \ldots, l_{s-1}\}, \\ \mathbf{Z}[i] & \text{otherwise,} \end{cases} \tag{11}$$

  where $\{l_0, \ldots, l_{s-1}\}$ defines a set of indices related to each PoI, $\oplus_{b=0}^{n} Y[b] = Y[0] \oplus \ldots \oplus Y[n]$, $Y[b] = Sbox[X \oplus k^*][b]$ denotes the b^{th} bit of the output of the Sbox, and $\mathbf{Z}[i]$ is a Gaussian noise following $\mathcal{N}(0, \sigma^2)$ such that $\sigma^2 = 1$. The SNR result is provided in App.B.

- **Scenario 2** – We assume that the leakage model induces interactions of degree 2 between bits (*i.e.* $\mathcal{F}_3$) but differs by the location of the PoIs. The i^{th} time sample of the trace $\mathbf{T}$ is defined as follows:

$$\mathbf{T}[i] = \begin{cases} 1 \cdot (X \oplus k^*)[5] + 1 \cdot (X \oplus k^*)[3] & \\ \oplus\, (X \oplus k^*)[7] + \mathbf{Z}[i] & \text{if } i \in \{l_0', \ldots, l_{s'-1}'\}, \\ 1 \cdot Sbox\left[X \oplus k^*\right][3] + 1 \cdot Sbox\left[X \oplus k^*\right][6] + \mathbf{Z}[i] & \text{if } i \in \{l_0, \ldots, l_{s-1}\}, \\ \mathbf{Z}[i] & \text{otherwise,} \end{cases} \tag{12}$$

  where $Sbox\left[X \oplus k^*\right][b]$ denotes the b^{th} bit of the output of the Sbox considering a plaintext X and the secret key k^*, $\mathbf{Z}[i]$ is a Gaussian noise following $\mathcal{N}(0, \sigma^2)$ such that $\sigma^2 = 1$. The SNR result is provided in App.B.

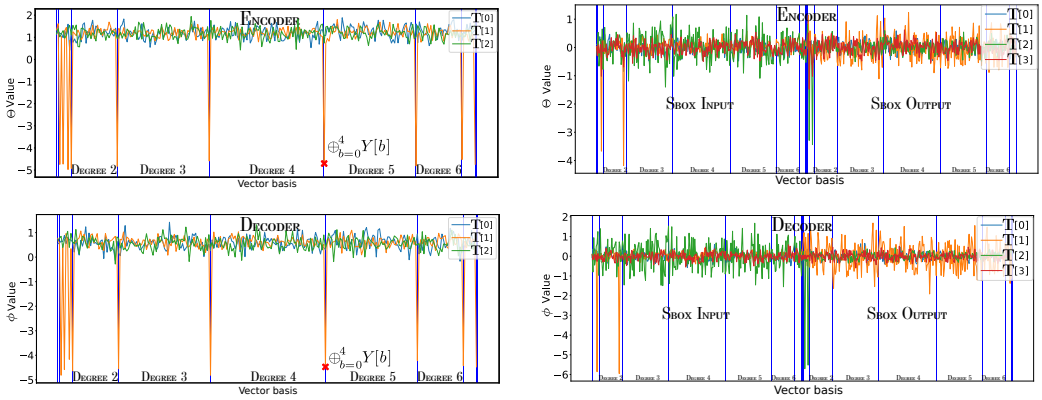
A set of $10,000$ traces ($9,000$ for the profiling phase and $1,000$ for the validation phase) is simulated for each scenario. The choice of these scenarios have been motivated to assess

the ability of the cVAE-SA to capture the interactions between bits as well as simultanously targeting multiple sensitive variables. Further experiments with other scenarios has been investigated in App.A and App.B considering additional Gaussian noise parameters.

Limitations of the monomial basis. In [SLP05], Schindler *et al.* suggest a non-orthonormal monomial basis to approximate the leakage model ψ. This proposition limits the evaluator's interpretation when combination of bits are induced in the leakage model. To ease its interpretation, Guilley *et al.* propose a decomposition of the monomial basis to isolate the leakage from the combination of bits [GHMR17]. Through the application of the well-known *Gram-Schmidt orthonormalization* on the monomial basis, Guilley *et al.* introduce a new orthonormal monomial basis that uncorrelates each basis vector and preserve the degree of bits' interaction. Hence, constructing the cVAE-SA on this orthonormal monomial basis is beneficial to evaluate the ability of the neural network to retrieve the leakage model and maintain its interpretability. This approach will be considered in the rest of the paper. Using the orthonormal monomial basis has a major benefit. As shown by Kasper *et al.* [KSS10], when the basis is able to describe the switching activity of the circuit, the estimated basis coefficients highlight specific exploitable security flaws in the studied implementation. Hence, visualizing the basis coefficients that characterize the cVAE-SA model $F_{\Theta,\phi}$, namely Θ and ϕ, is useful to get deeper information on the exploitable security flaws. The next section proposes to visualize the trainable parameters Θ and ϕ in order to assess the suitability of the cVAE-SA to extract the expected leakage model ψ.

4.2 Leakage model estimation & multi-task learning

Single-sensitive variable attacks. As mentioned in Sec.3.2, the encoder (resp. decoder) is trained to retrieve the trainable parameters Θ (resp. ϕ) in order to maximize their correlation with the targeted leakage model. Once the cVAE-SA is correctly trained, the evaluator can visualize these trainable parameters (*i.e.* Θ and ϕ) in order to find the security flaws induced in the studied implementation. In the considered scenario (see Fig.3a), the weight visualization can be used to assess the ability of the encoder (resp. decoder) to retrieve the leakage function defined in Eq.11. Indeed, these figures illustrate the coefficients associated to each vector of the orthonormal monomial basis. The first coefficients of each figure define the lowest bits' interaction induced in the leakage model. For example, the first element, included in the interaction of degree 5 area, is characterized by $\oplus_{b=0}^{4} Y[b]$. While the related weight is non-negligible, the cVAE-SA identifies that the interaction $\oplus_{b=0}^{4} Y[b]$ influences the leakage model. This observation can be confirmed with Eq.11. Proceeding this analysis for the entire set of non-negligible weights can be helpful to evaluate the ability of the cVAE-SA to retrieve the leakage model. Indeed, if we compare the real simulated leakage model defined in Eq.11 with the non-negligible weights depicted in Fig.3a, we can see that all the peaks are associated with the correct basis vector. In addition, each coefficient associated with the sensitive interactions seems to get approximately the same impact which corresponds to the real leakage function defined in Eq.11. Consequently, if the cVAE-SA is correctly trained, it sounds helpful to retrieve complex leakage models as well as the related security flaws. Moreover, through the visualization provided in Fig.3a, the evaluator can also identify the time samples where the sensitive information leaks. Indeed, this figure highlights that only $\mathbf{T}[1]$ is useful to extract the leakage model. Hence, once the cVAE-SA is correctly trained, the evaluator can easily retrieve the PoIs. Then, during the attack phase, the evaluator can decide to focus its attack by computing Eq.10 only on $\mathbf{T}[1]$ instead of the entire trace dimension as mentioned in Sec.3.4. Contrary to classical profiled SCA approach, the cVAE-SA provides a more flexible solution during the exploitation phase



(a) Single-sensitive variable attacks (see Eq.11).

(b) Multi-task learning attacks (see Eq.12).

Figure 3: Weight visualization assessing the suitability of our generative model to retrieve the leakage model.

by reducing the impact of irrelevant samples. Another solution to assess the suitability of the training process consists in visualizing the leakage trace distributions (see App.A).

One advantage of the stochastic model is to approximate the data that depends on the secret key. Through this process, the evaluator directly obtains a score related to the key manipulated by the crypto-system. Hence, the evaluator can adapt the orthonormal monomial basis to target simultaneously multiple cryptographic primitives (e.g. input and output of the Sbox). Therefore, the cVAE-SA can be adapted to perform multi-tasking attacks. Additional experiments are proposed in App.B in order to assess the impact of the Gaussian noise on this visualization techniques.

Multi-task learning attacks. This approach has been in [Mag20] for enhancing the discriminative approach. While this learning strategy is beneficial to improve the performance of the key recovery phase, the design of such discriminative models remains an open question. Through this paragraph, we illustrate the flexibility of the cVAE-SA to deal with such solution. To exploit all the bits' interaction for each sensitive variable, we set $d = 8$ such that Fig.3b illustrates the impact of each vector of the basis $\mathcal{F}_9$ once the cVAE-SA is well trained. When the time sample $\mathbf{T}[1]$ is considered, we can see:

- An interaction of degree 1 and 2 that corresponds to the bit 5.
- An interaction between the bits 3 and 7 of the input of the Sbox.

Then, through Fig.3b, we can see that $\mathbf{T}[2]$ extracts a leakage model with two interactions of degree 1 associated with the 3^{rd} and the 6^{th} bit of the output of the Sbox. This result is consistent with Eq.12. Hence, through this simulation, we can validate the ability of the cVAE-SA to correctly retrieve the leakage model of multiple sensitive variables simultaneously. This approach is highly beneficial in SCA context as it gathers more information about the targeted secret key and results in a better performance. However, as mentioned in Sec.3.4, the degree d of the orthonormal monomial basis $\mathcal{F}_{d+1}$ directly affects the complexity of the cVAE-SA. Hence, considering the attacks of multi-sensitive variable increases by $2D \cdot (1 + \sum_{i=0}^{d} \binom{n}{i})$ the number of trainable parameters (*i.e.* weights) for each new targeted sensitive variable[j]. Thus, depending on his computational capacity,

[j]This number can be reduced to $2s' \cdot (1 + \sum_{i=0}^{d} \binom{n}{i})$ if the evaluator only targets the PoIs, denoted $\{l'_0, \ldots, l'_{s'-1}\}$.

Table 1: Impact of the trace dimension on the cVAE-SA performance (with $s = 1$, $N_v = 10$, batch-size = 10).

P_{RI}%	D	Learning rate	$\bar{N}t_{\text{rank}}$	Network complexity	Training time
33%	3	10^{-2}	98	$1,566$	$7s$
10%	10	10^{-2}	101	$5,360$	$8s$
2%	50	10^{-2}	108	$30,800$	$35s$
1%	100	10^{-2}	94	$71,600$	$47s$
0.2%	500	10^{-3}	115	$758,000$	$401s$
0.1%	1000	10^{-3}	135	$2,516,000$	$933s$

the evaluator has to define the most suitable structure to employ for defeating the targeted crypto-system.

However, in SCA context, the evaluator mainly deals with a non-negligible number of uninformative samples. The following section assesses the ability of the cVAE-SA to mitigate this constraint.

4.3 Curse of dimensionality

When the evaluator performs a side-channel attack, he wants to precisely find the relevant key-dependent time samples even if a large part of the trace contains uninformative time samples. Usually, the number of PoIs s is far lower than the trace dimension D (*i.e.* $s \ll D$). Thus, to assess the benefits of the cVAE-SA, we have to understand the ability of this new model to retrieve the PoIs when a lot of samples are irrelevant. In order to evaluate it under this restriction, we consider **Scenario 1** (see Sec.4.1) such that we construct 6 sub-scenarios where $D \in \{3, 10, 50, 100, 500, 1000\}$ and $s = 1$ such that the related *Signal-to-Noise Ratio* (SNR) equals to 0.549. Hence, for each case study, only a single PoI is configured while the dimension of the simulated traces increases. In Tab.1, we denote $P_{RI} = \frac{s}{D}$, the fraction of relevant information in each sub-scenario and evaluate the impact of this variable on other parameters, namely the batch-size and the learning rate. Finally, N_v denotes the number of samples $\mathbf{V}$ used to perform Eq.10.

As suggested in Sec.3.4, the attack process is performed only on the time samples defined as relevant[k] by the cVAE-SA. Hence, the weight visualization applied on ϕ and Θ is very helpful to define which samples can be considered as PoIs. Through Tab.1, we can see that increasing D does not impact significantly the resulted performance of the cVAE-SA (*i.e.* $\bar{N}t_{\text{rank}}$). Indeed, if the evaluator adequately finds the correct hyperparameters, namely batch-size and learning rate, he can expect to get similar results for high values of D. However, as detailed in Sec.3.4, increasing the input dimension highly impacts the complexity of the cVAE-SA. Finding a way to focus the interest of the model only on the relevant time samples can drastically reduce the network complexity without altering its resulted performance. Such investigations could be part of a future work to highlight even more the benefits of DL in SCA context.

Once the evaluator validates the ability of the generative model to deal with a low percentage of relevant information, he can question the benefits of the cVAE-SA to defeat boolean masking implementations. The next section deeply investigates this protection against this new model.

4.4 Generalization on boolean masking implementation

Typically, the discriminative models are built to automatically extract the relevant information from a trace without providing a clear interpretability of its decision-making. In

[k]Here, the relevance of a time sample is characterized by its coefficients ϕ and Θ such that the most relevant time samples have the highest coefficient values.

Table 2: Impact of boolean masking implementations on the conditional variational autoencoder performance (with batch-size = 64, $N_v = 10$).

Order	Learning rate	$\bar{N}t_{\text{rank}}$	Combining function	Network complexity
0	10^{-2}	9	Optimal product	2,630
1	10^{-2}	32	Optimal product	14,150
2	10^{-3}	100	Optimal product	95,750
3	10^{-3}	247	Absolute difference	1,103,750

opposition, the cVAE-SA characterizes the leakage model without altering its representation in order to get a global characterization of $\mathsf{Pr}[\mathbf{T}|Y]$. Consequently, our generative approach does not automatically recombine the PoIs but preserves the network's explainability that is mandatory for an evaluator. Thus, given a masked implementation, some preprocessing phases are needed. In particular, a masking scheme of order o consists in splitting the targeted sensitive variable into $(o+1)$ shares $Y_0, \ldots, Y_o$ which satisfy $Y = Y_0 + \ldots + Y_o$ over some additive finite group or field. To this end, o shares are randomly drawn (known as *masks*) while the remaining one (known as *masked variable*) is computed in order to satisfy the latter relation. Consequently, to perform a successful high-order attack, the evaluator has to find the $(o+1)$ shares in order to retrieve the sensitive information Y. Typically, classical approaches considered in profiled SCA use some *recombination techniques*[l] as preprocessing [CJRR99, Mes00, PRB09]. This approach involves the combination of $(o+1)$ shares in order to "*demask*" the masked values and perform the attacks on the unmasked value. To apply this proposition, various recombination techniques are introduced, namely *product combining* [CJRR99], *absolute difference combining* [Mes00] and *optimal product combining* [PRB09]. If the evaluator wants to apply one of these techniques, he has to recombine the samples related to each of the $(o+1)$ shares and then, target the unmasked sensitive value Y.

To evaluate the suitability of cVAE-SA in such scenarios, we decide to simulate a 5-dimensional trace with different levels of masking order $o \in \{0, 1, 2, 3\}$. For each case study, we apply the absolute difference, the product and the optimal product combining functions and list the best result we obtained in Tab.2. Through this table, we demonstrate the ability of the proposed cVAE-SA to defeat a high-order boolean masking implementation. Surprisingly, the number of shares does not highly impact the hyperparameters' value[m], namely the learning rate and the batch-size, unlike the network complexity. Indeed, for a given set of D-dimensional traces, the combining methods multiplied by D the number of time samples for each mask reduction. Hence, for performing an o order attack, the evaluator has to deal with traces of D^{o+1} samples. As the dimension of the traces impacts the network complexity (see Sec.3.4), the evaluator has to exponentially increase his computational ability with the attack order.

Once all these simulations validate the theoretical observations provided in Sec.3, we compare the benefits of considering the new cVAE-SA with the classical profiled side-channel attacks on real unprotected and protected implementations.

5 Experimental results

The experiments are implemented in Python using the *Keras* library and are run on a workstation equipped with 128GB RAM and a NVIDIA GTX1080Ti with 11GB memory. In the following section, the discriminative models are based on the CNN architectures provided by [ZBHV19] and then, a global benchmark is provided with other typology of

[l] An alternative consists in applying a Bayes classification approach [OM06] in order to retrieve the targeted value.

[m] The readers must be aware that this observation cannot be generalized on all implementations and would benefit from further investigations.

discriminative models (see Tab.4). For the generative models, the configurable hyperparameters, namely the batch-size and the learning rate, are respectively set to $\{8, 16, 32, 64\}$ and $\{10^{-1}, 10^{-2}, 10^{-3}\}$. We construct each model with the following number of epochs $\{10, 20, 30, 40, 50, 75, 100\}$ and select the value that provides the best rank. As mentioned in Sec.4, we denote $\bar{Nt}_{\text{rank}}$ the average value of Nt_{rank} over 10 shuffled experiments. In the following, we always capture the maximum amount of interactions (*i.e.* $\mathcal{F}_9$). This choice was made because an evaluator does not have *a priori* knowledge on the bits' interactions[n]. Finally, as suggested through the analysis of the KL-divergence loss (see Sec.3.3), the latent space dimension should be monitored depending on the number of PoIs. As the goal of our paper is to provide a fair comparison with the state-of-the-art result, the same number of PoIs as in [KPH+19, BPS+20] will be considered.

5.1 Presentation of the datasets

We used three different datasets for our experiments. All the datasets correspond to implementations of *Advanced Encryption Standard* (AES). The datasets offer a wide range of use cases: high-SNR unprotected implementation on a smart card, low-SNR unprotected implementation on a FPGA, low-SNR protected implementation with first-order masking.

- **DPA contest-v4**[o] is an AES software implementation with a first-order masking. Knowing the mask value, we can consider this implementation as unprotected and recover the secret key directly. In this experiment, we attack the first round Sbox operation. We identify each trace with the sensitive variable $Sbox[X[0] \oplus k^*] \oplus M$ where M denotes the known mask and $X[0]$ the first byte of the plaintext.

- **AES_HD**[p] is an unprotected AES-128 implemented on FPGA. The attack targets the register writing in the last round such that the label of the i^{th} trace is $Sbox^{-1}[C[j] \oplus k^*] \oplus C[j']$ where $C[j]$ and $C[j']$ are two ciphertext bytes such that $j = 12$ and $j' = 8$.

- **ASCAD-v1**[q] is introduced in [BPS+20]. The target platform is an 8-bit AVR microcontroller (ATmega8515) where a AES-128 protected with a boolean masking scheme is implemented. The targeted sensitive variable is the first round Sbox operation such that $Y = Sbox[X[3] \oplus k^*]$. Currently, there are two versions of the ASCAD dataset. The distinction between these versions relies on the randomness of the secret key for the profiling traces. In particular, the **ASCAD-v1-F** version has a fixed secret key for the $50,000$ profiling traces and the $10,000$ attack traces. Each trace of this dataset is composed of 700 samples. On the other hand, the **ASCAD-v1-R** version has random keys for the $200,000$ profiling traces and a fixed key for the $100,000$ attack traces. In the **ASCAD-v1-R** version, each trace is composed of $1,400$ samples.

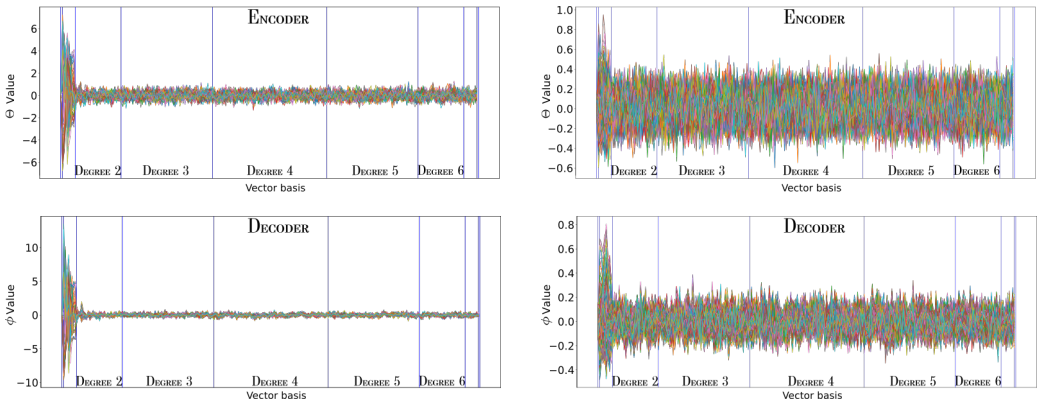
Remark 2. While this work bridges DL with SCA, no investigation has been conducted on the desynchronization effect. Indeed, as the Machine Learning community has already demonstrated the benefits of the use of shift-invariant layers (e.g. convolutional layers) to mitigate the desynchronization effect [ITLW20], further theoretical investigations should be provided to clearly explain how those layers should be configured regarding the works

[n] In order to find the best trade-off between bits' interactions and the statistical model, one solution consists in evaluating the model quality of a linear regression model. In [MOW17], McCann *et al.* suggest the use of the F-statistic in order to test the improvement of considering additional interaction terms.

[o] http://www.dpacontest.org/v4/42_traces.php

[p] https://github.com/AESHD/AES_HD_Dataset

[q] https://github.com/ANSSI-FR/ASCAD



(a) DPA contest-v4 dataset. **(b)** AES_HD dataset.

Figure 4: Weight visualization of 50 time samples assessing the suitability of our generative model to retrieve the leakage model.

on dimensionality reduction in SCA (e.g. [BGH+15]). This investigation is out of the scope of this paper and should be part of a future work.

5.2 A comparison with state-of-the-art SCA

In this section, we evaluate the benefits of the cVAE-SA against the classical side-channel attacks (*i.e.* template attacks, stochastic attacks) by respecting the same experimental conditions as the state-of-the-art results. For the DPA contest-v4 and AES_HD datasets, Kim *et al.* [KPH+19] propose to select 50 features with the highest SNR in order to reduce the needs of computation when classical side-channel attacks are considered. Following Sec.3.4, the related cVAE-SA architecture is composed by $30,800$ parameters for both datasets (*i.e.* $d = 8$, $s = 50$ and $n = 8$). For both ASCAD-v1 datasets, we select the 8 most relevant samples related to the mask r_3 and the masked values with r_3 (see [BPS+20] for deeper details on the implementation) and then, apply the three combining functions introduced in Sec.4.4. After the application of the recombination technique, the generated cVAE-SA is composed by $41,216$ parameters (see Sec.3.4 such that $d = 8$, $s = 64$ and $n = 8$).

DPA contest-v4. Once the cVAE-SA is trained, the evaluator can observe the coefficients related to each time sample as illustrated in Fig.4a. Through this visualization tool, the evaluator is able to identify the leakage model extracted by the cVAE-SA. In particular, it is can be observed that the leakage model is only influenced by the bits of Y. Therefore, the bits' interaction do not have any impact of the leakages extracted from the DPA contest-v4 dataset. Once this analysis is conducted, the evaluator can select those with the highest trainable parameters (*i.e.* Θ and ϕ) and perform his attack on this subset. This post-selection is beneficial to reduce even more the impact of noisy samples (*i.e.* time samples where the related weights are close to 0) during the key-recovery phase. This new feature can be proposed and explained thanks to the interpretability of the cVAE-SA architecture (see Sec.3). For this dataset, we compute Eq.10 on the 50 time samples previously extracted. When a high-SNR unprotected implementation is considered, we observe that our generative model has the same performance as classical profiled side-channel attacks (see Tab.3). Hence, for this implementation, similar results can be obtained whatever the attack performed. Consequently, in this configuration, considering the cVAE-SA is equivalent to classical profiled side-channel attacks.

Table 3: Comparison of $\bar{N}t_{\text{rank}}$ value depending on datasets. The best performance for each dataset is denoted in blue.

	Stochastic Attacks	**Template Attacks**	**cVAE-SA** [This work]
DPA-contest v4	**4**	**4** [KPH+19]	**4**
AES_HD	4,500	25,000 [KPH+19]	**300**
ASCAD-v1-F	290	351	**194**
ASCAD-v1-R	1,330	2,850	**250**

AES_HD. Following the state-of-the-art results [KPH+19], a template attack needs approximately 25,000 attack traces to retrieve the secret key. In this setting, performing the stochastic attack on the same dataset highly improves the related performance. Indeed, when this approach is considered, the evaluator can recover the secret key with 4,500 attack traces which is 5.5 times better. Finally, when the cVAE-SA is applied, an even better attack can be performed. As mentioned in Sec.3.4, the attack phase is based on sample similarity measures. Hence, the evaluator can only compute Eq.10 on the relevant samples detected during the profiling phase. Thus, we drastically reduce the impact of the uninformative samples during the attack phase. In this configuration, we only consider the samples where the related ϕ coefficients are greater than 0.5. Through Fig.4b, it can be mentioned that the key-recovery phase is only impacted by the leakages related to some bits of Y. Accordingly, while the training process was performed on traces with 50 samples, the computation of Eq.10 was made on the 14 time samples complying with the configured restriction. This processing tremendously increases the performance of the resulted attack. Indeed, the cVAE-SA model divides by 83 (resp. 15) the number of attack traces that are needed to perform a template attack (resp. stochastic attack).

ASCAD-v1. As mentioned in Sec.4.4, we perform high-order attacks with the help of combining functions as preprocessing (*i.e. product combining*, *optimal product combining*, *absolute difference combining*) for both ASCAD-v1 datasets. Then, we profile the generative models on the unmasked value in order to extract the relevant information. In Tab.3, the optimal product combination provides the best performance on the ASCAD-v1-F and ASCAD-v1-R datasets. Through the experiment on ASCAD-v1-F dataset, we observe that the cVAE-SA performs better than template or stochastic attacks. While 351 (resp. 290) attack traces are needed to reach a constant rank of 1 when the template attack (resp. stochastic attack) is considered, our generative model retrieves the secret key within 194 attack traces. The same observation can be highlighted for ASCAD-v1-R dataset where the cVAE-SA model retrieves the secret key within 250 attack traces. As previously mentioned for the AES_HD dataset, those results can be explained by the ability of the cVAE-SA to target a specific range of relevant combined time samples during the attack phase. For both ASCAD-v1 datasets, only the time samples with Θ and ϕ coefficients greater than 1 are kept for the key recovery phase. On the contrary, the classical profiled SCA have to consider the 64 time samples (*i.e.* 8 time samples related to the masks and the masked value) used to perform the related attacks. Hence, resulted noisy time samples can highly influence the performance of the resulted attacks. A detailed discussion on ASCAD-v1-R dataset with explainability/interpretability results is provided in App.C.

In conclusion, when a classical profiled SCA is trained on D-dimensional traces, the evaluator has to perform the exploitation phase on the same trace dimension. Unfortunately, the evaluator does not know *a priori* which time samples are considered as relevant once the profiling phase is applied. Hence, performing the exploitation phase on the D-dimensional traces could be impacted by the uninformative time samples. On the other hand, once the profiling phase is performed on the D-dimensional traces, the cVAE-SA is beneficial

to select a subset of s time samples, such that $s \ll D$, in order to compute Eq.10 only on the informative time samples. Hence, this new proposition is more flexible than classical profiled SCA and results in a better attack perspective as we are less impacted by uninformative time samples. However, the evaluator can question the benefits of the generative approach with respect to discriminative models. The following section highlights the benefits and the limitations of both approaches in DLSCA.

5.3 A comparison with state-of-the-art DLSCA

When the discriminative approaches are considered, a major drawback can be highlighted regarding the architecture configuration. Indeed, the resulted models have a plethora of hyperparameters to tune. The more effort we spend on the hyperparameter tuning of the network architecture, the more efficient the resulted attack is expected. In addition, due to their black-box property, the discriminative models are difficult to interpret. However, the main benefit of this approach is about automatically combining the points of interest to limit the masking effect. While the discriminative model considers all the samples of the trace, we focus the interest of the generative model only on the most relevant samples. As highlighted in Sec.4.3, increasing the number of irrelevant samples highly impacts the network complexity and the training time without altering the related performance.

Remark 3. All the cVAE-SA architectures use in this section are similar to those introduced in Sec.5.2.

DPA contest-v4. First, on Fig.5a, we can visualize the rank evolution of the generative on the DPA contest-v4 dataset. While a discriminative model can retrieve the secret key with 3 attack traces (see Tab.4), the cVAE-SA reaches similar performance depending on the number N_v of latent samples. As mentioned in Sec.3.4, the value of N_v depends on the ability of the cVAE-SA to correctly approximate $\mathsf{Pr}[\mathbf{T}|Y,\phi]$. The higher the N_v, the more confident the resulted attack. This observation can be confirmed in Fig.5a. Indeed, if $N_v = 1$, two attack traces are needed to retrieve the secret key. However, a poor rank stabilization is observed. To rectify this point, increasing the N_v value preserves a constant rank convergence towards 1.

AES_HD. The same observation can be made when the AES_HD dataset is considered. Indeed, when the N_v value increases, a rank stabilization is observed when the number of attack traces grows. In addition, Fig.5b highlights a better model when the generative approach is considered in comparison with the discriminative state-of-the-art result (see Tab.4). Indeed, for $N_v = \{100; 1,000\}$, the resulted model converges towards a constant rank of 1 with 300 attack traces. Even if the discriminative approach directly estimates $\mathsf{Pr}[Y|\mathbf{T}]$, the state-of-the-art result indicates a lower performance when most of classical DLSCA models are considered. As illustrated by Ng and Jordan [NJ02], this result suggests that a better discriminative model can be found on this dataset. Indeed, an optimal discriminative model should be, at least, as efficient as a generative approach. However, finding the best discriminative model can be difficult due to the broad hyperparameter selection. This result highlights the benefits of the cVAE-SA in comparison with the classical DLSCA models from an evaluation perspective as it provides a suited security bound related to the targeted device.

ASCAD-v1. One benefit of the discriminative approach is to automatically recombine the points of interest. In opposition, the generative approach does not take advantage of this property (see Sec.4.4). Through Tab.4, we can visualize the benefits of automatically combining the points of interest. Indeed, the discriminative approach reaches better performance for both datasets (*i.e.* ASCAD-v1-F and ASCAD-v1-R). While the cVAE-SA

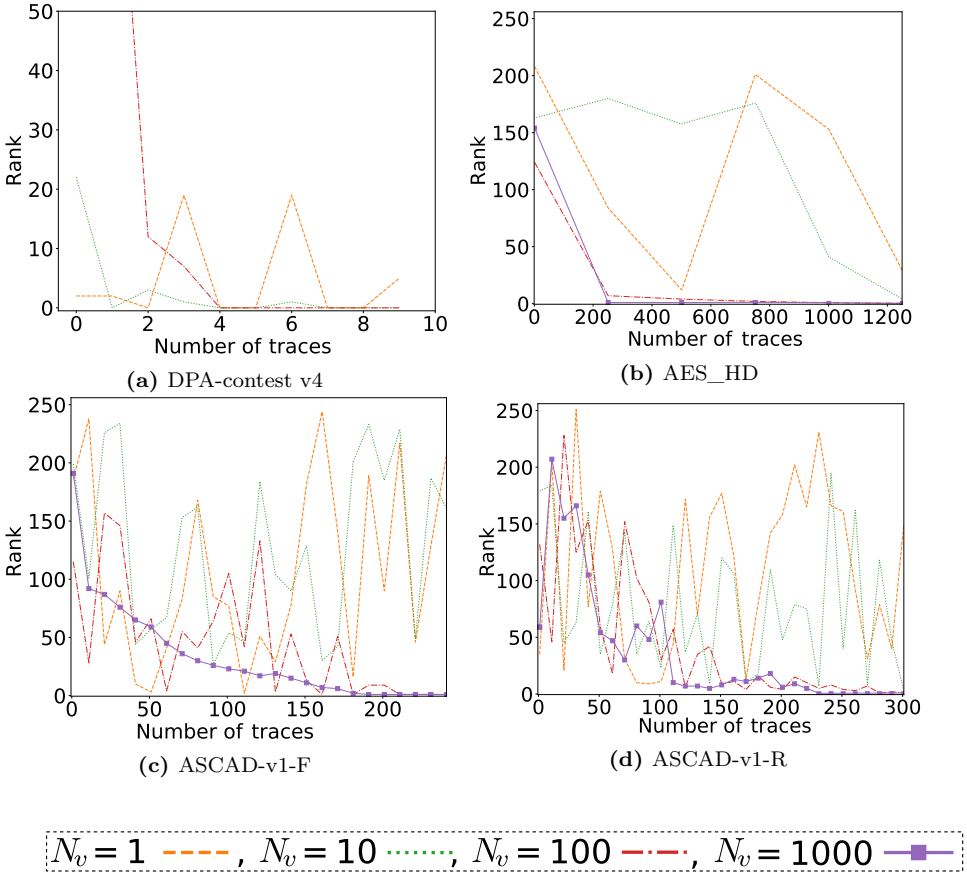


Figure 5: Mean rank evolution for cVAE-SA models depending on the number of latent representation N_v.

Table 4: DLSCA benchmark of $\bar{N}t_{\text{rank}}$ value depending on datasets. The best performance for each dataset is denoted in blue.

	Discriminative model						Generative model
	CNN	**FCNN** (or **MLP**)	**ResNet**	**TransNet**	**SVM**	**Random Forest**	**cVAE-SA** [This work]
DPA contest-v4	**3** [ZBHV19]	4 [PCP20, PHJ+18]	**3** [JZHY20]	-	**3** [PHJ+18]	5 [PHJ+18]	4
AES_HD	831 [ZXF+19]	350 [MDP19]	2,100 [JZHY20]	-	6,653 [PHJ+18]	2,877 [PHJ+18]	**300**
ASCAD-v1-F	**87** [PWP22]	104 [PWP22]	552 [JZHY20]	300 [HSAM22]	-	-	194
ASCAD-v1-R	**78** [WPP21]	129 [PWP22]	-	-	-	-	250

retrieves the secret key within 194 traces (resp. 250 traces) with $N_v = \{100; 1,000\}$ when ASCAD-v1-F (resp. ASCAD-v1-R) is considered, the best discriminative model recovers this sensitive variable within 87 traces (resp. 78 traces). This result could be explained by the ability of the discriminative models to find a custom combining function that maximizes the posterior probabilities $\mathsf{Pr}[Y|\mathbf{T}]$. Hence, this custom unknown function can be more adapted for the targeted dataset. On the other hand, the cVAE-SA model is trained on combined traces that are constructed from classical approaches (*i.e.* optimal product combining). Consequently, when masking implementations are considered, a discriminative approach is beneficial to reduce the preprocessing phase and, it can provide better result than the cVAE-SA. However, applying the discriminative approach can be limited from an interpretation point of view. In addition, the discriminative approach required plethora of additional settings (*i.e.* architecture, activation function, weight initialization, etc.) that do not have to be considered when the cVAE-SA is constructed. Hence, the configuration of discriminative models can be an issue from a practical perspective.

The results provided in this section highlight the benefits and the limitations of cVAE-SA against classical DLSCA models. Particularly, Tab.4 refers the main models introduced in the DLSCA literature. Through this benchmark, it can be mentioned that the cVAE-SA is the only model which considers the generative approach such that it performs similarly, or even better, than classical DLSCA models. As suggested by Ng and Jordan [NJ02], the asymptotic error of the discriminative model is lower or equal to the one related to the generative approach. Therefore, discriminative models should be at least as efficient as a cVAE-SA. However, as their construction phase is not deterministic, an irrelevant model can be designed to solve a given classification task and the resulted performance can be less efficient than the cVAE-SA due to a poor approximation of the true unknown leakage model. This observation can be confirmed through the results provided in Tab.4.

To conclude, for the SCA community, the cVAE-SA can be helpful to evaluate the feasibility of an attack and get a security bound of a device. However, while the configuration of such neural network is simple in comparison to DLSCA models, this new proposition can perform worse under some conditions. Indeed, when masking coutermeasures are implemented, an evaluator using generative models (e.g. cVAE-SA) can use sub-optimal combining functions while a discriminative approach finds a custom unknown combining function which can be more adapted depending on the targeted implementation. In addition, while the cVAE-SA suggests that the true leakage distribution is Gaussian, a discriminative approach is not restricted to such assumption. As a perspective, we suggest considering a hybrid approach combining the discriminative and the generative models in order to keep the explainability and the interpretability while preserving the benefits of the automatic recombination.

6 Discussion

Through this paper, we have demonstrated that a derivation of the conditional variational autoencoders (cVAE) can be considered in side-channel context in order to perform physical attacks. From an evaluation perspective, this new neural network architecture is suitable as it respects the following requirements:

1. **Theoretical similarities with classical profiled side-channel attacks** – As illustrated in Sec.3, the cVAE can be monitored to fit with the stochastic attacks paradigm introduced by Schindler *et al.* [SLP05] and briefly recalled in Sec.2.3. From the evaluator point of view, this approach is useful to ease the configuration of the neural network and get a clear overview of the decision-making process. Indeed, as the cVAE-SA is designed on well-known theoretical attack strategy, the evaluator can be confident on the employed neural network structure and thus, expects to get a resulted predictive model as efficient as classical profiled side-channel attacks, namely template attacks [CRR03] and stochastic attacks [SLP05]. Based on the solution provided in [Mag20], the cVAE-SA can be easily adapted to deal with the multi-task learning process that consists in targeting simultaneously multiple sensitive variables. From this new bridge, the evaluator can deeply understand the future improvements that can be provided in the DLSCA field in order to fully exploit the automation process proposed by the ML/DL community.

2. **Explainability & Interpretability** – One major benefit of the proposed cVAE-SA is to preserve the interpretability and the explainability on the results provided by the learning algorithm. As our contribution is constructed from the classical profiled side-channel attacks, the evaluator can adapt its interpretation tools (e.g. visualization) in order to deeply explain the results provided by the model. As suggested in Sec.4.2, in App.A and in App.B, the evaluator can visualize the trainable parameters of the conditional variational autoencoder in order to assess the ability of the encoder and the decoder to retrieve a hypothetical leakage model as similar as possible to the true unknown ψ. Once the evaluator retrieves an approximation of ψ, he highlights the security flaws induced by the targeted implementation and thus, can alert the developer on potential vulnerabilities and ease the development of countermeasures. An example on the ASCAD-v1-R dataset is provided in App.C.

3. **Hiding countermeasures** – Even if this paper does not assess the robustness of the cVAE-SA against desynchronization effect, an intuitive solution suggests adding convolutional layers to the encoder [ITLW20]. However, it should be validated in practice. While this intuition could be a suitable solution to mitigate the desynchronization effect, it also helps the network to automatically select the points of interest and prevent the effect of uninformative time samples. Indeed, as defined in Sec.3.3 and in Sec.3.4, the empirical risk as well as the decision process are only affected by the points of interest. Hence, this dimensionality reduction technique can also be useful to quadratically reduce the network complexity. However, to maintain the interpretability/explainability result provided by the cVAE-SA proposal, further investigations should clarify how those convolutional layers should be designed to fit with the side-channel state-of-the-art result (e.g. [BGH+15]).

However, the cVAE-SA also has some limitations that are listed below:

1. **Combining function** – As the generative approach captures the conditional distribution $\mathsf{Pr}[\mathbf{T}|Y]$, it cannot handle masking implementations as the targeted unmasked sensitive variable Y is not directly observable through the leakage trace $\mathbf{T}$. Thus, the evaluator has to consider combining functions in order to reveal the dependence

between $\mathbf{T}$ and Y. Unfortunately, this suggests the need for preprocessing phase which is not necessarily optimal from an attack perspective. Indeed, as this combining function is not automatically learned by the generative model (contrary to the discriminative approach), the evaluator may not converge towards the optimal statistical model defined in [HRG14].

2. **Performance** – While the goal of a side-channel attack is to optimize a learning algorithm which approximates $\mathsf{Pr}[Y|\mathbf{T}]$ in order to discriminate a sensitive variable Y from a set $\mathcal{Y}$, the application of generative models can be considered as suboptimal [NJ02]. In particular, the cVAE-SA assumes that the leakage noise follows a Gaussian law which is not the case for classical DLSCA discriminative models [MPP16, CDP17a, CCC+19, ZBHV19, BPS+20, MS21].

However, regarding the latter issue, the experiment provided in Sec.5.3 illustrates that similar (or even better) performance results can be obtained regardless the approach (*i.e.* discriminative vs. generative) considered. Indeed, the actual publicly available datasets seem too easy to target (*i.e.* the number of traces to retrieve the secret key is low) in order to fully assess the performance gain of a discriminative approach over a generative one. A slight performance gains of few traces or, even dozens of traces, cannot be considered as a huge improvement and the benefits of each new DLSCA tool can be difficult to interpret. Through this analysis, we highlight the theoretical benefits/limitations of using the cVAE-SA and we define this solution as a concrete and generic alternative to the classical discriminative DLSCA models. Consequently, using the generative approach can give a first insight about the security bound of the targeted system.

7 Conclusion

This paper proposes to reduce the gap between historical SCA (*i.e.* generative models) and classical DLSCA (*i.e.* discriminative models). In that purpose, we introduce the first DLSCA model based on generative approach. From the stochastic attack introduced by Schindler *et al.* [SLP05], we first design an explainable and interpretable architecture that aims at retrieving the real unknown leakage model. Based on stochastic attack modeling, this new model can be easily constructed whatever the implementation an evaluator has to deal with. Furthermore, this analogy helps us to define theoretical bounds on the network complexity (e.g. number of trainable parameters) as well as identifying mutual problematic and perspectives (e.g. dimensionality reduction, multi-task learning). Then, we theoretically explain the impact on each individual loss in SCA such that, the reconstruction loss penalizes the model in order to estimate a trace as similar as possible to the real one. On the other hand, we demonstrate that the KL-divergence loss is beneficial to correctly estimate the latent space. Compared with historical profiled SCA, the cVAE-SA is beneficial by providing the ability to carefully select the samples the evaluator wants to focus on during the exploitation phase. Hence, by providing a more flexible generative approach, we drastically reduce the impact of uninformative samples on the attack performance. This observation was confirmed on real case study.

To bridge the gap between generative and discriminative approaches, we conduct experiments on simulations and public datasets on a wide range of use cases and observe that the generative approach does not perform worse than a discriminative one. As suggested by Ng and Jordan [NJ02], the discriminative models should be at least as efficient as the generative ones. However, as their construction phase is time consuming and not deterministic, an irrelevant model can be designed by an evaluator and the resulted performance can be less efficient than the cVAE-SA due to a poor approximation of the true unknown expected leakage model. Therefore, considering the cVAE-SA is a good starting point to define a security bound related to the targeted device. However, on the other

hand, using the discriminative approach seems beneficial when masked implementations are targeted because more appropriate unknown combining function can be automatically retrieved by the related model. This solution cannot be considered with generative models. Thus, depending on the time he wants to spend on the construction phase, the evaluator has to select the best way to mount his supervised attacks.

All these results suggest a lot of future works. First, as the cVAE-SA is derived from the Stochastic Attacks, further investigations can be conducted on this new model in order to extend the work provided by Choudary *et al.* [CK15] which consists in performing profiled attacks beyond 8 bits. Then, as the generative approach aims at approximating a joint distribution between two random variables, we have to assess the suitability of using the cVAE-SA as a model to perform non-profiled SCA, or even more generally, blind SCA. In addition, while the limitations of the discriminative (resp. generative) approach seem solved by the generative (resp. discriminative) approach, one solution could be to consider a hybrid model that preserves the automatic sample recombination property (*i.e.* discriminative approach) while keeping the explainability/interpretability and reducing the hyperparameter selection (*i.e.* generative approach). Finally, while the discriminative approach does not make any assumption on the noise distribution, its application is more generic. A solution to enhance the cVAE-SA approach consists in configuring other latent spaces, maybe more generic than the Gaussian one proposed in this paper. Those suggestions can be considered as an additional step towards the use of generative machine learning models in the side-channel context.

Acknowledgement

The authors would like to thank Shivam Bhasin and the anonymous reviewers for their valuable comments which helped to improve this work.

References

[APSQ06] Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template attacks in principal subspaces. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, pages 1–14, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[BGH+15] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Damien Marion, and Olivier Rioul. Less is more. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems – CHES 2015*, pages 22–41, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[BPS+20] Ryad Benadjila, Emmanuel Prouff, Rémi Strullu, Eleonora Cagli, and Cécile Dumas. Deep learning for side-channel analysis and introduction to ASCAD database. *Journal of Cryptographic Engineering*, 10(2):163–188, 2020.

[Car10] Claude Carlet. *Boolean Functions for Cryptography and Error-Correcting Codes*, page 257–397. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2010.

[CCC+19] Mathieu Carbone, Vincent Conin, Marie-Angela Cornélie, François Dassance, Guillaume Dufresne, Cécile Dumas, Emmanuel Prouff, and Alexandre Venelli. Deep learning to evaluate secure rsa implementations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(2):132–161, Feb. 2019.

[CDP17a] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures - profiling attacks without pre-processing. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 45–68. Springer, 2017.

[CDP17b] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Kernel discriminant analysis for information extraction in the presence of masking. In Kerstin Lemke-Rust and Michael Tunstall, editors, *Smart Card Research and Advanced Applications*, pages 1–22, Cham, 2017. Springer International Publishing.

[CJRR99] Suresh Chari, Charanjit Jutla, Josyula Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO' 99*, pages 398–412, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[CK15] Marios Choudary and Markus Kuhn. Efficient stochastic methods: Profiled attacks beyond 8 bits. In Marc Joye and Amir Moradi, editors, *Smart Card Research and Advanced Applications*, pages 85–103, Cham, 2015. Springer International Publishing.

[CRR03] Suresh Chari, Josyula Rao, and Pankaj Rohatgi. Template attacks. In *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '02, pages 13–28, London, UK, UK, 2003. Springer-Verlag.

[DSVC14] François Durvaux, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to certify the leakage of a chip? In Phong Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, pages 459–476, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[GDG+15] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1462–1471, Lille, France, 07–09 Jul 2015. PMLR.

[GHMR17] Sylvain Guilley, Annelie Heuser, Tang Ming, and Olivier Rioul. Stochastic side-channel leakage analysis via orthonormal decomposition. In Pooya Farshim and Emil Simion, editors, *Innovative Security Solutions for Information Technology and Communications*, pages 12–27, Cham, 2017. Springer International Publishing.

[GJS20] Aron Gohr, Sven Jacob, and Werner Schindler. Subsampling and knowledge distillation on adversarial examples: New techniques for deep learning based side channel evaluations. Cryptology ePrint Archive, Report 2020/165, 2020. https://eprint.iacr.org/2020/165.

[HMP+17] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burges, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

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

[HSAM22] Suvadeep Hajra, Sayandeep Saha, Manaar Alam, and Debdeep Mukhopadhyay. Transnet: Shift invariant transformer network for side channel analysis, 2022.

[ISUH21] Akira Ito, Kotaro Saito, Rei Ueno, and Naofumi Homma. Imbalanced data problems in deep learning-based side-channel attacks: Analysis and solution. *IEEE Transactions on Information Forensics and Security*, pages 1–1, 2021.

[ITLW20] Maximilian Ilse, Jakub M. Tomczak, Christos Louizos, and Max Welling. Diva: Domain invariant variational autoencoders. In Tal Arbel, Ismail Ben Ayed, Marleen de Bruijne, Maxime Descoteaux, Herve Lombaert, and Christopher Pal, editors, *Proceedings of the Third Conference on Medical Imaging with Deep Learning*, volume 121 of *Proceedings of Machine Learning Research*, pages 322–348. PMLR, 06–08 Jul 2020.

[IUH22] Akira Ito, Rei Ueno, and Naofumi Homma. Perceived information revisited: New metrics to evaluate success rate of side-channel attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(4):228–254, Aug. 2022.

[JZHY20] Minhui Jin, Mengce Zheng, Honggang Hu, and Nenghai Yu. An enhanced convolutional neural network in side-channel attacks and its visualization. *CoRR*, abs/2009.08898, 2020.

[KPH+19] Jaehun Kim, Stjepan Picek, Annelie Heuser, Shivam Bhasin, and Alan Hanjalic. Make some noise. unleashing the power of convolutional neural networks for profiled side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(3):148–179, May 2019.

[KSS10] Michael Kasper, Werner Schindler, and Marc Stöttinger. A stochastic method for security evaluation of cryptographic fpga implementations. In *2010 International Conference on Field-Programmable Technology*, pages 146–153, 2010.

[KW14] Diederik Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[KW19] Diederik Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[KWKT15] Tejas Kulkarni, William Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In Corinna Cortes, Neil Lawrence, Daniel Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[LLY+20] Guanlin Li, Chang Liu, Han Yu, Yanhong Fan, Libang Zhang, Zongyue Wang, and Meiqin Wang. Scnet: A neural network for automated side-channel attack. *CoRR*, abs/2008.00476, 2020.

[LZC+21] Xiangjun Lu, Chi Zhang, Pei Cao, Dawu Gu, and Haining Lu. Pay attention to raw traces: A deep learning architecture for end-to-end profiling attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(3):235–274, Jul. 2021.

[Mag19] Houssem Maghrebi. Assessment of common side channel countermeasures with respect to deep learning based profiled attacks. In *2019 31st International Conference on Microelectronics (ICM)*, pages 126–129, 2019.

[Mag20] Houssem Maghrebi. Deep learning based side-channel attack: a new profiling methodology based on multi-label classification. Cryptology ePrint Archive, Report 2020/436, 2020. https://eprint.iacr.org/2020/436.

[MCHS22] Loïc Masure, Gaëtan Cassiers, Julien Hendrickx, and François-Xavier Standaert. Information bounds and convergence rates for side-channel security evaluators. Cryptology ePrint Archive, Paper 2022/490, 2022. https://eprint.iacr.org/2022/490.

[MDP19] Loïc Masure, Cécile Dumas, and Emmanuel Prouff. A comprehensive study of deep learning for side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(1):348–375, Nov. 2019.

[Mes00] Thomas Messerges. Using second-order power analysis to attack dpa resistant software. In Çetin Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, pages 238–251, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[MHM14] Zdenek Martinasek, Jan Hajny, and Lukas Malina. Optimization of power analysis using neural network. In Aurélien Francillon and Pankaj Rohatgi, editors, *Smart Card Research and Advanced Applications*, pages 94–107, Cham, 2014. Springer International Publishing.

[MOW17] David McCann, Elisabeth Oswald, and Carolyn Whitnall. Towards practical tools for side channel aware software engineering: 'grey box' modelling for instruction leakages. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 199–216, Vancouver, BC, August 2017. USENIX Association.

[MPP16] Houssem Maghrebi, Thibault Portigliatti, and Emmanuel Prouff. Breaking cryptographic implementations using deep learning techniques. In Claude Carlet, Anwar Hasan, and Vishal Saraswat, editors, *Security, Privacy, and Applied Cryptography Engineering - 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings*, volume 10076 of *Lecture Notes in Computer Science*, pages 3–26. Springer, 2016.

[MS21] Loïc Masure and Rémi Strullu. Side channel analysis against the anssi's protected aes implementation on arm. Cryptology ePrint Archive, Report 2021/592, 2021. https://eprint.iacr.org/2021/592.

[MZ13] Zdenek Martinasek and Vaclav Zeman. Innovative method of the power analysis. *Radioengineering*, 22(2):586–594, 2013.

[NJ02] Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Thomas Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2002.

[OM06] Elisabeth Oswald and Stefan Mangard. Template attacks on masking—resistance is futile. In Masayuki Abe, editor, *Topics in Cryptology – CT-RSA 2007*, pages 243–256, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[PCP20] Guilherme Perin, Lukasz Chmielewski, and Stjepan Picek. Strength in numbers: Improving generalization with ensembles in machine learning-based profiled side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(4):337–364, Aug. 2020.

[PHJ+18] Stjepan Picek, Annelie Heuser, Alan Jovic, Shivam Bhasin, and Francesco Regazzoni. The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(1):209–237, Nov. 2018.

[Pin99] Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1999.

[PRA20] Servio Paguada, Unai Rioja, and Igor Armendariz. Controlling the deep learning-based side-channel analysis: A way to leverage from heuristics. In Jianying Zhou, Mauro Conti, Chuadhry Mujeeb Ahmed, Man Ho Au, Lejla Batina, Zhou Li, Jingqiang Lin, Eleonora Losiouk, Bo Luo, Suryadipta Majumdar, Weizhi Meng, Martín Ochoa, Stjepan Picek, Georgios Portokalidis, Cong Wang, and Kehuan Zhang, editors, *Applied Cryptography and Network Security Workshops*, pages 106–125, Cham, 2020. Springer International Publishing.

[PRB09] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical analysis of second order differential power analysis. *IEEE Transactions on Computers*, 58(6):799–811, 2009.

[PWP22] Guilherme Perin, Lichao Wu, and Stjepan Picek. Exploring feature selection scenarios for deep learning-based side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(4):828–861, Aug. 2022.

[RWPP21] Jorai Rijsdijk, Lichao Wu, Guilherme Perin, and Stjepan Picek. Reinforcement learning for hyperparameter tuning in deep learning-based side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(3):677–707, Jul. 2021.

[SLP05] Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In Josyula Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2005*, pages 30–46, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[SLY15] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In Corinna Cortes, Neil Lawrence, Daniel Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[SSB17] Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 627–637, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[Vap98] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[Wei20] Léo Weissbart. Performance analysis of multilayer perceptron in profiling side-channel analysis. In Jianying Zhou, Mauro Conti, Chuadhry Mujeeb Ahmed, Man Ho Au, Lejla Batina, Zhou Li, Jingqiang Lin, Eleonora Losiouk, Bo Luo, Suryadipta Majumdar, Weizhi Meng, Martín Ochoa, Stjepan Picek, Georgios Portokalidis, Cong Wang, and Kehuan Zhang, editors, *Applied Cryptography and Network Security Workshops*, pages 198–216, Cham, 2020. Springer International Publishing.

[WPP20] Lichao Wu, Guilherme Perin, and Stjepan Picek. I choose you: Automated hyperparameter tuning for deep learning-based side-channel analysis. Cryptology ePrint Archive, Report 2020/1293, 2020. https://eprint.iacr.org/2020/1293.

[WPP21] Lichao Wu, Guilherme Perin, and Stjepan Picek. The best of two worlds: Deep learning-assisted template attack. Cryptology ePrint Archive, Report 2021/959, 2021. https://eprint.iacr.org/2021/959.

[WPP22] Lichao Wu, Guilherme Perin, and Stjepan Picek. The best of two worlds: Deep learning-assisted template attack. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(3):413–437, Jun. 2022.

[YAGF21] Rabin Yu Acharya, Fatemeh Ganji, and Domenic Forte. Infoneat: Information theory-based neuroevolution of augmenting topologies for side-channel analysis, 2021.

[Yu20] Ronald Yu. A tutorial on vaes: From bayes' rule to lossless compression, 2020.

[ZBD+20] Gabriel Zaid, Lilian Bossuet, François Dassance, Amaury Habrard, and Alexandre Venelli. Ranking loss: Maximizing the success rate in deep learning side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(1):25–55, Dec. 2020.

[ZBHV19] Gabriel Zaid, Lilian Bossuet, Amaury Habrard, and Alexandre Venelli. Methodology for efficient cnn architectures in profiling attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(1):1–36, Nov. 2019.

[ZS19] Yuanyuan Zhou and François-Xavier Standaert. Deep learning mitigates but does not annihilate the need of aligned traces and a generalized ResNet model for side-channel attacks. *Journal of Cryptographic Engineering*, 10(1):85–95, April 2019.

[ZXF+19] Libang Zhang, Xinpeng Xing, Junfeng Fan, Zongyue Wang, and Suying Wang. Multi-label deep learning based side channel attack. In *2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, pages 1–6, 2019.

[ZZN+20] Jiajia Zhang, Mengce Zheng, Jiehui Nan, Honggang Hu, and Nenghai Yu. A novel evaluation metric for deep learning-based side channel analysis and its extended application to imbalanced data. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(3):73–96, Jun. 2020.

A Visualization of distributions

Evaluators may want to assess the impact of the cVAE-SA model on the leakage distribution. In this appendix, we use some visualization tools in order to illustrate the impact of the encoder and the decoder on the leakage distribution. In particular, it helps to illustrate the theoretical results stated in Sec.3. In that purpose, the following scenario is considered:

a set of 2-dimensional leakage traces is simulated such that the leakage model does not induce interactions between bits. In detail, the i^{th} time sample of the simulated trace $\mathbf{T}$ is defined as follows:

$$\mathbf{T}[i] = \begin{cases} 1 \cdot Y[3] + 1 \cdot Y[6] + \mathbf{Z}[i] & \text{if} \quad i = 1, \\ \mathbf{Z}[i] & \text{otherwise,} \end{cases} \tag{13}$$

where $Y[b] = Sbox[X \oplus k^*][b]$ denotes the b^{th} bit of the output of the Sbox and $\mathbf{Z}[i]$ is a Gaussian noise following $\mathcal{N}(0, \sigma^2)$ such that $\sigma^2 = 1$. Following this scenario, three leakage distributions can be observed depending on the value of the bits $Y[3]$ and $Y[6]$:

- If $Y[3] = Y[6] = 0$, the leakage distribution performs similarly to $\mathbf{Z}$. The related label is denoted by 0.
- If $Y[3] = 1$ or $Y[6] = 1$, one bit of Y influenced the leakage distribution. The related label is denoted by 1.
- If $Y[3] = Y[6] = 1$, both bits influenced the leakage distribution. The related label is denoted by 2.

Based on those three leakage distributions, we want to illustrate the ability of the cVAE-SA to capture the mutual dependency between the leakage traces and the targeted variable Y. Before the application of the cVAE-SA (see the left plot in Fig.6), it can be observed that depending on the informative value, an evaluator can retrieve some information regarding the label processed. Consequently, as the cVAE-SA constructs synthetic traces that should be similar to the input, the output leakage distribution should be similar to the one before the application of the encoder, if the cVAE-SA is well trained. Based on the leakage trace $\mathbf{T}$, the cVAE-SA isolates the deterministic part, *i.e.* ψ, from the noise $\mathbf{Z}$ (see Sec.3.2). In particular, if the encoder is well configured during the training process, the $\hat{\psi}_\Theta$ layer approximates the real unknown ψ function. One solution to empirically verify such approximation is to visualize the weights that composed the $\hat{\psi}_\Theta$ layer (see Sec.4.2). Therefore, the latent space representation should be only characterized by the noise part $\mathbf{Z}$ if the cVAE-SA is well trained. This observation is validated by the middle plot in Fig.6. As no distinctions can be made regarding the label value, it can be assumed that the latent space behaves similarly whatever the underlying secret value. This empirical result confirms the theoretical ones provided in Sec.3.2 and Sec.3.3 (see the analysis related to the KL-divergence loss). Finally, an evaluator constructs a new set of synthetic traces based on the latent space and the $\hat{\psi}_\phi$ layer. This construction is performed by the decoder and the resulting distributions are illustrated in the right plot of Fig.6. Through this Figure, it can be observed that the cVAE-SA discriminates each label following the mean of the related conditional distribution. This confirms the ability of the model to identify the mutual dependency between the initial leakage traces and the targeted variable Y. If the cVAE-SA is perfectly trained, the output distributions should be similar to those introduced in input. In Fig.6, this statement can be confirmed. First, through the visualization of the latent space, the evaluator can assess that the noise part of $\mathbf{T}$ is well approximated. Indeed, as the informative and non-informative samples look similar, it can be assumed that the latent representation of the cVAE-SA is successfully trained. Then, through the visualization of the synthetic leakage traces, the evaluator observes that the informative sample introduces relevant information regarding the targeted variable (*i.e.* $\hat{\psi}_\phi(Y)$). In particular, thanks to Eq.4, it can be assumed that the synthetic traces follow the Gaussian distribution $\mathcal{N}_D(\boldsymbol{\mu}_{\tilde{\mathbf{T}}}, \Sigma_{\tilde{\mathbf{T}}})$ such that, $\boldsymbol{\mu}_{\tilde{\mathbf{T}}} = \hat{\psi}_\phi$ and $\Sigma_{\tilde{\mathbf{T}}} = \Sigma_{\mathbf{V},\Theta}$. Therefore, when the evaluator conducts a key recovery phase, he exploits the first-order moment in order to recover information about the secret key. This confirms the statement provided in Sec.3.4. However, if $\mathcal{F}_9$ is configured to approximate the leakage model when Eq.13 is considered, the model quality can be badly impacted by a poor

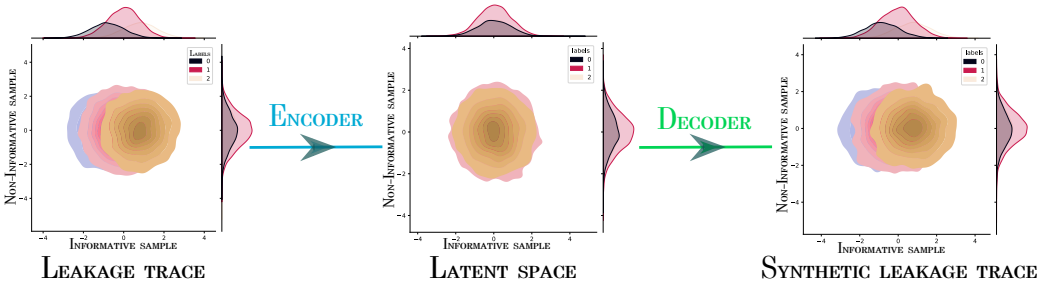


Figure 6: Evolution of the distributions over the cVAE-SA model when a successful attack is performed ($\mathcal{F}_9$).

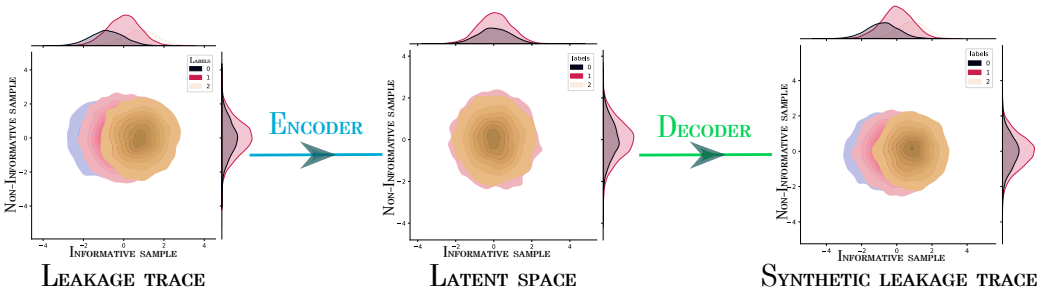


Figure 7: Evolution of the distributions over the cVAE-SA model when a successful attack is performed ($\mathcal{F}_2$).

estimation of the high order degree of bits' interaction [MOW17]. A cVAE-SA inducing a better quality model should consider $\mathcal{F}_2$ which alleviates the leakage model complexity by setting aside bits' interaction. This statement is illustrated in Fig.7. No huge differences can be highlights between Fig.6 and Fig.7. However, using such visualization tool can give a first insight to the evaluator in order to assess the impact of a poor basis choice.

Before performing the key recovery phase introduced in Sec.3.4, an evaluator may want to assess the suitability of the cVAE-SA training process. In Fig.8, we visualize all the distributions related to the latent space and the synthetic traces. While the latent space suggests a good approximation of the noise part, the distributions related to the synthetic leakage traces illustrate that the informative sample does not perform any discrimination regarding the targeted variable Y. This observation is consistent with Sec.3.4 which defines the key recovery phase as successful if the first-order moment of the synthetic leakage traces provides depends on Y. Consequently, plotting the distributions of each part of the cVAE-SA (*i.e.* input data, latent space, output data) can be beneficial to have a better understanding of the model quality as well as the impact of the leakage model against the noise.

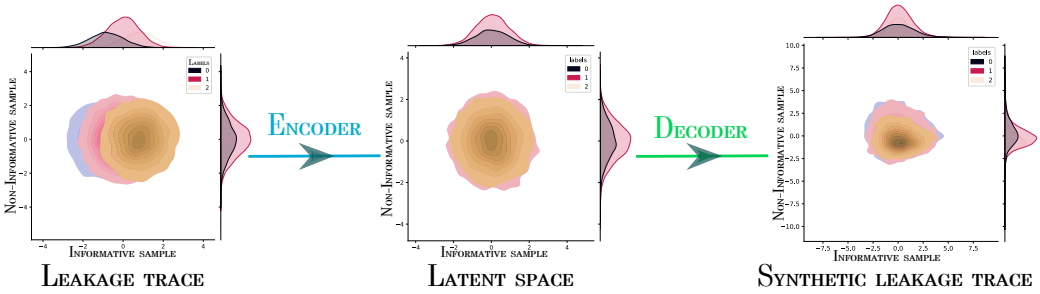


Figure 8: Evolution of the distributions over the cVAE-SA model when an unsuccessful attack is performed ($\mathcal{F}_9$).

B Impact of the Noise on Leakage Model Estimation

To verify the benefits of the cVAE-SA, we simulate $10,000$ D-dimensional traces from a 8-bit sensitive variable Y and assess the ability of this new architecture to extract leakage models. As mentioned in Sec.4, the weight visualization is a suitable tool to identify the leakage model extracted by the cVAE-SA. Therefore, an evaluator is able to retrieve the impact of each bit independently as well as all the bits' interaction. This tool has been confirmed on different use-cases in Sec.5.2. In this appendix, some simulated traces are built following three scenarios with different amounts of noise:

- **Scenario 1** – We assume that each leakage trace is configured by 3 time samples such that the leakage model induces the maximum amount of interactions between bits (*i.e.* $\mathcal{F}_9$). In this scenario, all bits influencing the leakage model have the same weights. Hence, the i^{th} time sample of the simulated trace $\mathbf{T}$ is defined as follows:

$$\mathbf{T}[i] = \begin{cases} 1 \cdot Y[1] + 1 \cdot Y[3] + 1 \cdot Y[6] \\ + 1 \cdot \oplus_{b=0}^{1} Y[b] + 1 \cdot \oplus_{b=0}^{2} Y[b] + 1 \cdot \oplus_{b=0}^{3} Y[b] \\ + 1 \cdot \oplus_{b=0}^{4} Y[b] + 1 \cdot \oplus_{b=0}^{5} Y[b] + 1 \cdot \oplus_{b=0}^{6} Y[b] \\ + 1 \cdot \oplus_{b=0}^{7} Y[b] + \mathbf{Z}[i] & \text{if} \quad i = 1, \\ \mathbf{Z}[i] & \text{otherwise,} \end{cases}$$

  where $\oplus_{b=0}^{n} Y[b] = Y[0] \oplus \ldots \oplus Y[n]$, $Y[b] = Sbox[X \oplus k^*][b]$ denotes the b^{th} bit of the output of the Sbox, and $\mathbf{Z}[i]$ is a Gaussian noise following $\mathcal{N}(0, \sigma^2)$ such that $\sigma^2 \in \{0.1, 1, 10\}$.

- **Scenario 2** – We assume that each leakage trace is configured by 4 time samples. The leakage model does not induce interactions between bits but differs from the location of the points of interest. Hence, the i^{th} time sample of the simulated trace $\mathbf{T}$ is defined as follows:

$$\mathbf{T}[i] = \begin{cases} 1 \cdot Y[3] + 1 \cdot Y[6] + \mathbf{Z}[i] & \text{if} \quad i = 1, \\ 1 \cdot Y[1] + 1 \cdot Y[7] + \mathbf{Z}[i] & \text{if} \quad i = 2, \\ \mathbf{Z}[i] & \text{otherwise,} \end{cases}$$

  where $Y[b] = Sbox[X \oplus k^*][b]$ and $\mathbf{Z}[i]$ is a Gaussian noise following $\mathcal{N}(0, \sigma^2)$ such that $\sigma^2 \in \{0.1, 1, 10\}$.

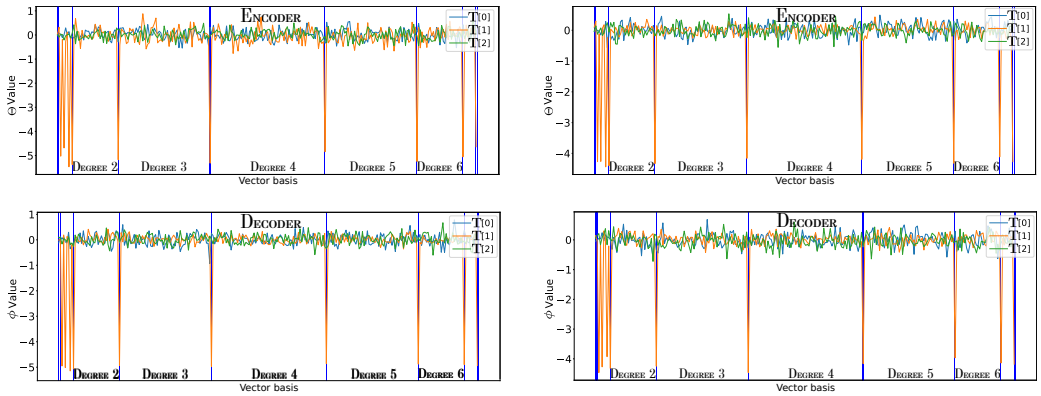
- **Scenario 3** – We assume that each leakage trace is configured by 3 time samples. The leakage model does not induce interactions between bits such that all bits

influencing the leakage model have different weights. Hence, the i^{th} time sample of the simulated trace $\mathbf{t}$ is defined as follows:

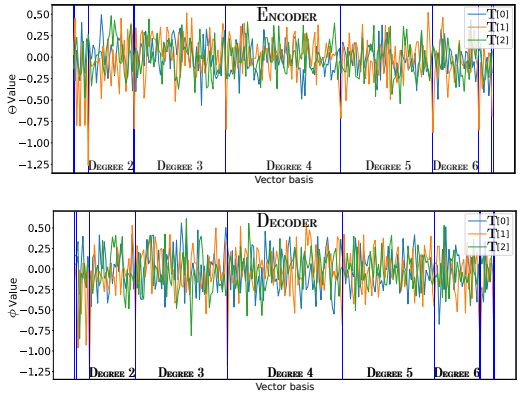
$$\mathbf{T}[i] = \begin{cases} 1 \cdot Y[3] + 0.5 \cdot Y[6] + \mathbf{Z}[i] & \text{if} \quad i = 1, \\ \mathbf{Z}[i] & \text{otherwise,} \end{cases}$$

where $Y[b] = Sbox[X \oplus k^*][b]$ and $\mathbf{Z}[i]$ is a Gaussian noise following $\mathcal{N}(0, \sigma^2)$ such that $\sigma^2 \in \{0.1, 1, 10\}$.

Based on the results obtained in Fig.9, Fig.10 and Fig.11, it can be observed that the cVAE-SA retrieves all the leakage models when moderate SNR level is considered (*i.e.* SNR $\geq 10^{-1}$). Hence, the cVAE-SA can be used to evaluate the security flaws when large bits' interactions are observed, when the deterministic part differs between PoIs and when a non-uniform weight distribution occurs between bits. As a consequence, large use-cases can be considered when the cVAE-SA is applied. However, if low SNR level is defined, the extraction of the leakage model becomes more difficult. Indeed, while an evaluator retrieves some information on the leakage model related to **Scenario 1** (e.g. the highest peaks of degree 1 indicate that the bits $Y[1]$, $Y[3]$ and $Y[6]$ influence the leakage model, see Fig.9c), this interpretation can be more difficult when the SNR result is lower (see Fig.10c and Fig.11c). This result is in accordance with the theoretical ones introduced in Sec.3.3 which suggest that increasing the noise in the traces makes the deterministic part extraction more difficult. One solution to mitigate this lack of leakage characterization consists in acquiring a larger amount of traces [DSVC14, MCHS22] in order to find the trainable weights Θ and ϕ which fit the most with the true unknown leakage model.

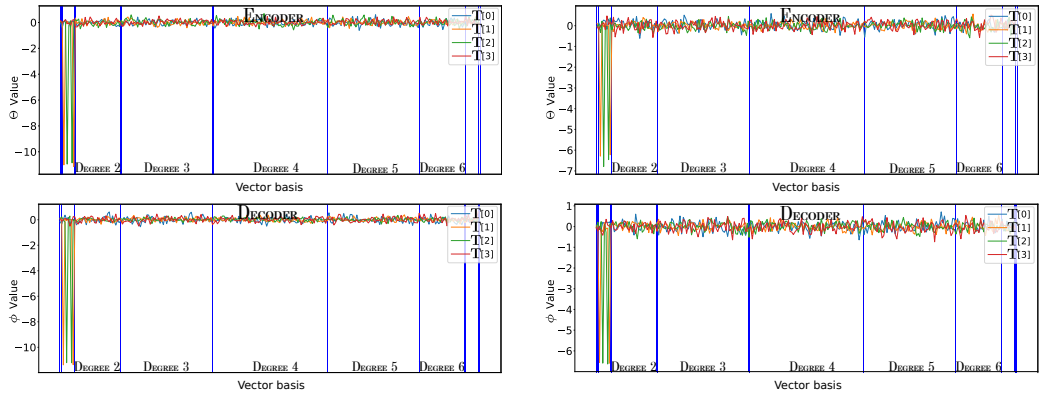


(a) $\sigma^2 = 0.1$ / SNR $= 258.51$. **(b)** $\sigma^2 = 1$ / SNR $= 2.578$.



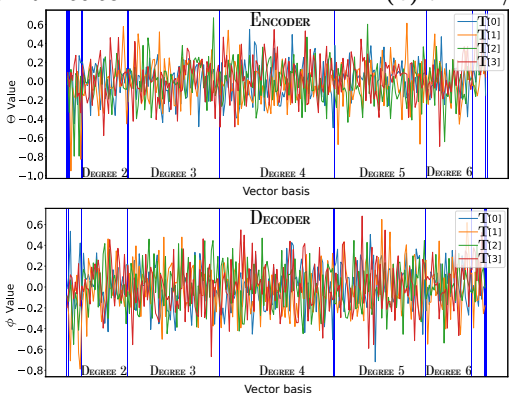
(c) $\sigma^2 = 10$ / SNR $= 0.0577$.

Figure 9: Weight visualization of the $\hat{\psi}_{\Theta}$ layer (encoder) and the $\hat{\psi}_{\phi}$ layer (decoder) for **Scenario 1**.



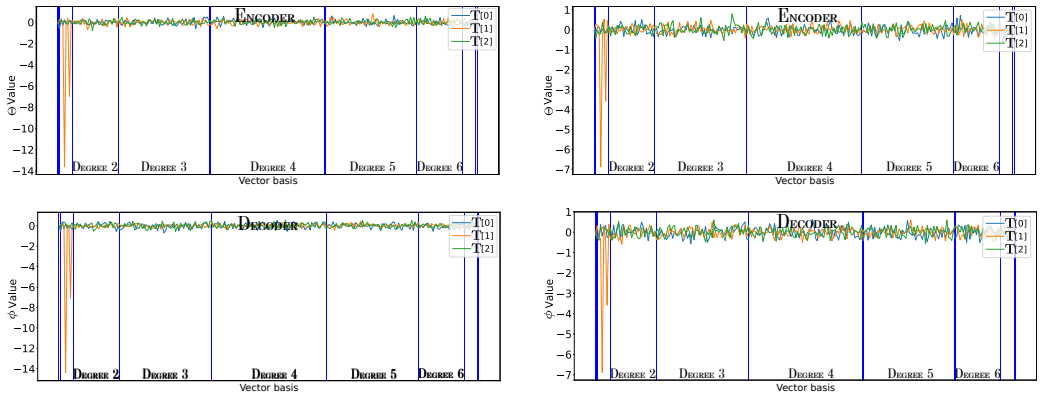
(a) $\sigma^2 = 0.1$ / SNR $= 50.58$.

(b) $\sigma^2 = 1$ / SNR $= 0.509$.



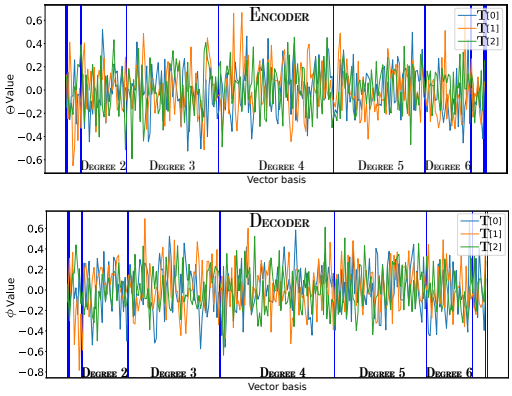
(c) $\sigma^2 = 10$ / SNR $= 0.0319$.

Figure 10: Weight visualization of the $\hat{\psi}_\Theta$ layer (encoder) and the $\hat{\psi}_\phi$ layer (decoder) for **Scenario 2**.



(a) $\sigma^2 = 0.1$ / SNR $= 32.38$.

(b) $\sigma^2 = 1$ / SNR $= 0.346$.



(c) $\sigma^2 = 10$ / SNR $= 0.0262$.

Figure 11: Weight visualization of the $\hat{\psi}_\Theta$ layer (encoder) and the $\hat{\psi}_\phi$ layer (decoder) for **Scenario 3**.

C Explainability on ASCAD-v1

Through this section, we assess the benefits of the cVAE-SA to better explain and interpret the decision-making of this new statistical model. As mentioned in Sec.2.1, the interpretation refers to the ability of the evaluator to clearly identify each operation induced in the model in order to exploit the sensitive information. This includes the construction of a statistical model, namely cVAE-SA, where the extraction of the leakage model related to each PoI is fully explainable in order to identify security flaws. Through this section, a focus is proposed on the ASCAD-v1-R dataset which is introduced in Sec.5.1. This choice has been motivated because it can be considered as the most challenging targeted dataset (*i.e.* protected implementation with first-order masking). Due to the implemented countermeasure, two scenarios can be considered. The first one suggests that the evaluator wants to target independently the mask r_3 and the masked values with r_3 (see [BPS+20] for deeper details on the implementation). This approach is beneficial to assess the robustness of the targeted implementation and identify the security flaws without any preprocessing phase. This scenario will be denoted as the *naive approach*. The second solution consists in combining the time samples related to the mask r_3 and the masked values with r_3 in order to target the unmasked values (see Sec.4.4). This latter solution is beneficial to identify the dependence generated by the combining function between the unmasked variable and a set of traces. This scenario will be denoted as the *combining approach*. To address the explainability and interpretability issue, this section will be decomposed into three parts: the construction of the cVAE-SA based on the theoretical results described in Sec.3.2, the detection and the extraction of the leakage models once the cVAE-SA is trained, and, the ability of the cVAE-SA to correctly characterize the first-order moment of the traces related to the ASCAD-v1-R dataset.

Model construction. Introduced in Sec.3.2, the cVAE-SA structure is adapted to capture the dependencies between a leakage trace $\mathbf{T} \in \mathbb{R}^D$ and a label $Y \in \mathbb{F}_2^n$. Therefore, the cVAE-SA can be used to capture how the mask r_3 and the masked values with r_3 influence the physical trace $\mathbf{T}$ when the *naive approach* is considered. The evaluator can construct two distinct cVAE-SA models (*i.e.* one for r_3 and one for the masked values) based on the recommendations defined in Sec.3.2. To construct the cVAE-SA architecture, the same configuration as in Sec.5.2 is considered. Indeed, we select the 8 most relevant samples related to the mask r_3 and the masked values with r_3 and then, construct each cVAE-SA model. The only difference between those models rely on the input provided to the related cVAE-SA model, namely the trace $\mathbf{T}$ and the orthonormal monomial basis used. As mentioned in Sec.3.4, the network complexity can be defined following the number of samples s included in the traces, the degree of interaction d between the time samples and the dimension n of the targeted variable such that it equals $(2s \cdot ((s+1) + 1 + \sum_{i=0}^{d} \binom{n}{i}))$ if $\Sigma_{\mathbf{V},\Theta}$ is reduced to $\sigma^2_{\mathbf{V},\Theta}$. Through this section, we define $s = 8$, $d = 8$ and $n = 8$. Thus, the complexity of the cVAE-SA model targeting the mask r_3, or the masked values, equals $4,256$. This is confirmed in Tab.5. Another solution consists in constructing a single cVAE-SA model by considering the concatenation of the orthonormal monomial basis of r_3 and the one related to the masked value with r_3 as label Y. This configuration can be defined as a multi-task learning strategy and has already been studied in Sec.4.2. Therefore, this section will be only focused on the construction of two distinct cVAE-SA models.

For the *combining approach*, the time samples related to the mask r_3 have to be combined with the one related to masked values in order to create dependency between the trace $\mathbf{T}$ and the targeted unmasked variable. Therefore, a preprocessing step is needed where the evaluator has to choose a combining function among the solutions introduced in the state-of-the-art [CJRR99, Mes00, PRB09]. Once this preprocessing is conducted, the evaluator constructs the cVAE-SA model (see Sec.3.2) such that the inputs of the encoder

Table 5: Information on the cVAE-SA model considered in this setting (**batch size** = 256).

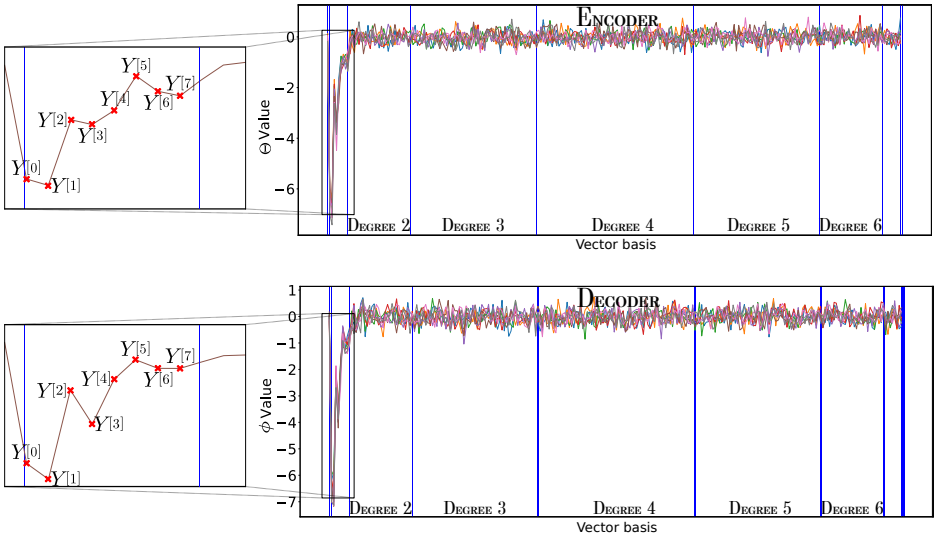
Approach	cVAE-SA model	Learning rate	Nb epochs	Network complexity	Training time
Naive	cVAE-SA$_{\text{msk}}$	10^{-1}	10	$4,256$	$37s$
	cVAE-SA$_{\text{msked}}$	10^{-3}	20	$4,256$	$66s$
Combining	cVAE-SA$_{\text{unmsked}}$	10^{-3}	100	$41,216$	$534s$

are defined by the combined traces and the orthonormal monomial basis related to the targeted unmasked variable. For this approach, the number of time samples to target equals 64. Therefore, by applying the same proposition as previously, the evaluator can easily configure the cVAE-SA with a complexity of $41,216$ trainable parameters. All the information related to each cVAE-SA model is provided in Tab.5.

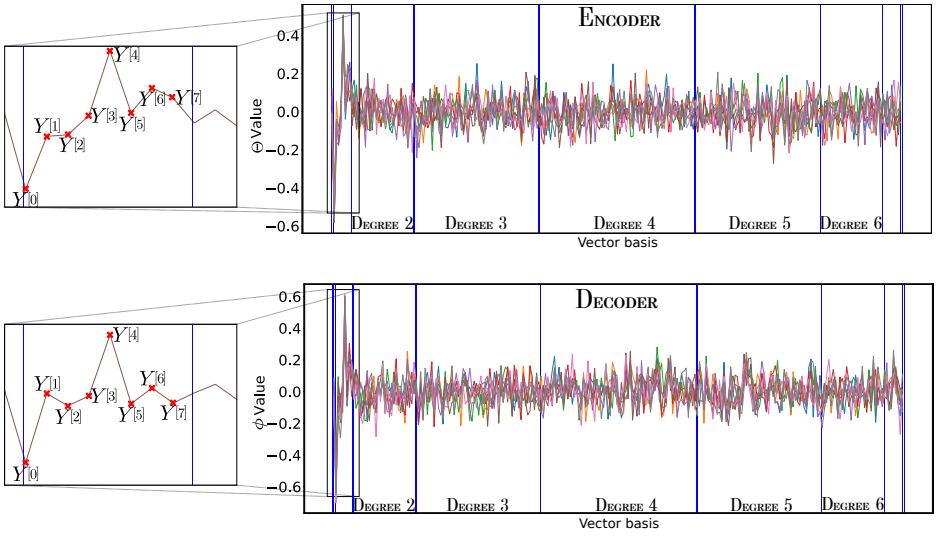
Leakage model extraction. Once all the cVAE-SA models are trained, the evaluators can take advantage of the explainability property of this new contribution to get a better insight on the exploited leakage models. As mentioned in Sec.4.2, the evaluator can visualize the trainable parameters Θ (resp. ϕ) that composed the $\hat{\psi}_{\Theta}$ (resp. $\hat{\psi}_{\phi}$) layer induced in the encoder (resp. the decoder) in order to detect which part of the targeted variable leaks. In other words, this analysis is helpful to identify the bits, and the interactions, that influence the physical consumption of the targeted implementation. This is highly beneficial to explain the information that is extracted by the cVAE-SA. Through Fig.12, it can be observed that depending on the targeted variable (*i.e.* the mask r_3, the masked values or, the unmasked values), the leakage model as well as the coefficient values differ. Indeed, the highest absolute value is observed when the cVAE-SA targets the mask r_3 while the lowest absolute value is denoted for the masked variable. Therefore, the approximation of the first-order moment varies depending on the targeted variable.

When the mask r_3 is considered, the visualization indices in Fig.12a is beneficial to recover the leakage model extracted by the cVAE-SA. A first observation can be made to denote that all the time samples have a similar leakage model. Even if the coefficient values related to each bit of r_3 differ, the same bits leak for all PoIs. Through Fig.12a, the evaluator can retrieve which bit influences the physical trace by highlighting the ones with a discriminative coefficient Θ and ϕ. For the mask r_3, the following bits $\{0, 1, 2, 3, 4, 6, 7\}$ have a coefficient that differs from the non-informative interaction, *i.e.* when Θ and ϕ are greater than 1. This analysis allows the evaluator to define an ascending leakage order to highlight the bit which leaks the most (in absolute value). In this configuration, the following order is observed: $r_3[6] < r_3[7] < r_3[4] < r_3[2] < r_3[3] < r_3[0] < r_3[1]$ such that $r_3[i]$ denotes the $(i+1)^{\text{th}}$ bit of r_3. Therefore, the bit that leaks the most is $r_3[1]$. The same process can be conducted for the masked variable (see Fig.12b) and the unmasked variable (see Fig.12c). When the masked values with r_3 is targeted, the leakage model approximated by the encoder and the decoder identify the 1^{st} bit and the 5^{th} bit as the only source of information that can be extracted from cVAE-SA$_{\text{msked}}$. All the PoIs share the same leaking bits. Finally, once the optimal recombination is conducted, the leakage model that is retrieved by the cVAE-SA$_{\text{unmsked}}$ network is influenced by the following bits $\{0, 1, 2, 3, 4\}$ such that the following leakage order is observed (in absolute value): $Sbox[X \oplus k^*][3] < Sbox[X \oplus k^*][2] < Sbox[X \oplus k^*][1] < Sbox[X \oplus k^*][4] < Sbox[X \oplus k^*][0]$.

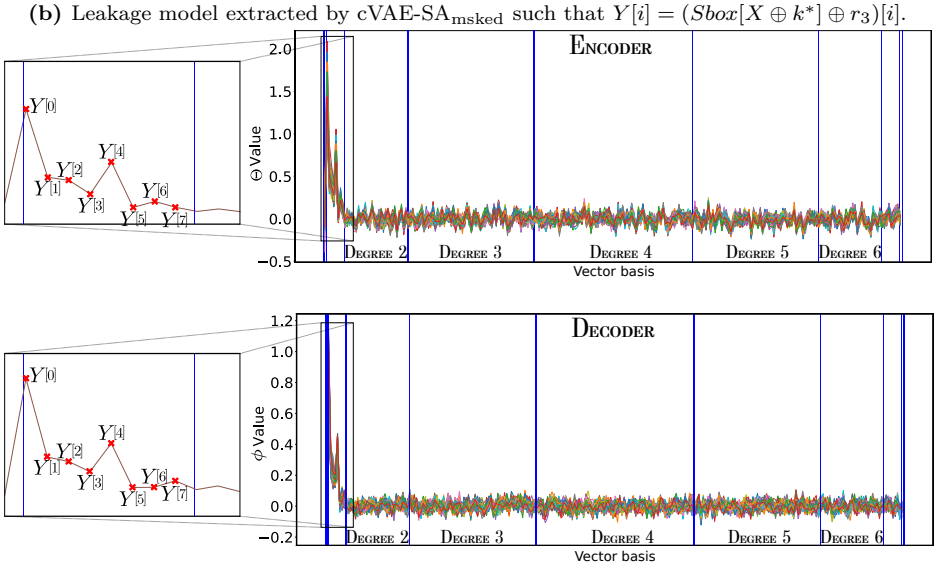
Those results are beneficial for the evaluator to explain and interpret the decision-making process of each cVAE-SA model. If this basis characterizes the switching activity of the circuit, this analysis highlights specific exploitable security flaws in ASCAD-v1-R. All those observations cannot be observed when classical (discriminative) DLSCA models are considered. Those interpretable results can only be provided because the cVAE-SA is designed from the fully explainable stochastic attack [SLP05].



(a) Leakage model extracted by cVAE-SA$_{\text{msk}}$ such that $Y[i] = r_3[i]$.

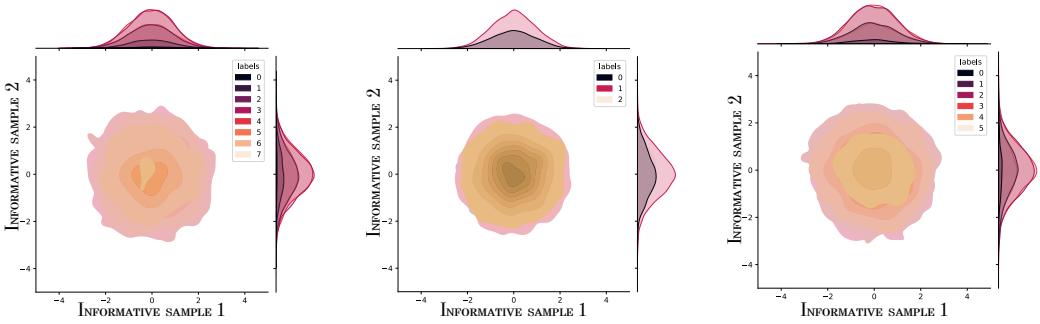


(b) Leakage model extracted by cVAE-SA$_{\text{msked}}$ such that $Y[i] = (Sbox[X \oplus k^*] \oplus r_3)[i]$.



(c) Leakage model extracted by cVAE-SA$_{\text{unmsked}}$ such that $Y[i] = Sbox[X \oplus k^*][i]$.

Figure 12: Leakage model extracted depending on the targeted variable.



(a) cVAE-SA$_{\text{msk}}$ **(b)** cVAE-SA$_{\text{msked}}$ **(c)** cVAE-SA$_{\text{unmsked}}$

Figure 13: Distribution of the latent space depending on the targeted sensitive variable.

Model quality. Once the leakage models extracted by the cVAE-SA are fully interpreted, the evaluator may wonder if additional information could be extracted. To conduct such verification, the evaluator can plot the evolution of the distribution over the cVAE-SA (see Sec.A) to assess the estimation of the first-order moment that is required to retrieve the secret key (see Sec.3.4). Therefore, he can visualize if the input and the output distributions of the cVAE-SA are similar. If this result is positive, the estimation of the first-order moment induced in the cVAE-SA can be considered as effective. Otherwise, some refinements can be provided on the hyperparameter values (e.g. learning rate, batch-size, number of epochs, ...). While App.A proposes to observe the distribution related to the leakage traces, the latent space and the synthetic leakage traces, an evaluator only requires the distribution of the latent space and the estimation of the leakage model of the encoder and the decoder to assess if the cVAE-SA correctly approximates the first-order moment induced in the leakage traces. Indeed, following Sec.3.2 and Sec.3.3, we can note that the latent space should be representative of the noise distribution such that it is forced to follow $\mathcal{N}_D(0, \mathbf{I}_D)$ by the KL-divergence loss. Therefore, if the encoder of the cVAE-SA model is correctly trained, the latent representation does not depend on the deterministic part of the leakage trace. Similar latent representations should be obtained whatever the targeted variable. This observation is confirmed in Fig.13 where the visualization of the distribution is proposed on two time samples in order to ease the readability of the experimental results. Therefore, the encoder is effectively trained and the related trainable parameters, namely Θ, correctly retrieve the targeted unknown leakage model. Then, *via* the visualization of the trainable parameters ϕ in Fig.12, it can be mentioned that the decoder approximates the targeted unknown leakage model because it is similar to the one extracted by the encoder. The extraction of the leakage model is consequently effective for both samples analyzed in Fig.13. The analysis of the model quality has also been conducted on other time samples in order to verify the extraction each leakage model. The obtained results help us to verify that the cVAE-SA extracts effectively the first-order moment that is needed to retrieve the secret key. As mentioned in Sec.3.4, if the latent space distribution follows $\mathcal{N}_D(0, \mathbf{I}_D)$, the first-order moment is only defined by the extracted leakage model $\hat{\psi}_\phi$. The justification provided in this section helps the evaluator to validate the suitability of the training process as well as justify the ability of an adversary to extract the secret information (see Sec.5).

Based on the ASCAD-v1-R dataset, we identify the benefits of using the cVAE-SA model in order to enhance the explainability and the interpretability of the neural network in the DLSCA field. Through this study, we demonstrate that the construction of cVAE-SA models can be easily conducted whatever the targeted variable. This is highly beneficial from an evaluation perspective because the time for the hyperparameters search becomes negligible. Then, because all the operations induced in the cVAE-SA are known (see

Sec.3.2), the evaluator can take advantage of this benefit in order to identify which leakage model is extracted in each PoI of the traces. Based on each leakage model, he can identify the security flaws induced in the circuit if the chosen basis characterizes the switching activity. Finally, because the trainable parameters of the cVAE-SA model are interpretable, the evaluator can assess the suitability of the training process by identifying if the first-order moment of the synthetic traces is representative of the first-order moment of the true leakage traces. All those observations cannot be conducted with classical DLSCA models due to the black-box property of the discriminative approach.