Efficiency through Diversity in Ensemble Models applied to Side-Channel Attacks

– A Case Study on Public-Key Algorithms –

Gabriel Zaid[1,2], Lilian Bossuet[1], Amaury Habrard[1] and Alexandre Venelli[3*]

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

Abstract. Deep Learning based Side-Channel Attacks (DL-SCA) are considered as fundamental threats against secure cryptographic implementations. Side-channel attacks aim to recover a secret key using the least number of leakage traces. In DL-SCA, this often translates in having a model with the highest possible accuracy. Increasing an attack's accuracy is particularly important when an attacker targets public-key cryptographic implementations where the recovery of each secret key bits is directly related to the model's accuracy. Commonly used in the deep learning field, ensemble models are a well suited method that combine the predictions of multiple models to increase the ensemble accuracy by reducing the correlation between their errors. Linked to this correlation, the *diversity* is considered as an indicator of the ensemble model performance. In this paper, we propose a new loss, namely *Ensembling Loss* (EL), that generates an ensemble model which increases the diversity between the members. Based on the mutual information between the ensemble model and its related label, we theoretically demonstrate how the ensemble members interact during the training process. We also study how an attack's accuracy gain translates to a drastic reduction of the remaining time complexity of a side-channel attacks through multiple scenarios on public-key implementations. Finally, we experimentally evaluate the benefits of our new learning metric on RSA and ECC secure implementations. The *Ensembling Loss* increases by up to 6.8% the performance of the ensemble model while the remaining brute-force is reduced by up to 2^{22} operations depending on the attack scenario.

Keywords: Side-Channel Attacks · Deep Learning · Ensemble Learning · Diversity · Mutual Information · Public-Key Algorithms

1 Introduction

Side-channel analysis (SCA) is a class of cryptographic attack in which an attacker tries to exploit the vulnerabilities of a system by analyzing its physical properties, including power consumption [KJJ99] or electromagnetic emissions [AARR03], to reveal secret information. One of the most powerful types of SCA attacks are *profiled attacks*. In this scenario, the attackers have access to a test device whose target intermediate values are known. Very similar to profiled attacks, the application of deep learning algorithms was inevitably explored in the side-channel context [MPP16, CDP17, PHJ+19, MDP19, ZBHV19].

*Work done when the author was at Thales ITSEF.

While most of the works published on Deep Learning Side-Channel Analysis (DL-SCA) target symmetric cryptographic implementations, some of them investigate the effectiveness of neural networks for defeating secure RSA [CCC+19] and elliptic curves [WPB19, ZS19, PCBP20]. Due to a careful combination of countermeasures (e.g. message blinding, modulus randomization, exponent/scalar blinding, point blinding), the attacker must be able to recover more than 90% of the secret bits from a single trace [Cor99, Gir06]. Attacking public key implementations requires to recover each of secret bits by repeating the attack. Hence, the accuracy of the attack is crucial in order to lower the remaining operations required to find the entire secret key. This focus on the attack accuracy is particular to the public key case, as for symmetric implementations, the attacker aggregates the output probabilities of the model on multiple traces. Moreover, as public keys are much larger than symmetric keys a small gain in the attack accuracy improves drastically the remaining attack complexity.

In this paper, we consider the two main types of exploitation scenarios for profiled attacks on public key implementations:

- **N traces exploitation** – The attacker has access to N leakage traces in order to recover the secret exponent d. This use case corresponds to ECDH (Elliptic Curve Diffie-Hellman) or RSA signature computations when exponent/scalar blinding countermeasure is applied.
- **1 trace exploitation** – The attacker has access to only 1 leakage trace in order to recover the secret exponent d. This use case corresponds to ECDSA (Elliptic Curve Digital Signature Algorithm) targeting the scalar multiplication with a random nonce.

In machine learning, ensemble methods combine individual predictions from all members of a pool via a consensus method (*i.e.* majority vote, average, ...) [HS90, Kun04, Zho12]. These approaches are useful when the members of the committee learn and predict uncorrelated errors. Hence, a simple consensus method can efficiently reduce the global error of the system. However, in practice, the errors induced by the committee members are correlated and the overall ensemble error reduction is hard. One solution to reduce this correlation is to conduct a diversity investigation on the members in order to reduce the global error and increase to some extent, the ensemble performance [Die00b]. Following Liu *el al.* [LWC+19], three ways exist to create diversity in ensemble learning. *Type I diversity* corresponds to the variety of the committee members structure (e.g. network architecture, optimizer hyperparameters, ...). This classical diversity was studied by Perin *et al.* in the side-channel context [PCP20]. The authors provide experimental results on symmetric algorithm implementations and show that combining predictions of multiple neural networks is useful to gain in performance. *Type II diversity* carefully chooses the network to promote error independence between the classifiers in the ensemble. Finally, *Type III diversity* captures the posterior probability distribution during the training process by maximizing the diversity between the learners to encourage a convergence towards different hypotheses. Combining these types of diversity can be helpful to generate a powerful ensemble model.

Contributions. Our paper extends the preliminary results of ensemble methods in the side-channel context [PCP20] by providing theoretical observations and new propositions to increase the impact of ensembling in SCA. More precisely, our work mainly focuses on the type III diversity which has not been studied in the SCA literature to the best of our knowledge.

First, we propose a new loss, namely *Ensembling Loss* (EL), that maximizes the mutual information between the ensemble model and the sensitive information. This contribution, derived from [Bro09] and [ZBD+20], tends to maximize the type III diversity between the

committee members during the training process in order to ensure an ensemble of diverse members. Hence, the *Ensembling Loss* can be used in addition to types I and II diversity to increase the performance of an ensemble model.

Our theoretical observations are validated on two public datasets: a secure RSA implementation with exponent blinding [CCC+19] and a protected ECSM (Elliptic Curve Scalar Multiplication) implementation [NCOS17, Chm20] with scalar randomization. Each of these datasets correspond to a type of exploitation scenario detailed above.

While the goal of this paper is not to compare the benefits of the *Ensembling Loss* with the other diversity types, we combine the proposed loss (*i.e.* type III diversity) with types I and II diversity to evaluate its impact on the ensemble model's diversity.

Finally, ensemble methods are well-known to increase the performance of a model regardless of the training process. Hence, using these techniques could have a huge impact on the training time. To support the relevance of the *Ensembling Loss*, we evaluate the impact of the accuracy gain on the remaining complexity of a side-channel attack and the resulted training time. We study different remaining complexity methods for public keys: the *naive complexity*, the 2^n*-complexity* and the complexity of the Alternate Attack [SW14, SW17]. These wide-range scenarios illustrate the negligible impact of the increase in training time compared to the major attack complexity improvement. To evaluate the practicability of our result, we consider the European SOG-IS scheme as a reference to support the benefits of the Ensembling Loss.

The loss proposed in this paper can also be obviously applied on symmetric cryptographic implementations and more generally, on all types of machine learning problems where a gain in accuracy is crucial.

All these experiments can be reproduced through the following GitHub repository: https://github.com/gabzai/Ensembling-Loss-SCA.

Paper Organization. The paper is organized as follows. Section 2 recalls the learning metrics introduced in the side-channel context. It also explains the relationship between ensemble models and diversity. Section 3 proposes a new loss, called Ensembling Loss (EL), which generates an ensemble model converging towards the mutual information between a pool of classifiers and a set of labels. Section 4 presents the dataset used to validate the theoretical observation and the side-channel complexity measures. Section 5 illustrates the benefits of the ensembling loss through experimental results. Finally, Section 6 extends the ensembling loss on a binary classification problem and discusses on traditional methods used in ensemble learning (combination methods, ensemble methods, impact of the number of committee members, ...).

2 Preliminaries

2.1 Notation and terminology

Let calligraphic letters $\mathcal{X}$ denote sets, the corresponding capital letters X (resp. bold capital letters) denote random variables (resp. random vectors $\mathbf{T}$) and the lowercase x (resp. $\mathbf{t}$) denote their realizations. The i-th entry of a vector $\mathbf{t}$ is defined as $\mathbf{t}[i]$. Side-channel traces will be constructed as a random vector $\mathbf{T} \in \mathbb{R}^{1\times D}$ where D defines the dimension of each trace. The targeted sensitive variable is $Z = f(P, K)$ where f denotes a cryptographic primitive, P ($\in \mathcal{P}$) denotes a public variable (e.g. plaintext or ciphertext) and K ($\in \mathcal{K}$) denotes a part of the key (e.g. byte) that an adversary tries to retrieve. Z takes values in $\mathcal{Z} = \{s_1, ..., s_{|\mathcal{Z}|}\}$ such that s_j denotes a score associated with the j^{th} sensitive variable. Let us denotes k^* the secret key used by the cryptographic algorithm. We define the following information theory quantities needed in the rest of the paper [CT91] . The entropy of a random vector $\mathbf{X}$, denoted $H(\mathbf{X})$, measures the unpredictability

of a realization $\mathbf{x}$ of $\mathbf{X}$. It is defined by:

$$H(\mathbf{X}) = -\sum_{\mathbf{x}\in\mathcal{X}} \mathsf{Pr}\left[\mathbf{X}=\mathbf{x}\right] \cdot \log_2\left(\mathsf{Pr}\left[\mathbf{X}=\mathbf{x}\right]\right).$$

The conditional entropy of a random variable $\mathbf{X}$ knowing $\mathbf{Y}$ is defined by:

$$\begin{aligned} H(\mathbf{X}|\mathbf{Y}) &= -\sum_{\mathbf{y}\in\mathcal{Y}} \mathsf{Pr}\left[\mathbf{Y}=\mathbf{y}\right] \cdot H\left(\mathbf{X}|\mathbf{Y}=\mathbf{y}\right) \\ &= -\sum_{\mathbf{y}\in\mathcal{Y}} \mathsf{Pr}\left[\mathbf{Y}=\mathbf{y}\right] \cdot \sum_{\mathbf{x}\in\mathcal{X}} \mathsf{Pr}\left[\mathbf{X}=\mathbf{x}|\mathbf{Y}=\mathbf{y}\right] \cdot \log_2\left(\mathsf{Pr}\left[\mathbf{X}=\mathbf{x}|\mathbf{Y}=\mathbf{y}\right]\right). \end{aligned}$$

The *Mutual Information* (MI) between two random variables $\mathbf{X}$ and $\mathbf{Y}$ quantifies how much information can be extracted about $\mathbf{Y}$ by observing $\mathbf{X}$ and is defined as:

$$\begin{aligned} MI\left(\mathbf{X};\mathbf{Y}\right) &= H\left(\mathbf{X}\right) - H\left(\mathbf{X}|\mathbf{Y}\right) \\ &= H\left(\mathbf{X}\right) + \sum_{\mathbf{x}\in\mathcal{X}} \mathsf{Pr}\left[\mathbf{X}=\mathbf{x}\right] \cdot \sum_{\mathbf{y}\in\mathcal{Y}} \mathsf{Pr}\left[\mathbf{Y}=\mathbf{y}|\mathbf{X}=\mathbf{x}\right] \cdot \log_2\left(\mathsf{Pr}\left[\mathbf{X}=\mathbf{x}|\mathbf{Y}=\mathbf{y}\right]\right). \end{aligned} \tag{1}$$

Introduced by McGill [McG54], interaction information is a multivariate generalization of mutual information for measuring dependence among multiple variables. The interaction information $MI(\{\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_n\})$ between $n+1$ random variables $\{\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_n\}$, denoted as $\{\mathbf{X}_{0:n}\}$ in the following sections, and the conditional interaction information $MI(\{\mathbf{X}_{0:n}\}|\mathbf{Y})$ are respectively defined as:

$$MI(\{\mathbf{X}_{0:n}\}) = \begin{cases} MI(\mathbf{X}_0;\mathbf{X}_1) & \text{if} \quad n=1, \\ MI(\{\mathbf{X}_{0:n-1}\}|\mathbf{X}_n) - MI(\{\mathbf{X}_{0:n-1}\}) & \text{for} \quad n \geq 2. \end{cases}$$

$$MI(\{\mathbf{X}_{0:n}\}|\mathbf{Y}) = \mathbb{E}_Y\left[MI(\{\mathbf{X}_{0:n}\})|\mathbf{Y}\right].$$

2.2 Learning Losses in Side-Channel Analysis

Profiled SCA can be formulated as a *classification problem*. Given an input, a neural network constructs a function $F_\theta : R^D \rightarrow R^{|\mathcal{Z}|}$ that computes an output called *a prediction*. During the training process, a set of parameters θ, called trainable parameters, are updated in order to generate the model. To solve a classification problem, the function F_θ must find the right prediction $z \in \mathcal{Z}$ associated with the input $\mathbf{t}$ with high confidence. To find the optimized solution, a neural network has to be trained using a profiled set of N_p pairs $(\mathbf{t}_i^p, z_i^p)$ where $\mathbf{t}_i^p$ is the i-th profiled input and z_i^p is the associated label. In SCA, the input of a neural network is a side-channel measurement and the related label is defined by the corresponding sensitive value. The input goes through the network and return a distribution that quantifies the probability of observing each hypothetical sensitive value. As a classical profiling attack, we can use the resulted probability distribution to compute the score for each key hypothesis and predict the correct targeted secret. To quantify the classification error of F_θ over the profiled set, a loss function has to be configured. Indeed, this function reduces the error of the model in order to optimize the prediction. For that purpose, the *backward propagation* [GBC16] is applied to update the trainable parameters (*e.g.* weights) and minimize the loss function. The classical loss function used in side-channel analysis is based on *cross-entropy*.

Definition 1 (Cross-Entropy)**.** Given a joint probability distribution of a sensitive cryptographic primitive Z and corresponding leakage $\mathbf{T}$ denoted as $\mathsf{Pr}[\mathbf{T}, Z]$, we define the Cross-Entropy of a deep leaning model F_θ as:

$$\mathcal{L}(\mathsf{Pr}[\mathbf{T}, Z], F_\theta) \triangleq \mathop{\mathbb{E}}_{\mathsf{Pr}[\mathbf{T}, Z]}\left[-\log_2 F_\theta\left(\mathbf{T}\right)[Z]\right].$$

Given a profiling set $\mathcal{T}$ of N_p pairs $(\mathbf{t}_i^p, z_i^p)_{1\leq i\leq N_p}$ and a classifier F_θ with parameter θ, the *Categorical Cross-Entropy* (CCE) loss function is an estimation of the *cross-entropy* such that:

$$\mathcal{L}_{CCE}(F_\theta, \mathcal{T}) = -\frac{1}{N_p}\sum_{i=1}^{N_p}\sum_{j=1}^{|Z|}\left(\mathbf{1}_{z_i^p=j}\cdot\log_2\left(F_\theta\left(\mathbf{t}_i^p\right)[j]\right)\right).$$

In other words, minimizing the categorical cross-entropy reduces the dissimilarity between the right distributions and the predicted distributions for a set of inputs. According to the *Law of Large Numbers*, the categorical cross-entropy loss function converges in probabilities towards the cross-entropy for any θ [SSBD14]. In [MDP19], Masure *et al.* demonstrate that minimizing the cross-entropy is asymptotically equivalent to maximizing the perceived information [RSVC+11]. Hence, some unexpected errors can be generated namely approximation, optimization and estimation errors. This loss was extended to the imbalanced data scenario [ZZN+20]. Recently, Zaid *et al.* propose a new loss, namely *Ranking Loss* [ZBD+20], adapted for the side-channel context.

Definition 2 (Ranking Loss [ZBD+20])**.** Given a profiling set $\mathcal{T}$ of N_p pairs $(\mathbf{t}_i^p, z_i^p)_{1\leq i\leq N_p}$, a classifier F_θ with parameter θ and a number of attack traces N_a such that $N_a|N_p$, we define the *Ranking Loss* (RkL) function as:

$$\mathcal{L}_{RkL}(F_\theta, \mathcal{T}, N_a) = \frac{N_a}{N_p}\sum_{i=1}^{N_p/N_a}\sum_{\substack{k\in\mathcal{K}\\ k\neq k^*}}\left(\log_2\left(1+e^{-\alpha\left(s_{N_a,i}(k^*)-s_{N_a,i}(k)\right)}\right)\right), \tag{2}$$

where $s_{N_a,i}(k) = \sum_{j=1}^{N_a} F_\theta\left(\mathbf{t}^p_{j+N_a\cdot(i-1)}\right)[f(p_j,k)]$ defines the output score [a] of the hypothesis $k\in|\mathcal{K}|$ for a given plaintext $(p_j)_{1\leq j\leq N_a}$ while α denotes one hyperparameter related to the sigmoid and approximates the identity function needed for estimating the success rate. For convenience, k^* is used to denote the class related to the correct label. Hence, k^* is also used to define the class associated with $f(p,k^*)$ such that f is a cryptographic primitive and p characterizes a plaintext value.

As illustrated in [ZBD+20], the selection of α greatly impacts the training process. Typically, $\alpha\in\{0.001, 0.01, 0.1, 1\}$. They demonstrate that the ranking loss maximizes the success rate for a given number of attack traces. Thus, this learning metric generates a model converging towards the optimal distinguisher introduced in [HRG14, BGH+17]. In the worst case, a model using the ranking loss function is as efficient as a model trained with the categorical cross-entropy [ZBD+20].

In [PCP20], Perin *et al.* use the categorical cross-entropy to evaluate the benefits of ensemble models applied to side-channel attacks against symmetric cryptographic implementations. In the next section, we explain the theoretical reasons why ensembling techniques are effective and introduce the diversity term that is essential to generate a powerful and efficient ensemble model.

2.3 Ensemble Models: A Source of Diversity

Reduction of the Global Error. In [TG96a, TG96b], Tumer and Ghosh provide theoretical observations for analyzing the interest of ensembling to solve a classification problem. They analyze the classification errors that are added to the *Bayes error* (*i.e.* the lowest possible error rate for any classifier of a random outcome) for an ensemble committee. Let $\mathcal{F} = \{F_{\theta_0}, F_{\theta_1}, \cdots, F_{\theta_{N_c-1}}\}$ be a set (or committee) of N_c classifiers (or members) with

[a] In [ZBD+20], the output score denotes the value before the softmax function. This choice is made to impact the training process accordingly to the relative order of the key hypotheses' relevance instead of the normalized probability distribution.

trainable parameter $(\theta_n)_{0\leq n<N_c}$ and E_{add} be the expected added error of the individual classifiers included in $\mathcal{F}$. In the following, F_{θ_n} will be denoted as F_n. The classifiers are assumed to have the same error. Tumer and Ghosh show the expected added error of the ensemble committee, denoted $E_{add,ens}$, as:

$$E_{add,ens} = E_{add}\left(\frac{1+\delta(N_c-1)}{N_c}\right), \tag{3}$$

where δ is a correlation factor that quantifies the error dependence among the classifiers and N_c is the number of classifiers (or members) in $\mathcal{F}$.

From Equation 3, we can easily evaluate the benefits of using ensemble methods to reduce the global error. If δ is 0, then the errors induced by the classifiers are independent and the ensemble expected added error is divided by N_c. Therefore, the global error will be N_c times smaller than the individual error provided by each classifier included in $\mathcal{F}$. On the other hand, if δ is 1, the errors induced by the classifiers are correlated and $E_{add,ens}$ characterizes the average error of each classifier. To insure uncorrelated errors, the classifiers included in the ensemble model must be diverse [Die00b].

Ensemble Diversity Definitions. Diversity has been recognized as a very important concept in classifier combination [CC00, Lam00]. However, in the machine learning literature, there is no strict common definition of what is perceived as diversity [Kun04]. For example, bagging [Bre96] and boosting [FS96] manipulate input data to promote diversity by choosing different subsets of input during the training process. In our paper, we define the diversity as follows:

Definition 3 (Diversity)**.** Given an ensemble model $\mathcal{F}$ composed by N_c committee members $(F_n)_{0\leq n<N_c}$, we define the diversity as the quantity measuring the difference in terms of prediction among the committee members.

This definition is not new and was already considered by the machine learning community (e.g. majority vote [MHA14], PAC-Bayesian theory [GMGA17], ...). From Definition 3, increasing the diversity reduces the overall ensemble error by distributing the wrong hypotheses uniformly once the combination of individual predictions is performed. In [FR05], Fumera and Roli found that the performance of ensembles depends on the performance of individual classifiers and their correlation. To efficiently promote the ensemble diversity, the output of the ensemble model $\mathcal{F}$ can be decomposed into three categories [XKS92, Kun04]. Let $\mathcal{F} = \{F_0, F_1, \cdots, F_{N_c-1}\}$ be a set (or committee) of N_c classifiers (or members) and $\mathcal{C} = \{c_0, c_1, \cdots, c_{|\mathcal{K}|-1}\}$ be a set of $|\mathcal{K}|$ labels (or classes). For a given input $\mathbf{t}$, we can define these categories as follows:

- **Abstract level**: the output of each classifier $F_n(\mathbf{t})$, denoted s_n, is included in $\mathcal{C}$. Thus, the N_c classifier outputs define a vector $s = [s_0, s_1, \cdots, s_{N_c-1}]^T \in \mathcal{C}^{N_c}$ that characterizes the output of $\mathcal{F}$.
- **Oracle level**: the output of $F_n(\mathbf{t})$ is 1 if $\mathbf{t}$ is correctly classified by F_n, and $F_n(\mathbf{t}) = 0$ otherwise. This representation is called *oracle* because we have to know the label for each input in order to configure the output.
- **Measurement level**: the output of $F_n(\mathbf{t})$ is defined by a vector of posterior probabilities $\big[\mathsf{Pr}[c_0|\mathbf{t}], \mathsf{Pr}[c_1|\mathbf{t}], \cdots, \mathsf{Pr}[c_{|\mathcal{K}|-1}|\mathbf{t}]\big]$. Hence, the output of the ensemble model $\mathcal{F}$ is characterized by N_c confidence vectors of size $|\mathcal{K}|$.

The measurement level contains the highest amount of information while the abstract level contains the lowest [XKS92]. In this paper, we want to precisely measure the diversity between each classifier of the committee. For that purpose, we focus only on the *posterior*

probability representation to evaluate the performance and the diversity of an ensemble model $\mathcal{F}$. These probabilities will be combined following the *Average Method* [XKS92] to define the overall performance of $\mathcal{F}$ but a comparison will also be provided with *Voting* in Section 6.

The diversity methods are legion and it could be hard to categorize them. In [LWC+19], Liu *et al.* decompose the diversity into three categories:

- **Type I diversity** characterizes the variety of committee members structure such as network architecture (e.g. MLP, CNN, RNN, ResNets, ...), weight initialization, training dataset, optimizer hyperparameters (e.g. optimizer algorithm, learning rate, number of epochs, ...).

- **Type II diversity** selects a subset of members that minimize their errors correlation from a pool of learners. Hence, the resulted ensemble model promotes independence between the members and tends to reduce the overall error.

- **Type III diversity** forces the set of learners $\mathcal{F}$ to decorrelate the errors generated by each committee member during the training process. Hence, an error decorrelation penalty term is incoporated in the loss function to create complementary members that reduce the overall error.

The type II and the type III diversities are both defined and quantified based on the disagreement among ensemble members. While the type II diversity captures the disagreement measure of each committee member after the training process for selecting a subset of learners, the type III diversity considers the *posterior probability representation* to create and promote interactions during the profiling phase. Hence, even if an ensemble model is composed by learners with a high disagreement measure, applying the type III diversity is useful to penalize the remaining error correlation. In this paper, we propose a new loss promoting the diversity during the training process (*i.e.* the type III diversity). This metric is based on the mutual information between an ensemble model and its related labels. The next section introduces the concept of mutual information ensemble diversity as a foundation of our proposition. In addition, to efficiently evaluate the overall benefits of using ensemble methods, we combine all types of diversity in Section 5.3.

2.4 Mutual Information Ensemble Diversity

Type III diversity can be characterized by the application of a specific loss function promoting the diversity between committee members. Unlike the correlation that is classically employed to measure the similarity between two entities, the mutual information captures non-linear statistical dependencies between variables. Hence, this measurement can be used as a real source of dependence information [KA14]. In [Bro09], Brown evaluates the benefits of using mutual information to improve ensemble models. He rewrites the ensemble problem as a communication channel problem. From an information theoretical point of view, let Y be a message sent through a communication channel and X be the received value such that X should be decoded to recover the input message Y. For that purpose, a decoding function $g(.)$ is defined such that an estimation of the message can be written as $\hat{Y} = g(X)$. From a machine learning perspective, X is the set of features characterizing the input of a learner $g(.)$ and Y is the true unknown label. During the training process, we want to minimize $\mathsf{Pr}\left[g(X) \neq Y\right]$. For any classifier g, [Fan61, HR70] provide theoretical bounds for $\mathsf{Pr}\left[g(X) \neq Y\right]$ such that:

$$\frac{H(Y) - MI(X;Y) - 1}{\log(|Y|)} \leq \mathsf{Pr}\left[g(X) \neq Y\right] \leq \frac{H(Y) - MI(X;Y)}{2}. \tag{4}$$

Hence, to minimize $\mathsf{Pr}\left[g(X) \neq Y\right]$, we have to maximize the mutual information between X and Y. In [Bro09], Brown proposes a solution to compute the mutual information between an ensemble model $\mathcal{F}$ and a set of true unknown labels Y.

Definition 4 (Mutual Information Ensemble Diversity [Bro09, ZL10])**.** Given an ensemble model $\mathcal{F}$ composed by N_c committee members $(F_n)_{0\leq n<N_c}$, a sensitive cryptographic primitive Z, we define the mutual information ensemble diversity as:

$$MI(\mathcal{F};Z) = \sum_{n=0}^{N_c-1} MI(F_n;Z) - \sum_{n=1}^{N_c-1} MI(F_n;F_{0:n-1}) + \sum_{n=1}^{N_c-1} MI(F_n;F_{0:n-1}|Z), \quad (5)$$

where $MI(F_n;Z)$ is called **relevancy**, $MI(F_n;F_{0:n-1})$ defines the **redundancy** and $MI(F_n;F_{0:n-1}|Z)$ characterizes the **conditional redundancy**.

The relevancy computes the mutual information between the n^{th} classifier of X and the target Z. The redundancy is independent of the class label Z and measures the interactions between all the classifiers. Hence a large $\sum_{n=1}^{N_c-1} MI(F_n;F_{0:n-1})$ indicates strong correlations between the classifiers. Finally, $\sum_{n=1}^{N_c-1} MI(F_n;F_{0:n-1}|Z)$ indicates that a strong class-conditional correlation is needed to perform an efficient ensemble model. However, from a practical perspective, it is quite difficult to estimate higher-order interaction information. Currently, there is no effective computational approach in the literature. Hence, Brown proposes to simplify Equation 5 by considering only pairwise components as follows [Bro09]:

$$MI(\mathcal{F};Z) \approx \sum_{n=0}^{N_c-1} MI(F_n;Z) - \sum_{n=0}^{N_c-2}\sum_{m=n+1}^{N_c-1} MI(F_n;F_m) + \sum_{n=0}^{N_c-2}\sum_{m=n+1}^{N_c-1} MI(F_n;F_m|Z), \quad (6)$$

where $MI(F_n;Z)$ computes the mutual information between the n^{th} classifier of F_n and the target Z, $MI(F_n;F_m)$ measures the mutual information between two models F_n and F_m and $MI(F_n;F_m|Z)$ measures the redundancy between two models F_n and F_m knowing Z.

Based on the pairwise approach, Equation 6 omits higher-order components. In the next section, we propose a loss that maximizes the pairwise mutual information between a committee $\mathcal{F}$ and a set of labels Z during the training process.

3 Ensembling Loss: A Pairwise Ensemble Diversity Metric

This section presents our main contribution: the *Ensembling Loss* (EL). In Section 3.1, we first define three sub-losses, namely *Relevance Loss*, *Conditional Redundancy Loss* and *Redundancy Loss*, derived from the mutual information ensemble diversity. This decomposition allows us to define the *Ensembling Loss* as a diversity learning metric. Then, Section 3.2 validates the theoretical aspects of the ensembling loss through visualization techniques.

3.1 Mutual Information Ensemble Diversity Estimation

This section proposes a loss derived from Equation 6 in order to maximize the pairwise mutual information and the diversity between the committee classifiers. To this end, we propose three losses namely **Relevance loss**, **Conditional Redundancy loss** and **Redundancy loss**. In order to achieve a general-purpose estimator, we base our propositions on the characterization of the mutual information as the Kullback-Leibler (KL-) divergence [KL51] between the joint distribution and the product of the marginals.

Relevance Loss. In Equation 6, the relevance $MI(F_n; Z)$ highlights the dependence of a learner $F_n \in \mathcal{F}$ and a label Z. Following [ZL10, Zho12], this term gives a bound on the accuracy of the individual classifiers. Hence, a large relevance is preferred to maximize the performance of the ensemble model. In [ZBD+20], Zaid *et al* propose the **Ranking Loss** to maximize a classical side-channel performance metric, namely *Success Rate* [SMY09]. Minimizing the ranking loss is asymptotically equivalent to maximizing the mutual information between a model and its related labels. The minimization of this loss function is exactly what the relevance quantifies in [Bro09]. Thus, given a set of N_p profiling traces, denoted $\mathcal{T}$, and a number of N_a attack traces such that $N_a | N_p$, we define the *Relevance Loss* as:

$$l^{\alpha}_{rel.}(F_n, \mathcal{T}, N_a) = \frac{N_a}{N_p} \sum_{i=1}^{N_p/N_a} \sum_{\substack{k=0 \\ k \neq k^*}}^{|\mathcal{K}|-1} \log_2 \left(1 + e^{-\alpha(s^{(n)}_{N_a,i}(k^*) - s^{(n)}_{N_a,i}(k))}\right), \tag{7}$$

where $s^{(n)}_{N_a,i}(k) = \sum_{j=1}^{N_a} F_n\left(\mathbf{t}^p_{j+N_a \cdot (i-1)}\right)[f(p_j, k)]$ defines the score related to the class k given a set of N_a traces, a classifier F_n and k^* the correct class. Finally, α denotes the hyperparameter of the sigmoid function that should be configured.

Minimizing Equation 7 tends to maximize the mutual information $MI(F_n; Z)$ through the minimization of the error induced by $\mathsf{Pr}[Z|\mathbf{t}]$. In other words, we want to penalize a model F_n when the correct label Z is not ranked as the highest hypothetical class. This penalization term depends on the distance between the score associated with the correct label Z and the other hypotheses. From a machine learning perspective, the maximization of $MI(F_n; Z)$ tends to generate compact clusters, one for each class. If False-Positives (FP) or False-Negatives (FN) appear during the training process, the ensemble model will be overconfident on its predictions and the resulted errors could be persistent. To reduce this effect, a solution is to provide diversity in order to limit the impact of these FP, FN examples. Hence, other losses defined below bring more diversity during the training process.

Remark 1. The relevance loss is actually the same as the ranking loss defined in [ZBD+20]. We reformulate it to facilitate the comprehension of the ensembling loss and the comparison with the mutual information ensemble diversity (see Definition 4) introduced by Brown [Bro09].

Conditional Redundancy Loss. The conditional redundancy $MI(F_n; F_m | Z)$ quantifies the dependence between F_n and F_m given a set of labels Z. This mutual information helps the committee members to converge towards the correct label hypothesis with the same confidence. Maximizing $MI(F_n; F_m | Z)$ is asymptotically equivalent to minimizing the error on $\mathsf{Pr}[F_m | F_n, Z]$ which defines the output probability of the model F_m given F_n and Z. In other words, we want to minimize the distance between the scores of F_n and F_m given the correct class. Thus, for a set of N_a traces, we introduce the *Conditional Redundancy Loss* as:

$$l^{\beta}_{cond.red.}(F_n, F_m, \mathcal{T}, N_a) = \frac{N_a}{N_p} \sum_{i=1}^{N_p/N_a} -\log_2 \left(e^{-\beta \left| s^{(n)}_{N_a,i}(k^*) - s^{(m)}_{N_a,i}(k^*) \right|}\right), \tag{8}$$

where the β parameter of the sigmoid function should be configured and $s^{(n)}_{N_a,i}(k^*)$ defines the score related to the class k^* given a set of N_a traces and a classifier F_n.

Through Equation 8, we want to penalize the learning process when the score $s^{(n)}_{N_a,i}(k^*)$ and $s^{(m)}_{N_a,i}(k^*)$ are different. Hence, we want to minimize the dissimilarity between the pairwise model F_n and F_m knowing Z. This will have the effect of increasing the confidence

of the network on the True-Positive (TP) and True-Negative (TN) examples. Consequently, we consolidate the good predictions with more persistency. However, this loss does not interact with the False-Positive (FP) and False-Negative (FN) examples. The following redundancy loss reduces this gap.

Redundancy Loss. The redundancy $MI(F_n; F_m)$ measures the pairwise dependence between all the committee members without considering the ground truth. A large mutual information induces a strong correlation among the pairwise classifiers and promotes similarities which is not desired when we want to construct an efficient ensemble model. Hence, we want to minimize this mutual information to improve the ensemble performance. The redundancy loss maximizes the distance between the score distribution of the models F_n and F_m. Therefore, we propose a loss penalizing the training process when this condition does not hold. We introduce the *Redundancy Loss* as:

$$l^{\gamma}_{red.}(F_n, F_m, \mathcal{T}, N_a) = \frac{N_a}{N_p} \sum_{i=1}^{N_p/N_a} \sum_{k=0}^{|\mathcal{K}|-1} \sum_{k'=0}^{|\mathcal{K}|-1} -\log_2 \left(1 - e^{-\gamma \left| s^{(n)}_{N_a,i}(k) - s^{(m)}_{N_a,i}(k') \right|}\right), \quad (9)$$

where the γ parameter of the sigmoid function should be configured.

Minimizing $MI(F_n; F_m)$ is equivalent to maximizing $H(F_m|F_n)$. Consequently, we want to increase the uncertainty of F_m given F_n. Through the minimization of Equation 9, we promote the cluster scattering and reduce the global confidence of the committee members on the False-Positives and False-Negatives to decrease their persistency.

Ensembling Loss. We integrate the mutual information ensemble diversity during the training process to promote the diversity between the committee members. Through our individual losses provided in Equation 7, Equation 8 and Equation 9, we formulate an *Ensembling Loss* (EL) that maximizes an estimation of the mutual information between an ensemble $\mathcal{F}$ and a label Z.

Definition 5 (Ensembling Loss - Our contribution)**.** Given a profiling set $\mathcal{T}$ of N_p pairs $(\mathbf{t}^p_i, z^p_i)_{1 \leq i \leq N_p}$, a set of classifiers $\mathcal{F} = \{F_0, F_1, \cdots, F_{N_c-1}\}$ and a number of attack traces N_a such that $N_a | N_p$, we define the *Ensembling Loss* (EL) function as:

$$\begin{aligned}\mathcal{L}_{EL}(\mathcal{F}, \mathcal{T}, N_a) = &\frac{1}{N_c} \sum_{n=0}^{N_c-1} l^{\alpha}_{rel.}(F_n, \mathcal{T}, N_a) \\ &+ \frac{2\mu}{N_c(N_c-1)} \sum_{n=0}^{N_c-2} \sum_{m=n+1}^{N_c-1} \left(l^{\gamma}_{red.}(F_n, F_m, \mathcal{T}, N_a) + l^{\beta}_{cond.red.}(F_n, F_m, \mathcal{T}, N_a) \right), \quad (10)\end{aligned}$$

where μ quantifies the impact of the diversity term during the training process, α (resp. β, γ) is a hyperparameter that configures the relevance loss (resp. conditional redundancy and redundancy losses) effect.

We normalize each term of the ensembling loss to reduce the impact of exploding gradient. Appendix A highlights the benefits of each individual loss from a training perspective. Through this study, the reader can understand how the network would train if the conditional redundancy loss or the redundancy loss are individually used. Finally, in the following sections, the number of attack traces N_a will be configured to 1 during the profiling phase as in [ZBD+20].

Remark 2. Due to the wide range of hyperparameters (*i.e.* $\mu, \alpha, \beta, \gamma$), the ensembling loss seems difficult to tune. From a practical perspective, these hyperparameters are dataset-dependent. Hence, it seems very challenging to define a generalized configuration

for all types of implementations because it highly depends on the number of classes $|\mathcal{K}|$, the noise induced in each trace, the implemented countermeasures, the targeted algorithm (e.g. RSA, ECC), etc. However, during our experiments, the tuning process was not a pitfall. Indeed, in the following section, α, β, γ values follow the strategy defined in [ZBD+20]. Hence, they are configured in $[0.001, 0.1]$. In opposition, μ is not optimized in this work and always equals 1.

Remark 3. As our framework is generic, we argue it is adequate to target private-key implementations, in particular AES [DR02] and DES [Des77] (*i.e.* $|\mathcal{K}| = 256$). However, the training time increases exponentially with the number of output classes $|\mathcal{K}|$. Hence, from a practical perspective, the application of the *Ensembling Loss* seems more suitable for low multiclass problems (*i.e.* $|\mathcal{K}| \leq 5$) such as attacks against asymmetric implementations. This proposition fits with asymmetric algorithm implementations which consider low multiclass problems. Finally, even if this work is only focusing on the side-channel context, the *Ensembling Loss* can be used to solve any machine learning problems (*i.e.* image classification, image recognition, fraud detection, ...).

3.2 Visual Validation of the Ensemble Diversity

Diversity among the committee members is deemed to be a key issue in ensemble learning and should reduce the global error (see Section 2.3). In this section, we want to validate the theoretical observations provided in Section 3.1. Hence, we analyze the diversity evolution depending on the loss used during the training process. The ensemble model can be trained to follow one of the next three processes:

- **Independent learning strategy** – There is no interaction among the classifiers. For example, each classifier could be trained on different training set to reduce the features' correlation [Bre96];
- **Sequential training** – This process induces a set of learners that are trained sequentially on data sets with entirely different distributions;
- **Simultaneous ensemble learning** – a set of committee members are trained interactively to promote uncorrelation and diversity.

In this paper, we focus on the simultaneous training strategy for allowing interaction between the committee members during the training process. This strategy fits perfectly with the ensembling loss. Furthermore, it is helpful to promote the diversity between the members even if similar architectures are used.

Dataset setup for visualization. Assessing the benefits of the ensembling loss can be illustrated through the t-SNE visualization [vdMH08] and diversity measures [KW03]. For that purpose, we use a secure RSA dataset with three classes such that each input is associated with one of these labels (see Section 4.1 for more details on the dataset). The ensemble model is configured with 5 members (*i.e.* $\mathcal{F} = \{F_0, F_1, F_2, F_3, F_4\}$) such that each of them has the same architecture. Generating 5 committee members with the same architecture is helpful to efficiently evaluate the suitability of the ensembling loss in contrast with the categorical cross-entropy and the ranking loss. These members are CNNs architectures with 1 convolutional block based on 2 filters of size 1, a BatchNormalization layer [IS15] and an average pooling layer with stride 2. Then, a flatten layer is applied to reduce the space dimension of the convolutional part. Finally, a predictive layer is applied with a softmax function. The optimizer hyperparameters are set such that each network is trained during 40 epochs, with a batch-size of 128, a learning rate set to 0.001 and the Adam optimizer [KB15].

Remark 4. In the following sections, we only consider 5 committee members because this configuration provide us the best trade-off between training time and network performance. A deeper investigation is performed in Section 6.1 to evaluate the impact of the number of committee members on the ensemble accuracy.

t-distributed Stochastic Neighbor Embedding (t-SNE) visualization. Introduced in [vdMH08], the t-SNE visualization tool maps high-dimensional data into two or three-dimensional space while preserving local structure and also revealing important global structure (e.g. clusters). T-SNE employs a nonlinear and iterative process to convert similarities between data points to joint probabilities and tries to minimize the KL-divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. This representation is helpful to evaluate the network capacity to distinguish each class and validate the theoretical approach presented in Section 3.1.

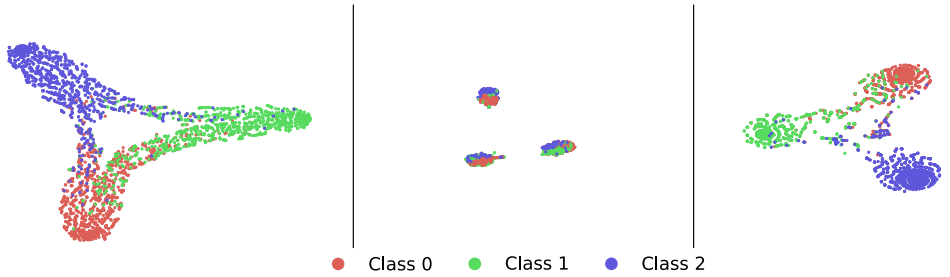

Figure 1: t-SNE embeddings. Left: Cross-Entropy Loss. Middle: Ranking Loss. Right: Ensembling Loss.

Figure 1 illustrates the t-SNE visualizations depending on the loss used during the training process. When the cross-entropy is considered, we estimate that the network is not trained enough to efficiently discriminate each class. Indeed, there are many connections between each class leading to a loss of the global performance. Through this visualization, we can question the relevance of the cross-entropy in our context. Hence, many FP and FN can badly influence the global performance of the model.

On the other hand, the ranking loss [ZBD+20] generates three separate clusters. As mentioned in Section 3.1, the ranking loss can be formulated as the relevance loss (see Equation 7). Through the minimization of this function, we minimize the conditional entropy $H(Z|F_n)$ which promotes the generation of three compact clusters. Hence, Figure 1 confirms the theoretical results of the previous section. The ensemble model is overconfident in the features captured during the training process. Consequently, it detects discriminative patterns to avoid connections between each cluster. However, following the t-SNE illustration, the FP and FN induced by the ranking loss are persistent and seem difficult to detect. Indeed, these errors are fully included in a wrong cluster. This phenomenon can be explained by the overfitting effect. Using more training example could be useful to reduce this impact and reduce the error rate. However, when the number of profiling traces is limited (as often in practice), a solution has to be found to improve the ensemble model.

The best solution should create three separate clusters when the ensemble model is confident in its prediction while, the errors or the uncertain predictions should convergence towards the equidistant point of the centroid of the clusters. These examples are called *data uncertainty*. Introduced in [MMG20], data uncertainty is the irreducible uncertainty in predictions which arises due to the complexity or noise in the data. The ensembling loss converges towards this solution. Indeed, in Figure 1, the combination of the relevance loss, the conditional redundancy loss and the redundancy loss creates three separate clusters

(see Appendix A for deeper details). When the network is confident in its predictions, it will assign the related examples to the correct class. However, the ensembling loss creates some connections between the clusters which seem defined by the data uncertainty. This result tends to reduce the number of consistent FP and FN such that few errors can be detected on each cluster in contrast with the cross entropy or the ranking loss. However, the t-SNE does not provide information related to the diversity growth. To validate the suitability of the ensembling loss, we evaluate its model's diversity against the cross-entropy and the ranking loss.

Diversity visualization. As explained in Section 2.3, diversity has a crucial impact on the ensemble model's performance. Conventionally the diversity measures can be decomposed into two categories:

- **Pairwise measures** that compute the relationship between two learners, and then average all the pairwise measurements to define the overall diversity of an ensemble model $\mathcal{F}$ (Disagreement measure [Ska96, Tin98], Q-statistic [Yul00], Correlation coefficient [SS73], κ-statistic [Coh60], Double-Fault measure [GR01], ...);
- **Non-Pairwise measures** that assess the ensemble diversity directly rather than by averaging pairwise measurements (Kohavi-Wolpert Variance [KW96], Interrater agreement [Die00a, FLP03], Entropy [CC00], ...).

One advantage of pairwise measures it that they can be easily visualized and interpreted. Choosing a specific pairwise measure does not make significant difference in our experiments, so we chose the fraction of disagreement for simplicity.

Let $N^{ab}_{n,m}$ be the joint counts between two learners F_n and F_m. We denote $a = 0$ (resp. $b = 0$) if F_n (resp. F_m) wrongly predicts a value and $a = 1$ (resp. $b = 1$) otherwise. For example, $N^{01}_{n,m}$ defines the number of elements such that F_n obtains an incorrect value for a given input while F_m correctly predicts the related class for the same input.

Definition 6 (Disagreement Measure [Ska96, Tin98]). Given two classifiers F_n and F_m, the disagreement measure defines the proportion of examples on which these classifiers make different predictions:

$$Dis(F_n, F_m) = \frac{N^{01}_{n,m} + N^{10}_{n,m}}{N^{11}_{n,m} + N^{10}_{n,m} + N^{01}_{n,m} + N^{00}_{n,m}}. \tag{11}$$

This metric is 0 when two functions are making identical predictions, and 1 when they differ on every single examples in the test set. Hence, the larger the value, the larger the diversity. From an ensembling perspective, we want to generate a set of classifiers $\mathcal{F}$ maximizing the disagreement measure such that each individual learner keeps a high performance for classifying unseen examples. In [FHL19], Fort *et al.* propose to plot a normalized disagreement measure with respect to the accuracy of each classifier. The diversity measure is normalized by the error rate to prevent the case where random predictions provide the best diversity. From the set of classifiers $\mathcal{F}$, one member is randomly picked to be used as the base model. This model is denoted as F_n. Then, we calculate the diversity measure of other ensembling members against the base model.

Figure 2 illustrates the diversity of each model of $\mathcal{F}$ against F_n. In this figure, the y-axis characterizes the fraction of labels, returned by each model of $\mathcal{F}$, that differs from F_n while the x-axis defines their validation accuracy. Consequently, the sample with a 0 y-axis value defines F_n. Three ensemble models are generated with the three losses used in order to investigate the benefits of the ensembling loss. In [FHL19], Fort *et al.* propose a theoretical approach to explain the results obtained in Figure 2. Let F_n and F_m be two committee members from $\mathcal{F}$. If F_n and F_m have identical validation accuracy and high diversity then,

they converge towards different local optimum with identical depth. In opposition, if F_n and F_m have identical validation accuracy and a low diversity then, they converge towards the same local optimum. Consequently, from a loss landscape perspective, it sounds beneficial to construct an ensemble model with high diversity members such that their prediction distributions and their selected features differ from each other. In contrast, the accuracy of individual models does not reflect the performance of the ensemble committee. Indeed the combination of poor performance (*i.e. weak*), but complementary, classifiers can generate a very effective ensemble model. While the same configuration with effective, but correlated, classifiers is not beneficial for an ensembling approach. Consequently, an ensemble model composed by weak classifiers can outperform a combination of effective individual models.

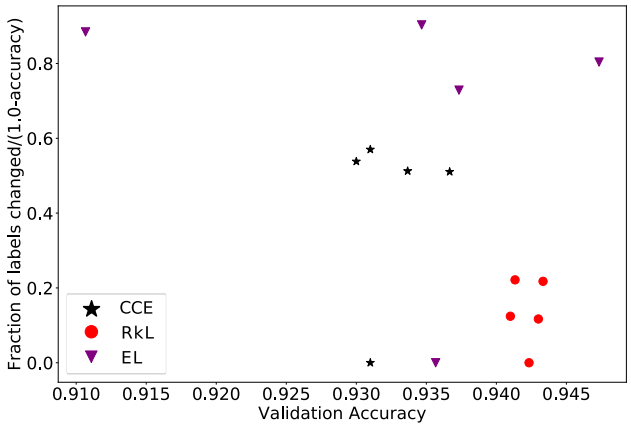

Figure 2: Diversity versus label accuracy plots for 3 ensemble models trained on Categorical Cross-Entropy (CCE), Ranking Loss (RkL) and Ensembling Loss (EL).

The ensemble model trained with the ranking loss provides the worst diversity scenario. Even if individual classifiers are more efficient than most of the other learners (*i.e.* validation accuracy $> 94\%$), the lack of diversity is an issue for developing uncorrelated members. Indeed, following [FHL19], all the committee members converge towards the same local optimum. Hence, a lack of complementarity can be exposed when the adversary only considers the ranking loss. Consequently, the resulted ensemble model performance should be equal to the average accuracy of its members. For the cross-entropy loss function, the members are more diverse than the ranking loss ensemble model. Consequently, the resulted learners are less correlated and the resulted probability combination should reduce the overall error. Finally, in comparison with the categorical cross-entropy and the ranking loss, the ensembling loss provides the most diverse models. Indeed, in Figure 2, the normalized diversity measure is the highest for the ensembling loss model. This observation is confirmed with the κ-statistic measure in Appendix B Figure 7. Interestingly, even if the committee members have the same architecture, the ensembling loss provides a clear diversity benefit. Hence, from a loss landscape perspective, the ensembling loss helps the committee members to converge towards independent local optimum with different depths.

These observations validate the theoretical observations introduced in Section 3.1. Indeed, using the ensembling loss increases the diversity between ensemble members to reduce the correlation between the errors made in order to propose an efficient ensemble model. In the next section, we evaluate this diversity gain on the practical ensembling performance.

4 Experimental Settings

The experiments are implemented in Python using the *Keras* library [C+15] and are run on a workstation equipped with 32GB RAM and a NVIDIA GTX1080Ti with 11GB memory.

4.1 Dataset: Secure RSA Implementation

Target presentation. Introduced in [CCC+19], the targeted RSA implementation is based on a *Left-to-Right Square & Multiply Always* exponentiation algorithm [Cor99] combined with three countermeasures: input randomization, modulus randomization and exponent randomization. The software part of the targeted RSA implementation does not provide specific security mechanisms to defeat horizontal or address-bit side-channel attacks. This choice has been done deliberately by CryptoExperts' team[b] who was responsible for the development of the RSA software part. This paper highlights that the application of advanced deep learning-based side-channel attacks makes security mechanisms against horizontal and address-bit attacks mandatory to reduce the adversary's scope.

For two 512 bits primes p and q, the combination of the three masking countermeasures corresponds to the following equation:

$$m^d \bmod N = ((m + k_1 \cdot N)^{d+k_2 \cdot \phi(N)} \bmod (k_0 \cdot N)) \bmod N, \tag{12}$$

with k_0, k_1, k_2 three random values of bit-length 64, $N = p \times q$ the modulus of $1,024$ bits. More details on the countermeasures and their benefits are provided in [CCC+19].

In the *Square & Multiply Always* algorithm (see [CCC+19, Algorithm 1]), Carbone *et al.* identify a vulnerability related to the manipulation of an index named seg_{free}. Indeed, this index stays unchanged for two consecutive exponentiations if the related exponent bit equals 1. If an adversary retrieves the value of this index, he can gradually learn the entire exponent bits except for the last one. For each consumption trace, this index value is defined in $\{0, 1, 2\}$ (see [CCC+19, Equation 5]). Consequently, we consider a multi-class classification problem with 3 outputs. For deeper information on the device under test, we suggest the readers to refer to [CCC+19].

Neural Network Architecture. While the original network performs very well ($= 99.91\%$), we decide to reduce its complexity while preserving the related performance. The network we used is composed of one convolutional block with 2 filters of size 1, one batch normalization layer [IS15] and an average pooling. Then a flatten layer is applied to connect the detected features to a predictive layer configured with 3 outputs defining the value of seg_{free}. Optimization is done using the Adam optimizer [KB15] approach on a batch-size of 128 and the learning rate is set to 10^{-3}. The batch-size and the learning rate follow the values provided in [CCC+19]. The optimization of these hyperparameters is not considered in this paper. We use the SeLU activation function to avoid vanishing and exploding gradient problems [KUMH17]. In the following sections, we only keep the model achieving its best performance (e.g. accuracy) over 100 epochs. This new model has a similar performance to the architecture proposed in [CCC+19] ($= 99.89\%$) while being much more efficient computational wise (*i.e.* $1,950,323$ against $39,015$ trainable parameters). In this paper, we want to evaluate the suitability of the ensemble models when the number of profiling traces is limited (as often in practice). Hence, we only use $30,000$ profiling traces and $3,000$ validation traces instead of using the $750,000$ traces considered by [CCC+19]. However, when an adversary trains a model with the $30,000$ raw profiling traces of $13,000$ samples, he already generates a classifier with very high performance ($= 98.30\%$). Hence to efficiently evaluate the suitability of the ensembling loss, we add Gaussian noise $\mathcal{N} \sim \mathcal{B}\left(0, \sigma^2\right)$ such that σ defines the standard deviation of

[b] https://www.cryptoexperts.com/

the noise. Table 1 shows the evolution of the accuracy depending on the added noise on the secure RSA dataset. In the following sections, σ is set to 6 in order to evaluate the benefits of the ensembling loss against the categorical cross-entropy and the ranking loss. The model trained with the categorical cross-entropy is considered as the state-of-the-art result because it uses the classical learning metric in the side-channel context. This result will be considered as our reference in order to highlight the performance provided by the ensembling loss.

Table 1: Evolution of accuracy depending on σ ($30,000$ profiling traces & $3,000$ validation traces)

Accuracy \ σ	0	10^{-3}	10^{-2}	10^{-1}	1	6	8	10	50
Categorical Cross-Entropy	98.30%	97.60%	97.77%	96.33%	95.70%	92.50%	85.80%	80.60%	41.77%

Evaluation metrics. While the accuracy is questioned when symmetric cryptographic implementations are considered [PHJ+19], it is totally relevant to assess the training process on asymmetric datasets. As previously mentioned, an adversary exploits the index seg_{free} such that 3 values can be assigned. We denote Acc_{label} the accuracy expressing the capacity of the network to retrieve the correct value of seg_{free} for a given leakage trace. This metric is used to mitigate the underfitting and overfitting issues.

However, an attacker wants to retrieve secret key bits. Hence, we have to convert the balanced ternary representation (*i.e.* $\{0, 1, 2\}$) into a binary representation (*i.e.* $\{0, 1\}$). We denote Acc_{bit} the accuracy expressing the capacity of the network to retrieve the amount of correct bit values. Its related error rate is denoted ϵ_{bit}.

4.2 Practicability and Remaining Brute-force Complexity

If the resulted Acc_{bit} is less than 100%, the adversary has to perform additional operations to retrieve the full secret key. While no theoretical results link the accuracy and the remaining operations, we experimentally evaluate how the accuracy impacts the final attack complexity. This is an open problem in the literature.

Given a number N_{op} of remaining operations, we define the brute-force complexity as $\log_2(N_{op})$. The European SOG-IS scheme[c] considers that a maximum brute-force complexity of around 2^{100} operations is practical. Hence, we consider this threshold to evaluate if an attack becomes feasible. Note that the notion of time complexity is independent of the computational power available to the attacker. In the following sections, we consider three complexity measures depending on the attack scenario:

- *Naive Complexity* – Given a secret exponent of K bits, a blinding scalar of bit-length R and an error rate ϵ_{bit}, the *Naïve Complexity*, denoted $\mathcal{C}_{NC}$, is defined as the worst-case scenario such that:

$$\mathcal{C}_{NC}\left(K, R, \epsilon_{bit}\right) = \log_2\left(\sum_{i=0}^{\lceil (K+R)\times\epsilon_{bit}\rceil} \binom{K+R}{i}\right). \tag{13}$$

  In this scenario, the adversary cannot locate the wrongly predicted bits induced by the attack. Hence, he has to compute all the combinations for each wrong assumption in order to correct the remaining errors.

[c]The Senior Officials Group Information Systems Security (SOG-IS) agreement defines a set of requirements and evaluation procedures related to cryptographic aspects of Common Criteria security evaluations of IT products and mutually agreed by SOG-IS participants. Participants in this Agreement are government organisations or government agencies from countries of the European Union or EFTA (European Free Trade Association). The interested readers may find useful information in https://www.sogis.eu/index_en.html.

- 2^n-*Complexity* – Given a secret exponent of K bits, a blinding scalar of bit-length R and an error rate ϵ_{bit}, the 2^n-*Complexity*, denoted $\mathcal{C}_{2^n}$, is defined as the best-case scenario such that:

$$\mathcal{C}_{2^n}(K, R, \epsilon_{bit}) = \lceil (K + R) \times \epsilon_{bit} \rceil. \tag{14}$$

  In this scenario, the adversary perfectly knows the location of each potential error. Thus, each assumption error has 2 possible values and the resulted number of remaining operations is $2^{\lceil (K+R)\times\epsilon_{bit} \rceil}$. In the following, an attack that can be performed with 2^n-*Complexity* is called a 2^n-*Attack*.

- *Alternate Attack Complexity* – Introduced by Schindler and Wiemers in [SW14], the Alternate Attack (AA) targets RSA modular exponentiation protected with an exponent blinding. From this attack, we propose a complexity measure to estimate its practicability. Our proposition is developed in Appendix C. Given a blinding scalar of bit-length R, a secret exponent of K bits, an error rate ϵ_{bit} and a number of attack traces N_a, the *Alternate Attack Complexity*, denoted $\mathcal{C}_{AA}$, is defined as:

$$\mathcal{C}_{AA}(K, R, \epsilon_{bit}, N_a) = \log_2 \left(N_a \cdot 2^{K-s+1} \cdot \sum_{i=0}^{\lceil (R+1)\times\epsilon_{bit} \rceil} \binom{R+1}{i} \right). \tag{15}$$

  In [SW17], Schindler and Wiemers set $s = K - R + 2$ and $t_0 = 2$ for $R = 32$. Even if using the same parameters is restrictive when $R = 64$, these conditions are respected in Appendix C. In the following, N_a characterizes the number of attack traces that are needed for retrieving the entire bits of $\phi(N)$ (see Appendix C). We consider an alternate attack as ineffective if the success rate related to $\phi(N)$ is less than 100% when 300 successive alternate attacks are performed.

All these complexity measures are helpful to evaluate the efficiency of an attack. These tools are suited to highlight the benefits of the ensembling loss on different attack scenarios and prove the negligible impact of the growth of the training time.

5 Experimental Results

This section proposes an experimental comparison between the categorical cross-entropy, the ranking loss and the ensembling loss when ensemble models are considered. In Section 5.1, we evaluate the complementarity of committee members depending on the loss used. Then, in Section 5.2, we combine the *type I diversity* with the different learning metrics to illustrate its impact of the resulted performance. In Section 5.3, all the diversity types are combined to exploit the entire benefits of the ensemble methods and highlight the improvement in the resulted side-channel attack complexity.

In the following sections, $CCE_{i,j}$ (resp. $RkL_{i,j}$, $EL_{i,j}$) denotes an ensemble model trained with the categorical cross-entropy (resp. the ranking loss, the ensembling loss), composed by i committee members such that the type j diversity is performed. Due to the interactions between the committee members during the training process, the ensembling loss can be considered as the only learning metric promoting the type III diversity.

Remark 5. In this paper, we only consider CNN architectures because the benefits of these networks were demonstrated in the side-channel context [MPP16, CDP17, KPH+19, CCC+19, ZBHV19, Mag20]. Obviously, the combination of diverse network architectures (*Multi-Layer Perceptrons*, *Recurrent Neural Networks* [SP97, HS97], *Residual Neural Networks* [HZRS16], *U-Nets* [RFB15], etc.) can also be considered to promote complementary features selection.

Remark 6. In [DDFP20], Destouet *et al.* investigate a solution that consists of the aggregation of multiple models targeting different sensitive value (*i.e.* hamming weight, first big-endian bit, identity). In this paper, we assume that all the learners are trained on the same single label.

5.1 Learning Ensemble Diversity

This section evaluates the benefits of using the ensembling loss instead of the categorical cross-entropy or the ranking loss when ensemble models are considered. To assess the diversity growth, we generate an ensemble model composed of 5 committee members with the same architecture (see Section 4.1). Consequently, the diversity provided by the following ensemble models only depends on the loss used.

Table 3 illustrates the performance evolution depending on the diversity type and the learning metric applied. If the adversary only considers the state-of-the-art result, he trains a unique model with the categorical cross-entropy (*i.e.* CCE_1) to perform its attack. Recently, Zaid *et al.* proposed the *Ranking Loss* for the side-channel context [ZBD+20]. However, their work was only focused on symmetric implementations. Here, we extend this work by investigating the benefits of using this loss to evaluate asymmetric implementations. In our scenario, the ranking loss can be considered as more effective than the categorical cross-entropy (see Table 3). While a classifier trained with the categorical cross-entropy loss function does not provide powerful models (*i.e.* $\mathcal{C}_{\{NC,2^n,AA\}} \geq 100$), using the ranking loss an attacker can potentially break the RSA implementation (*i.e.* $\mathcal{C}_{2^n} \leq 100$).

When 5 committee members are considered in the ensemble model, we can observe a meaningful improvement. Even if the training time is multiplied by 9 in the worst case, it stays reasonable from a practical perspective. In opposition, Acc_{label} is increased by up to 2.69% and an adversary can extend its attack scenario. Following the SOG-IS recommendations, an adversary can successfully perform an alternate attack if he applies the ensembling loss to train its ensemble model while the state-of-the-art (*i.e.* CCE_1) result cannot. This result highlights the benefit of using the ensembling loss in terms of ensemble performance. In addition, considering the ensembling loss reduces $\mathcal{C}_{2^n}$ by 25. Hence, the theoretical features of the ensembling loss, which are validated through the visualizations of Section 3.2, translate an actual gain in model accuracy as well as a realistic improvement for a full attack scenario. The ensembling loss increases the overall diversity and reduces the global error rate induced in the ensemble model. Thence, the ensembling loss is helpful to promote the complementary between the committee members.

5.2 Ensembling Loss Combined with Type I Diversity

As mentioned in Section 2.3, the type I diversity refers to the heterogeneity between the committee members' structure. This diversity is employed by Perin *et al.* [PCP20] to argue the generalization improvement induced by this ensemble method. In this section, we propose to combine the type I diversity with the different loss functions to evaluate the resulted gain in attack complexity. For that purpose, we randomly generate 5 networks with a wide range of hyperparameters (details are provided in Appendix D Table 9). In [Zho12], Zhou recommends the configuration of heterogeneous networks with high individual performance. From a bias-variance trade-off perspective, this procedure is powerful to reduce the bias as well as the variance by aggregation. Even if this solution can be intuitive, this is not necessary the best one as discussed in Section 3.2.

From a diversity perspective, using efficient heterogeneous networks seem to increase the uncorrelated errors. Through Figure 3, combining the type I diversity with the ensembling loss reduces the overall κ-statistic measure. Following Appendix B Definition 7,

this observation confirms the gain in diversity. This result can also be verified with the disagreement measure (see Appendix B Figure 8).

From a performance perspective, the individual committee members do not exceeded 94.33% for retrieving the bits of blinding exponent when the ranking loss is considered (see Appendix D Table 9). However, applying the ensembling loss adjusts the efficiency of each learner to increase their complementarity. Indeed, the most powerful member finds 95.84% of all bits while the least significance one finds only 88.21% of all bits. The interaction between the committee members during the training process tends to accentuate the discrepancy in order to force the gain in diversity. Table 3 illustrates the benefits of combining the type I diversity with the ensembling loss from a performance perspective. Adding type I diversity reduces the remaining attack complexity regardless of the adversary capacity. Finally, even if the resulted training time increases, it stays marginal related to the gain in attack complexity. Indeed, depending on the scenario, the attack can be performed by up to 2^{50} operations. In comparison with the previous state-of-the-art result (*i.e.* CCE_1), the number of operations is reduced by 2^{58} while the training time is only increased by 10.

5.3 Combining All Types of Diversity

The type I+II diversity consists of the selection of members from a pool such that the diversity measure is maximized between all the learners. The pool members are selected by randomly picking out the hyperparameters from ranges defined in Table 2. The resulted pool is composed by 100 members. As recommended in [LWC+19], we retain a set of classifiers with high performance (*i.e.* $Acc_{label} > 85\%$) such that their disagreement measure is maximized. The 5 selected architectures are identified in Appendix D Table 10.

Remark 7. In some cases (e.g. boosting [CG16]), weak learners (*i.e.* models that are only slightly better than random guessing) can be helpful to increase the performance of the ensemble model. The benefit of these strategies is considered as out of our scope.

Table 2: Range of hyperparameters selection

	Values
$N_{filters}$	$\{2, 4, 8, 16, 32, 64\}$
filter size	$\{1, 5, 11, 21, 43\}$
$N_{conv.blocks}$	$\{1, 2, 3, 4, 5\}$
pooling operation	$\{Average, Max\}$
pooling stride	$\{2, 4, 6\}$
$N_{FC\ layers}$	$\{0, 1, 2, 3\}$
$N_{nodes\ per\ FC\ layers}$	$\{2, 4, 8, 16, 32, 64\}$

The type I+II diversity promotes the error uncorrelation between the individual committee members. Following the κ-statistic measure (see Figure 3), the diversity brought by the type I+II is significant in comparison with the previous experiments. Indeed, the overall κ-statistic measure is reduced in comparison with the other experiments. This observation can also be made with the disagreement measure (see Appendix B Figure 8). When the ranking loss or the categorical cross-entropy is considered, even if no interaction is proposed between the committee members during the training process, using the type I+II diversity is useful to bring more diversity in the ensemble model. However, combining the type I+II diversity with the ensembling loss accentuates the gain in diversity in order to generate a more powerful model.

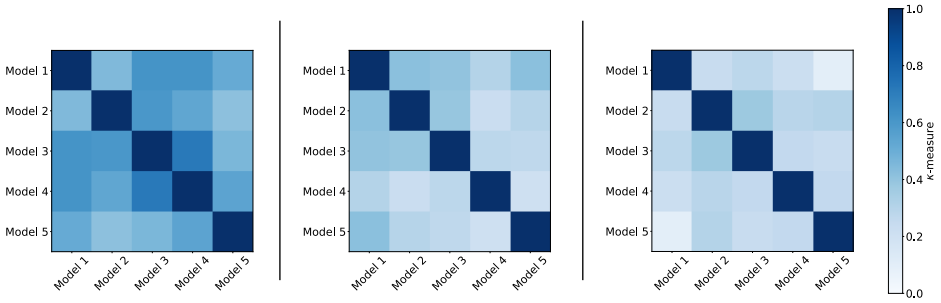

Figure 3: κ-statistic. Left: Ensembling Loss. Middle: Ensembling Loss + Type I diversity. Right: Ensembling Loss + Type I + II diversity.

From Table 3, we observe a significant improvement when the ensembling loss is used in comparison to the ranking loss and the categorical cross-entropy. Generating interactions between the committee members provides more consistency during the training process. As mentioned in Section 3.2, the ensembling loss leads to converge the uncertain predictions towards the equidistant point of the centroid of the clusters. Thus, the impact of the FP and FN is reduced when the ensembling loss is performed. Combining all the diversity techniques provides the most effective model in terms of performance. While an ensemble model trained with the ranking loss needs 2^{56} operations to retrieve the remaining bits in the best case scenario, the addition of the ensembling loss with the type I+II diversity needs only 2^{34} operations. Even if the resulted training time is multiplied by 3, the gain in performance is significant to justify the benefits of the ensembling loss.

Table 3: Performance evaluation depending on the diversity's type (Average over 10 physical traces of 1,088 bits each). Green (resp. Red) cells are considered as practicable (resp. unpracticable) following the SOG-IS recommendations.

Section	Model	Acc_{label}	Acc_{bit}	ϵ_{bit}	$\mathcal{C}_{NC}$	$\mathcal{C}_{2^n}$	$\mathcal{C}_{AA}$	Training Time
Section 5.1	CCE_1	92.50%(±2.06%)	90.16%(±2.32%)	0.0984	500.08	108	102.74	$86s$
	RkL_1	94.03%(±1.85%)	91.26%(±2.21%)	0.0874	460.66	96	102.74	$81s$
	CCE_5	93.60%(±1.91%)	91.04%(±2.23%)	0.0896	467.39	98	102.74	$450s$
	RkL_5	94.33%(±1.81%)	91.66%(±2.16%)	0.0834	443.55	91	99.37	$340s$
	$EL_{5,III}$	95.19%(±1.67%)	92.45%(±2.07%)	0.0754	415.25	83	96.06	$740s$
Section 5.2	$CCE_{5,I}$	95.31%(±1.65%)	93.28%(±1.96%)	0.0672	381.96	74	92.43	$688s$
	$RkL_{5,I}$	95.93%(±1.55%)	94.48%(±1.79%)	0.0552	330.81	61	92.43	$780s$
	$EL_{5,I+III}$	96.57%(±1.42%)	95.49%(±1.62%)	0.0451	284.21	50	88.45	$884s$
Section 5.3	$CCE_{5,I+II}$	96.23%(±1.49%)	94.66%(±1.76%)	0.0534	322.58	59	88.45	$2,116s$
	$RkL_{5,I+II}$	96.27%(±1.48%)	94.94%(±1.71%)	0.0506	310.01	56	88.45	$1,092s$
	$EL_{5,I+II+III}$	97.33%(±1.26%)	96.96%(±1.34%)	0.0304	209.52	34	80.71	$3,392s$

As a conclusion, combining all the diversity techniques provides a clear advantage from a side-channel point of view. Indeed, when the type I+II diversity techniques are combined with the ensembling loss (*i.e.* type III diversity), we promote the diversity between the classifiers in order to reduce the global error. In comparison with the previous state-of-the-art result (*i.e.* CCE_1), Acc_{bit} is increased by 6.8% and the number of remaining operations is reduced by $2^{290.56}$ (resp. 2^{74} and $2^{22.03}$) when the adversary wants to perform a naive attack (resp. a 2^n-attack and an alternate attack). Even if the training time is increased by up to 39.44, it stays negligible regarding the gain to perform the full attack. Indeed, following the SOG-IS recommendation, the previous state-of-the-art result considers the RSA implementation as secure while combining the different diversity techniques leads an adversary to retrieve the secret exponent. Hence, the combination of type I+II with the ensembling loss should be considered during the evaluation of the asymmetric implementations to generate more powerful attacks.

6 Discussion

This discussion evaluates the classical ensemble methods (*i.e.* Bagging [Bre96], Boosting [FS96, CG16]), the classifier fusion's techniques (*i.e.* average accuracy, voting) and the impact of the number of committee members. Then, we evaluate the benefits of the ensembling loss for a binary classification problem. Obviously, the results provided in this paper can be improved by using additional techniques defined as suitable in side-channel context [CDP17, PHJ+19, WJB20, Mag20, PCBP20].

6.1 Classical Ensemble Methods

Ensemble Methods. Traditionnally, the methods considered in ensembling are the *Bootstrap Aggregating* [Bre96], also known as *Bagging*, and the *Boosting* [FS96, CG16] techniques. Through this discussion, we evaluate the benefits of these techniques in addition to the current ensemble models.

The bagging and boosting approaches are not new in side-channel context [MPP16, PSK+18, PCP20]. While these algorithms are essentially performed with *Random Forest* (RF) [Bre01], it can also be proposed for neural networks. The details on the hyperparameters selection are provided in Appendix E Table 11 for the bagging selection and in Appendix E Table 12 for the *eXtreme Gradient Boosting* (XGBoost) [CG16] and the CNN-XGBoost [RGL+17].

The best results for all the models are reported in Table 4. In our experiment, this table illustrates that bagging and XGBoost do not provide a clear advantage when they are added to the standard proposition introduced in Table 3. However, if an improvement is observed, these algorithms can be combined with those introduced in this paper (*i.e.* Type I+II+III diversity) in order to generate a more powerful ensemble model.

Table 4: Acc_{label} for each ensemble method (Average over 10 physical traces of $1,088$ bits each)

	Bagging			XGBoost & CNN-XGBoost			
	$RMSE$	CCE_5	RkL_5	$RMSE$	CCE_5	RkL_5	$EL_{5,III}$
Acc_{label}	84.97% (±2.80%)	93.02% (±1.99%)	93.70% (±1.90%)	91.43% (±2.19%)	93.70% (±1.90%)	94.13% (±1.84%)	94.77% (±1.74%)
Training Time	$4,073s$	$315s$	$415s$	$7,597s$	$481s$	$372s$	$775s$

Remark 8. The ensembling loss cannot be considered when the bagging technique is applied. Indeed, given a profiling set $\mathcal{T}$, the N_p pairs $((\mathbf{t}_i^p, y_i^p)_{0\leq i<N_p})$ should be the same for all the committee members when the ensembling loss is computed. This condition is a limitation regarding the application of the bagging algorithm.

Combination Methods. One major issue when ensemble model is considered is to find the best way to combine the posterior probabilities of each committee member. There are several consensus methods for combining the outputs of multiple learners. We compare the two most useful combining methods:

- *Averaging* – This consensus is considered as a linear combining method. The average prediction returned by the committee members is computed. An advanced combination technique consists of weighting the average of each classifier to promote the order of the classes. However, this method stays out of our scope.
- *Voting* – This method is considered as a non-linear decision-making based on ranked information. The majority voting process predicts the value with the highest number of occurrences. Hence the collective decision has a major impact on the final prediction.

Table 5: Acc_{label} for each combination technique (Average over 10 physical traces of $1,088$ bits each)

	CCE_5	RkL_5	$EL_{5,III}$
Averaging	93.60%(±1.91%)	94.33%(±1.81%)	95.19%(±1.67%)
Voting	93.63%(±1.91%)	94.30%(±1.81%)	95.16%(±1.68%)

These results shown in Table 5 are closely correlated with those defined in the previous sections. Hence, for the experiments investigated in this paper, these aggregating functions do not impact the performance of the ensemble model.

Number of Committee Members. The number of committee members can also be considered as an issue in ensemble methods. Indeed, no useful methods define *a priori* the best number of committee members that maximize the ensemble model performance. In the following, we explore this variable in order to identify its impact on the ensembling loss performance. To that purpose, we increase the number of committee members up to 32 in order to evaluate the gain in performance and the impact on the training time. Through Table 6, we can estimate the best trade-off between the training time and the ensemble performance. For the RSA implementation, the best Acc_{bit} value is obtained for $N = 10$ committee members. While increasing the number of members seems helpful to improve the ensemble model's accuracy, in our context we seem to reach the maximal possible performance. Adding too many learners can reduce the diversity effect because some committee members can share the same errors and promote irrelevant outputs. The best number of committee members should be defined for each case-study.

Table 6: Performance evolution depending on the committee members (Average over 10 physical traces of $1,088$ bits each)

	Acc_{label}	Acc_{bit}	ϵ_{bit}	C_{NC}	C_{2^n}	C_{AA}	Training Time
$EL_{2,I+II+III}$	96.77%(±1.38%)	95.12%(±1.69%)	0.0488	301.54	54	88.45	$1,482s$
$EL_{5,I+II+III}$	97.33%(±1.26%)	96.96%(±1.34%)	0.0304	209.52	34	80.71	$3,392s$
$EL_{10,I+II+III}$	97.43%(±1.24%)	97.33%(±1.26%)	0.0267	189.23	30	80.71	$5,460s$
$EL_{16,I+II+III}$	97.90%(±1.12%)	97.10%(±1.31%)	0.0290	199.47	32	80.71	$6,942s$
$EL_{32,I+II+III}$	97.37%(±1.25%)	96.67%(±1.40%)	0.0333	225.26	37	84.03	$9,548s$

6.2 Binary Classification Problem: Attacking an ECC implementation

To emphasize the benefits of the ensembling loss, we evaluate its suitability on a classical binary classification problem. While the secure RSA dataset can be defined as a multi-class classification task (3 outputs), we perform the same experimental process on a protected ECSM (Elliptic Curve Scalar Multiplication) implementation[d] [NCOS17, Chm20] where each trace corresponds to a multiplication with a random scalar. This scenario is a 1-trace exploitation which is considered when targeting the scalar multiplication of ECDSA. Note that remaining brute-force attacks that require N_a exploitation traces, such as [SW14], cannot be used in this context.

Proposed in [NCOS17, Chm20], the ECSM secured implementation employs the Montgomery Ladder with randomized projective coordinates and a *conditional swap* (cswap) (see [NCOS17, Algorithm 1]). Starting from two (or more) curve points, the cswap countermeasure performs the scalar multiplication algorithm on one of these points depending on a mask value. Hence, if an adversary learns all the cswap condition bits from one side-channel trace, he retrieves the secret key (*i.e.* 256 bits) [NCOS17]. To be successful, the secret bits have to be recovered from a single side-channel trace. In the dataset, each trace represents a single iteration of the Montgomery Ladder scalar multiplication and the

[d] http://doi.org/10.5281/zenodo.3609789

related label corresponds to the cswap condition bit value. For deeper information on the device under test, we suggest the readers to refer to [NCOS17, Chm20].

Similarly to the secure RSA dataset, we have to add Gaussian noise $\mathcal{N} \sim \mathcal{B}\left(0, \sigma^2\right)$ to characterize the benefits of the ensembling loss. Table 7 shows the evolution of the accuracy depending on the added noise and the loss used when $20,000$ profiling traces are used. To evaluate the suitability of each network, $2,000$ validation traces are considered and the evolution of the accuracy is used to limit the overfitting/underfitting effect. For our analysis, we set the added noise to $\sigma = 30$. Once again, we clearly evaluate the benefits of the ranking loss when the added noise is high in comparison to the cross-entropy loss function. Indeed, we increase by up to 3.5% the resulted performance.

Table 7: Evolution of accuracy depending on σ ($20,000$ profiling traces & $2,000$ validation traces)

Accuracy \ σ	0	10	20	30	50	100
Categorical Cross-Entropy	99.43%	98.20%	93.63%	89.10%	83.87%	72.10%
Ranking Loss	99.47%	99.00%	96.77%	92.60%	86.15%	74.30%

First, we validate the theoretical observations provided in Section 3.1 for the binary classification problem. For that purpose, we visualize the t-SNE maps for the models trained with the different losses. Figure 4 confirms all the theoretical results introduced in this paper. The cross-entropy representation does not seem relevant to efficiently discriminate each cluster. The resulted ensemble model seems to select joint patterns such that many false positives and false negatives can deteriorate the overall performance. In opposition, the model trained with the ranking loss tends to overfit such that the false positives and false negatives can be considered as consistent. Finally, from a theoretical perspective, the ensembling loss seems the most suitable. Indeed, the data uncertainty seems to converge towards the centroid between the clusters. Indeed, the data uncertainty seems to converge towards the equidistant point of the centroid of the clusters. Hence, the resulted ensemble model tends to gather the uncertain examples towards a uniform probability distribution. Furthermore, using the ensembling loss provides a clear benefit from a diversity perspective (see Figure 5).

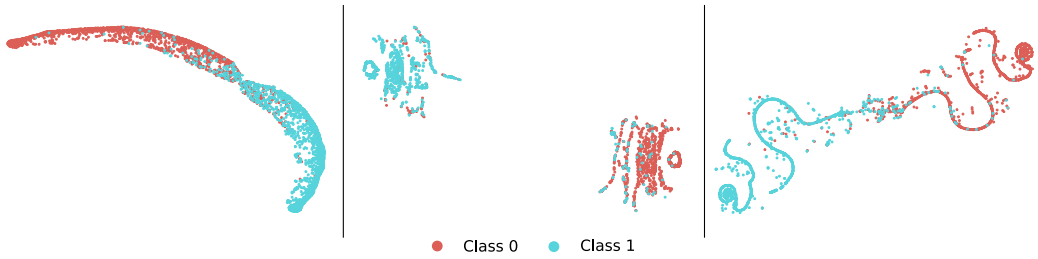

Figure 4: t-SNE embeddings. Left: Cross-Entropy Loss. Middle: Ranking Loss. Right: Ensembling Loss.

Through Table 8, we confirm the benefits of the ensembling loss for increasing the performance of the ensemble model. In comparison with the previous state-of-the-art result (*i.e.* CCE_1), the accuracy expressing the performance to retrieve the cswap bit value is increased by 6.5% when the ensembling loss is combined with the type I and II diversities. From a side-channel attack perspective, we reduce the overall number of remaining operations by $2^{58.41}$ (resp. 2^{16}) for naive attack (resp. 2^n-attack). Hence, using the ensembling loss against a binary classification problem still performs well.

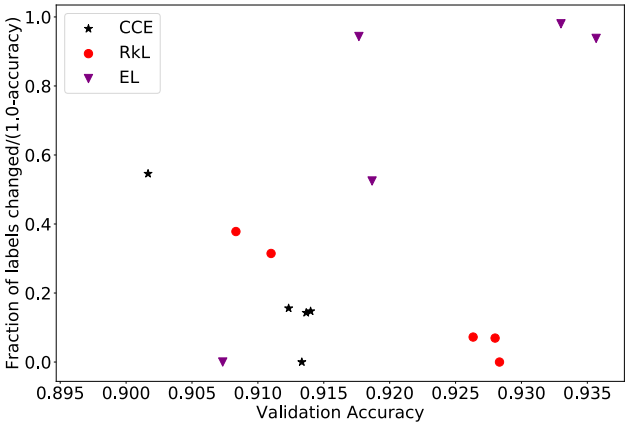

Figure 5: Diversity versus label accuracy plots for 3 ensemble models trained on Categorical Cross-Entropy (CCE), Ranking Loss (RkL) and Ensembling Loss (EL) for the binary classification.

Table 8: Performance evolution depending on the diversity applied (Average over 10 physical traces of 256 bits each). Green (resp. Red) cells are considered as practicable (resp. unpracticable) following the SOG-IS recommendations.

	Acc_{bit}	ϵ_{bit}	$\mathcal{C}_{NC}$	$\mathcal{C}_{2^n}$	Training Time
CCE_1	89.10%(±2.44%)	0.109	120.91	28	$27s$
RkL_1	92.60%(±2.05%)	0.074	90.72	19	$28s$
CCE_5	91.60%(±2.17%)	0.084	101.45	22	$440s$
RkL_5	92.63%(±2.04%)	0.0737	90.72	19	$387s$
$EL_{5,III}$	93.33%(±1.95%)	0.0667	86.98	18	$588s$
$CCE_{5,I+II}$	94.13%(±1.84%)	0.0587	79.24	16	$618s$
$RkL_{5,I+II}$	94.70%(±1.75%)	0.053	71.1	14	$750s$
$EL_{5,I+II+III}$	95.60%(±1.60%)	0.044	62.50	12	$884s$

From a naive attack perspective, an adversary using the previous state-of-the-art result (*i.e.* CCE_1) considers the ECC implementation as secure following the SOG-IS's recommendations ($C_{NC} > 100$). However, if the adversary combines all the diversity types (including the ensembling loss), he can reconsider the security of the targeted device.

Remark 9. During our experiments, we have noticed that increasing the diversity is more difficult when binary classification problems are considered in comparison to the multi-class classification problem with 3 outputs. Indeed, we had to fine tune more precisely the hyperparameters for all types of diversity. For a binary classification problem, this phenomenon can be explained by the lack of error distribution.

7 Conclusion

This paper presents a new loss, namely the *Ensembling Loss*, that increases the performance of ensemble models. Promoting the interactions between the committee members during the training process, this loss increases the resulted diversity to reduce the correlation between the errors induced by the members. First, we link this new learning metric with the mutual information between the ensemble model and its related label introduced by Brown in [Bro09]. Then, through the disagreement measure and the t-SNE visualization, we show that ensemble models trained with the *Ensembling Loss* increase the diversity between the committee members.

To assess the benefits from a side-channel perspective, we evaluate the accuracy growth on the remaining attack complexity through multiple attack scenarios. This investigation shows that applying deep learning-based side channel attacks can be inadapted to defeat secure RSA/ECC implementations if the previous state-of-the-art is considered (*i.e.* a single model trained with the cross-entropy loss function). Following the SOG-IS security guidances, the improvement provided by the combination of different types of diversity lead to a reconsideration of the targeted system's security.

Furthermore, considering the *Ensembling Loss* outperforms all the current learning metrics classically used in side-channel analysis. Hence, this loss could be considered for generating efficient ensemble models.

A future work could extend this proposition to ensemble model with diverse architectures (*Multi-Layer Perceptrons*, *Recurrent Neural Networks* [SP97, HS97], *Residual Neural Networks* [HZRS16], *U-Nets* [RFB15], etc.) and additional countermeasures for Public-Key Algorithms (e.g. address masking). Moreover, while our work mainly focuses on the gain in the attack accuracy brought by the diversity, a future work can evaluate the benefits of the ensembling loss to ease the detection of a threshold for performing a $2^n - Attack$. Finally, we can also consider its application to any broad machine learning problem that requires high accuracy.

Acknowledgement

The secure RSA implementation used in this paper is part of a challenge organized by ANSSI's laboratory of hardware security for industrial partners. It has been developed by CryptoExperts (https://www.cryptoexperts.com/) who deliberately did not include countermeasures against horizontal and address-bit attacks. The authors would also like to thank TCHES reviewers for the fruitful comments on the submitted version of this paper.

References

[AARR03] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The em side—channel(s). In Burton S. Kaliski, çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, pages 29–45, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[BGH+17] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Damien Marion, and Olivier Rioul. Optimal side-channel attacks for multivariate leakages and multiple models. *Journal of Cryptographic Engineering*, 7:331–341, 2017.

[Bre96] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, August 1996.

[Bre01] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[Bro09] Gavin Brown. An information theoretic perspective on multiple classifier systems. In Jón Atli Benediktsson, Josef Kittler, and Fabio Roli, editors, *Multiple Classifier Systems*, pages 344–353, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[C+15] François Chollet et al. Keras. https://keras.io, 2015.

[CC00] Pádraig Cunningham and John Carney. Diversity versus quality in classification ensembles based on feature selection. In Ramon López de Mántaras and Enric Plaza, editors, *Machine Learning: ECML 2000*, pages 109–116, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[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.

[CDP17] 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.

[CG16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[Chm20] Lukasz Chmielewski. Reassure (h2020 731591) ecc dataset, January 2020. Contact: chmielewski@riscure.com.

[Coh60] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[Cor99] Jean-Sébastien Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In Çetin K. Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems*, pages 292–302, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA, 1991.

[DDFP20] Gabriel Destouet, Cécile Dumas, Anne Frassati, and Valérie Perrier. Wavelet scattering transform and ensemble methods for side-channel analysis. Cryptology ePrint Archive, Report 2020/310, 2020. https://eprint.iacr.org/2020/310.

[Des77] Data encryption standard. In *FIPS PUB 46, Federal Information Processing Standards Publication*, pages 46–2, 1977.

[Die00a] Thomas Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40, 12 2000.

[Die00b] Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[Fan61] Robert M. Fano. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29(11):793–794, 1961.

[FHL19] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective, 2019.

[FLP03] Joseph L Fleiss, Bruce Levin, and Myunghee Cho Paik. *Statistical methods for rates and proportions; 3rd ed.* Wiley Series in Probability and Statistics. Wiley, Hoboken, NJ, 2003.

[FR05] G. Fumera and F. Roli. A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):942–956, 2005.

[FS96] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, page 148–156, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[GBC16] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016.

[Gir06] C. Giraud. An rsa implementation resistant to fault attacks and to simple power analysis. *IEEE Transactions on Computers*, 55(9):1116–1120, 2006.

[GMGA17] Anil Goyal, Emilie Morvant, Pascal Germain, and Massih-Reza Amini. Pac-bayesian analysis for a two-step hierarchical multiview learning approach. In Michelangelo Ceci, Jaakko Hollmén, Ljupčo Todorovski, Celine Vens, and Sašo Džeroski, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 205–221, Cham, 2017. Springer International Publishing.

[GR01] Giorgio Giacinto and Fabio Roli. Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19:699–707, 08 2001.

[HR70] M. Hellman and J. Raviv. Probability of error, equivocation, and the chernoff bound. *IEEE Transactions on Information Theory*, 16(4):368–372, 1970.

[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.

[HS90] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(10):993–1001, October 1990.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[HZRS16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015.

[KA14] Justin B. Kinney and Gurinder S. Atwal. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences*, 111(9):3354–3359, 2014.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[KL51] Salomo Kullback and Richard A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951.

[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 Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):148–179, 2019.

[KUMH17] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 971–980. Curran Associates, Inc., 2017.

[Kun04] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, USA, 2004.

[KW96] Ron Kohavi and David Wolpert. Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, page 275–283, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[KW03] Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.*, 51(2):181–207, May 2003.

[Lam00] Louisa Lam. Classifier combinations: Implementations and theoretical issues. In *Multiple Classifier Systems*, pages 77–86, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[LWC+19] L. Liu, W. Wei, K. Chow, M. Loper, E. Gursoy, S. Truex, and Y. Wu. Deep neural network ensembles against deception: Ensemble diversity, accuracy and robustness. In *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 274–282, 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.

[McG54] W. McGill. Multivariate information transmission. *Transactions of the IRE Professional Group on Information Theory*, 4(4):93–111, 1954.

[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.

[MHA14] Emilie Morvant, Amaury Habrard, and Stéphane Ayache. Majority vote of diverse classifiers for late fusion. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition - Volume 8621*, S+SSPR 2014, page 153–162, Berlin, Heidelberg, 2014. Springer-Verlag.

[MMG20] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. In *International Conference on Learning Representations*, 2020.

[MPP16] Houssem Maghrebi, Thibault Portigliatti, and Emmanuel Prouff. Breaking cryptographic implementations using deep learning techniques. In Claude Carlet, M. 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.

[NCOS17] Erick Nascimento, Łukasz Chmielewski, David Oswald, and Peter Schwabe. Attacking embedded ecc implementations through cmov side channels. In Roberto Avanzi and Howard Heys, editors, *Selected Areas in Cryptography – SAC 2016*, pages 99–119, Cham, 2017. Springer International Publishing.

[PCBP20] Guilherme Perin, Łukasz Chmielewski, Lejla Batina, and Stjepan Picek. Keep it unsupervised: Horizontal attacks meet deep learning. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(1):343–372, Dec. 2020.

[PCP20] Guilherme Perin, Łukasz 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+19] 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 Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(1):209–237, 2019.

[PSK+18] Stjepan Picek, Ioannis Petros Samiotis, Jaehun Kim, Annelie Heuser, Shivam Bhasin, and Axel Legay. On the performance of convolutional neural networks for side-channel analysis. In Anupam Chattopadhyay, Chester Rebeiro, and Yuval Yarom, editors, *Security, Privacy, and Applied Cryptography Engineering*, pages 157–176, Cham, 2018. Springer International Publishing.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[RGL+17] Xudie Ren, Haonan Guo, Shenghong Li, Shilin Wang, and Jianhua Li. A novel image classification method with cnn-xgboost model. In Christian Kraetzer, Yun-Qing Shi, Jana Dittmann, and Hyoung Joong Kim, editors, *Digital Forensics and Watermarking*, pages 378–390, Cham, 2017. Springer International Publishing.

[RSVC+11] Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and

side-channel attacks for nanoscale devices. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, pages 109–128, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[SI11] Werner Schindler and Kouichi Itoh. Exponent blinding does not always lift (partial) spa resistance to higher-level security. In Javier Lopez and Gene Tsudik, editors, *Applied Cryptography and Network Security*, pages 73–90, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[Ska96] David B. Skalak. The sources of increased accuracy for two proposed boosting algorithms. In *In Proc. American Association for Arti Intelligence, AAAI-96, Integrating Multiple Learned Models Workshop*, pages 120–125, 1996.

[SMY09] François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.

[SP97] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[SS73] P. H. A. Sneath and R. R. Sokal. *Numerical taxonomy. The principles and practice of numerical classification.* San Francisco, W.H. Freeman and Company., USA, 1973.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[SW14] Werner Schindler and Andreas Wiemers. Power attacks in the presence of exponent blinding. *Journal of Cryptographic Engineering*, 4, 06 2014.

[SW17] Werner Schindler and Andreas Wiemers. Generic power attacks on rsa with crt and exponent blinding: new results. *Journal of Cryptographic Engineering*, 7, 01 2017.

[TG96a] Kagan Tumer and Joydeep Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, 29(2):341 – 348, 1996.

[TG96b] Kagan Tumer and Joydeep Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3-4):385–404, 1996.

[Tin98] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[WJB20] Yoo-Seung Won, Dirmanto Jap, and Shivam Bhasin. Push for more: On comparison of data augmentation and smote with optimised deep learning architecture for side-channel. Cryptology ePrint Archive, Report 2020/655, 2020. https://eprint.iacr.org/2020/655.

[WPB19] Léo Weissbart, Stjepan Picek, and Lejla Batina. One trace is all it takes: Machine learning-based side-channel attack on eddsa. In Shivam Bhasin, Avi Mendelson, and Mridul Nandi, editors, *Security, Privacy, and Applied Cryptography Engineering*, pages 86–105, Cham, 2019. Springer International Publishing.

[XKS92] L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):418–435, 1992.

[Yul00] G. Udny Yule. On the association of attributes in statistics: With illustrations from the material of the childhood society. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 194:257–319, 1900.

[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.

[Zho12] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, 1st edition, 2012.

[ZL10] Zhi-Hua Zhou and Nan Li. Multi-information ensemble diversity. In Neamat El Gayar, Josef Kittler, and Fabio Roli, editors, *Multiple Classifier Systems*, pages 134–144, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[ZS19] Yuanyuan Zhou and François-Xavier Standaert. Simplified single-trace side-channel attacks on elliptic curve scalar multiplication using fully convolutional networks. 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 t-SNE Ensembling Loss

Figure 6 illustrates the evolution of the t-SNE visualizations [vdMH08] in order to evaluate the impact of the *Conditional Redundancy Loss* and the *Redundancy Loss* during the training process.

First, as mentioned in Section 3.1, the ranking loss [ZBD+20] can be formulated as the relevance loss (see Equation 7). Through its minimization, we minimize the conditional entropy $H(Z|F_n)$ which promotes the generation of three compact clusters. Figure 6 confirms this observation. The ensemble model is overconfident in the features captured during the training process. Consequently, it detects discriminative patterns to avoid connections between each cluster. However, following the t-SNE illustration, the False Positives (FP) and the False Negatives (FN) induced by the ranking loss are persistent and seem difficult to detect. Indeed, these errors are fully included in a wrong cluster. For a given number of profiling traces, a solution is to promote the interaction between the committee members in order to reduce this overconfidence and enhance the ensemble model.

In Equation 8, the *Conditional Redundancy Loss* function minimizes $(1 - \mathsf{Pr}\left[F_m|F_n, Z = z\right])$ which defines the output probability of the model F_m given F_n and Z. Hence, maximizing $\mathsf{Pr}\left[F_m|F_n, Z = z\right]$ is asymptotically equivalent to maximize $H\left(F_m|F_n, Z = z\right)$. Therefore, we force the network to generate three compact clusters given the correct label. This loss tends to increase the confidence of the network on the True Positives (TP) and the True Negatives (TN) while reducing the impact of the FP and the FN. This observation can be made on Figure 6. Indeed, adding the conditional redundancy to the ranking loss is helpful to distinguish TPs and FPs for each cluster. Hence, each cluster is divided into two parts: a part with high level of confidence in prediction and a part with uncertain predictions. This phenomenon highlights the benefits of the conditional redundancy loss function to reduce the intra-class variance and makes an easier distinction between confident and uncertain predictions. However, as illustrated in Figure 6, the conditional redundancy loss function does not clearly separate the confident and uncertain predictions into different clusters. Hence, an additional partial loss should be considered in order to increase the dissociation between these samples. This is provided by the *Redundancy Loss* function.

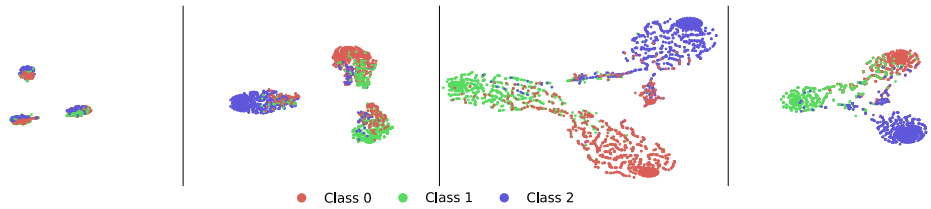

Figure 6: t-SNE embeddings. First: Ranking Loss. Second: Ranking Loss + Conditional Redundancy Loss. Third: Ranking Loss + Redundancy Loss. Fourth: Ensembling Loss (= Ranking Loss + Conditional Redundancy Loss + Redundancy Loss).

In Equation 9, the *Redundancy Loss* function minimizes $\mathsf{Pr}\left[F_m|F_n\right]$ which defines the output probability of the model F_m given F_n. From an information theory perspective, this can be considered as a minimization of $H\left(F_m|F_n\right)$ In other words, we want to maximize the inter-class variance between the models F_m and F_n. Hence, adding the redundancy loss to the ranking loss should increase the distance between each cluster by diversifying the features representation of each cluster. This observation can be validated thanks to Figure 6. Indeed, the third t-SNE visualization illustrates a model trained with the ranking and the redundancy losses. In comparison with the first t-SNE visualization, we can highlight the benefits of the redundancy loss to increase the distance between each cluster and make the FN and FP less persistent. However, in some extent, this approach

generates sparse representation of a given cluster and also reduces the confidence of the networks on some TP. Hence a good trade-off has to be found between maximizing the confidence of the TP (*i.e.* conditional redundancy loss) and minimizing the persistence of the FP (*i.e.* redundancy loss). The *Ensembling Loss* aims at finding this solution for a given α, β, γ, μ values (see Equation 10).

In Figure 6, the combination of the relevance loss, the conditional redundancy loss and the redundancy loss creates three separate clusters. When the network is confident in its predictions, it will assign the related examples to the correct cluster. Thanks to the conditional redundancy loss, we know that the predictions with high level of confidence will be assigned to the same compact cluster. However, the ensembling loss also creates some connections between the clusters which seem defined by the data uncertainty. This result tends to reduce the number of consistent FP and FN such that few errors can be detected on each cluster in contrast with the ranking loss. This observation highlights the benefits of the redundancy loss during the training process. In Figure 6, the ensembling loss find a good trade-off between maximizing the confidence of the TP and minimizing the persistence of the FP.

B Diversity Measures

Let $N^{ab}_{n,m}$ be the joint counts between two learners F_n and F_m. We denote $a = 0$ (resp. $b = 0$) if F_n (resp. F_m) wrongly predicts a value and $a = 1$ (resp. $b = 1$) otherwise. For example, $N^{01}_{n,m}$ defines the number of elements such that F_n obtains an incorrect value for a given input while F_m correctly predict the related class.

Definition 7 (κ-statistic [Coh60]). The κ-statistic measures the diversity between two classifiers F_n, F_m as follows :

$$\kappa(F_n, F_m) = \frac{2\left(N^{11}_{n,m}N^{00}_{n,m} - N^{01}_{n,m}N^{10}_{n,m}\right)}{\left(N^{11}_{n,m} + N^{10}_{n,m}\right)\left(N^{01}_{n,m} + N^{00}_{n,m}\right) + \left(N^{11}_{n,m} + N^{01}_{n,m}\right)\left(N^{10}_{n,m} + N^{00}_{n,m}\right)}. \quad (16)$$

Figure 7 illustrates the evolution of the diversity depending on the loss used. Indeed, the ensembling loss reduces the overall κ-statistic measure in comparison with the cross-entropy or the ranking loss. This figure confirms that the ensembling loss decorrelates the errors between the committee members. Moreover, combining different types of diversity is helpful to improve this effect (see Figure 3). These observation are in agreement with those introduced in Section 3.2.

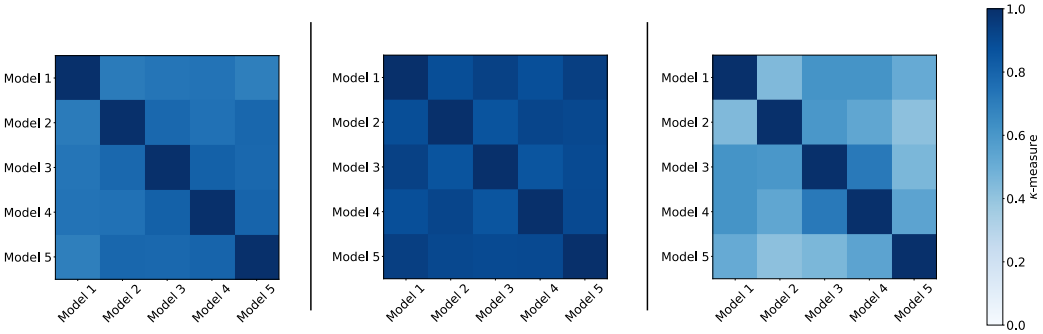

Figure 7: κ-statistic. Left: Cross-Entropy Loss. Middle: Ranking Loss. Right: Ensembling Loss.

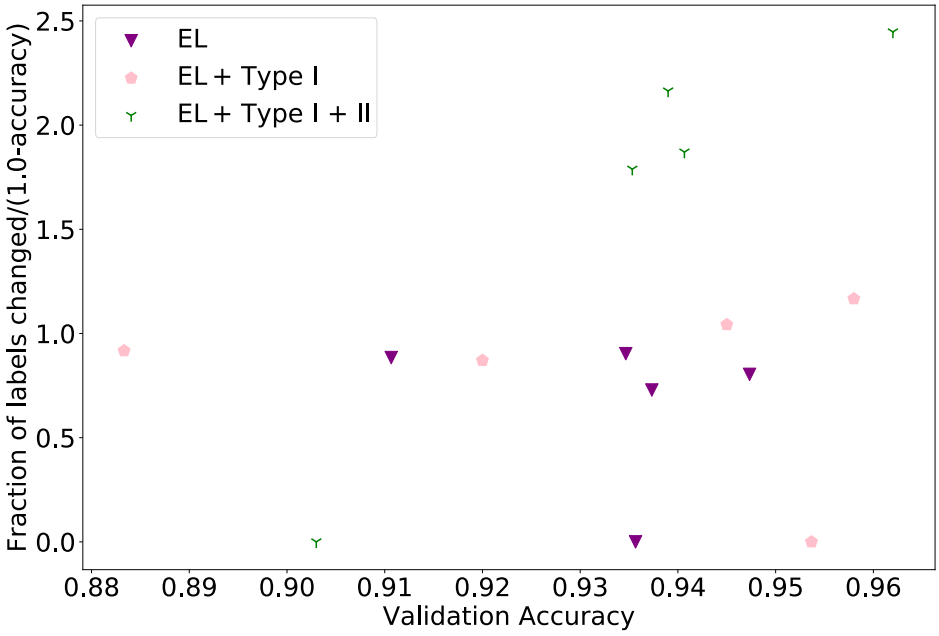

Figure 8: Diversity versus label accuracy plots for ensemble models trained on Ensembling Loss (EL), Ensembling Loss (EL) + Type I diversity and Ensembling Loss + Type I + II diversities.

C Alternate Attack on RSA without CRT

Introduced in [SW14], the *Alternate Attack* (AA) targets RSA modular exponentiation protected with exponent blinding. Based on the *Basic Attack* and the *Enhanced Attack* [SI11], the alternate attack retrieves the secret exponent bits from multiple traces. This attack can be extended to the Elliptic Curves [SW14] and RSA with CRT [SW17]. However, some tricks are specific to each case study. In this paper, we only focus on the application of the alternate attack on RSA without CRT. In particular, we formulate a complexity equation for this alternate attack that is missing from the original paper.

In [SW14], Schindler and Wiemers define the blinded exponent d' with a blinding scalar r' as:

$$d' = d + r' \cdot \phi(N), \tag{17}$$

where d is the secret exponent and $\phi(N)$ defines the Euler totient function of the modulus N.

Algorithm [SW14]. In the alternative attack scenario against RSA without CRT, it is assumed that the attacker knows the upper halves of the binary representation of $\phi(N)$ because it is similar to N. Let K be the bit-length of the secret exponent and $d >> s = \lfloor \frac{d}{2^s} \rfloor$ defines the bits of d shifted to the right by s places. If $s \geq \frac{K}{2} + R + 6$, then $\lfloor \frac{d}{2^s} \rfloor$ depends on the upper half of the bits of $\phi(N)$. Given a secret blinding exponent d', Schindler and Wiemers introduce $\alpha = \left\lfloor \frac{d'}{2^{k-1}} \right\rfloor$ and β such that $0 \leq \beta < 2^{K-1} < \phi(N)$ to rewrite d' in such a way that the $(R+1)$ most significant bits influence α while the $(K-1)$ least significant bits influence β. Then, the authors define $(d' >> s)$ as:

$$(d' >> s) = \left\lfloor \frac{d + r' \cdot \phi(N)}{2^s} \right\rfloor = \left\lfloor \frac{\alpha 2^{K-1} + \beta}{2^s} \right\rfloor,$$

and,

$$(d >> s) = \left\lfloor \frac{\alpha 2^{K-1} (\mathrm{mod} \phi(N)) + \beta - \omega\phi(N)}{2^s} \right\rfloor = \left\lfloor \frac{\alpha 2^{K-1} (\mathrm{mod} N) + \beta - \omega N}{2^s} \right\rfloor,$$

with high probability for an unknown $\omega \in \{0, 1\}$ and $s \geq \frac{K}{2} + R + 6$.

When an adversary captures the leakage traces, he guesses the randomized exponent to obtain an estimation $\hat{d}'$ of the true blinded exponent d':

$$\left\lfloor \frac{\hat{d}'}{2^s} \right\rfloor = \left\lfloor \frac{d' \oplus e}{2^s} \right\rfloor = \left\lfloor \frac{\hat{\alpha} 2^{K-1} + \hat{\beta}}{2^s} \right\rfloor,$$

where e expresses the guessing error induced by exponent $\hat{d}'$, '$\oplus$' denotes the bitwise XOR operation, $\hat{\alpha}$ (resp. $\hat{\beta}$, $\hat{\omega}$) is an estimation of α (resp. β, ω).

Given an error rate ϵ_{bit}, an adversary can estimate the number of erroneous bits in $\hat{\alpha}$. The idea of the alternative attack against RSA without CRT is to generate all candidates for α (denoted $\hat{\alpha}_c$) and compute the resulted blinding factor candidates as $\hat{r'}_c = (\hat{d}' - \hat{d})/N = \lfloor \hat{\alpha}_c 2^{K-1}/N \rfloor + \omega$ with $\omega \in \{0, 1\}$. Then, for each candidate $\hat{\alpha}_c$ and $\hat{r'}_c$, the adversary can compute an estimation of the resulted error $\hat{e}$ based on a guessed on the secret exponent d such that:

$$\hat{e} = \left(\hat{r'}_c N + \left\lfloor \frac{\hat{d}}{2^s} \right\rfloor 2^s \right) \oplus \left(\hat{\alpha}_c 2^{K-1} + \hat{\beta} \right).$$

If $\left\lfloor \frac{\hat{d}}{2^s} \right\rfloor = \lfloor \frac{d}{2^s} \rfloor$, a blinding factor estimation $\hat{r'}_c$ is defined as a candidate for r' if $HW(\lfloor \hat{e}/2^s \rfloor) \leq t_0$ with t_0 a threshold configured by the attacker. A smaller t_0 value induces a more restrictive candidate selection. The threshold t_0 should be selected such that no false candidates for r' are kept. More details on the alternative attack algorithm are provided in [SW14, Algorithm 4]. However, it is acceptable that some of the $\left\lfloor \hat{d}/2^s \right\rfloor$ candidates are wrongly guessed. Then, to retrieve the remaining bits of $\phi(N)$, the adversary has to perform the **Step 3** of the *Enhanced Attack* introduced by Schindler and Itoh [SI11]. Of course, for a number N_a of attack traces, we expect $q_{n_0,t_0} N$ candidates for $\lfloor d'/2^s \rfloor$ where,

$$q_{n_0,t_0} = \left(\sum_{i \leq n_0} \binom{R+1}{i} \epsilon_{bit}^i (1 - \epsilon_{bit})^{R+1-i} \right) \cdot \left(\sum_{i \leq t_0} \binom{K-1-s}{i} \epsilon_{bit}^i (1 - \epsilon_{bit})^{(K-1-s)-i} \right),$$

such that, the two brackets quantify the probabilities that $\hat{\alpha}$ and the relevant bits of $\hat{\beta}$ contain at most n_0 or t_0 guessing errors, respectively [SW17]. Through all these components, we can estimate the complexity of the resulted alternate attack for a given s and t_0 values.

Complexity. First, the adversary has to configure the s, t_0 and N_a values to perform successful attacks. Then, for a given $\hat{d}' = \hat{\alpha} 2^{K-1} + \hat{\beta}$, the adversary has to generate all $\hat{\alpha}_c$ candidates that differ by n_0 bits from $\hat{\alpha}$ at most. Hence, their is $M_0 = \sum_{i \leq n_0} \binom{R+1}{i}$ candidates for α.

The computation of each candidate $\hat{r'}_c$ and $\hat{e}_c$ depends on the number of $\hat{\alpha}_c$ elements. Therefore, their is $2 \cdot M_0$ candidates for r' and $2 \cdot M_0 \cdot 2^{K-s}$ candidates for e in the worst case (*i.e.* if $\left\lfloor \hat{d}/2^s \right\rfloor \neq \lfloor d/2^s \rfloor$) and $2 \cdot M_0$ candidates for e otherwise (*i.e.* if $\left\lfloor \hat{d}/2^s \right\rfloor = \lfloor d/2^s \rfloor$). In the following, we only consider the worst case scenario for the complexity estimation.

Given a secret exponent d of K bit-length, a blinding scalar of bit-length R, an error rate ϵ_{bit} and a number of attack traces N_a, the Alternate Attack Complexity $\mathcal{C}_{AA}$ is defined as:

$$\mathcal{C}_{AA}\left(K, R, \epsilon_{bit}, N_a\right) = \log_2\left(N_a \cdot 2^{K-s+1} \cdot \sum_{i=0}^{\lceil (R+1)\times\epsilon_{bit}\rceil} \binom{R+1}{i}\right), \tag{18}$$

with t_0 configured such that no false candidates for r' are selected.

Remark 10. To consider the *Alternate Attack* has a success, the adversary has to define the number of attack traces N_a that are needed for recovering the entire bits of $\phi(N)$. Hence, to correctly estimate $\mathcal{C}_{AA}$, the adversary has to perform the **Step 3** of the *Enhanced Attack* [SI11] in order to find a correct assumption about N_a.

D Architectures for All Types of Diversity

The architectures used for the type I diversity are randomly selected such that the number of convolutional layers (with BatchNormalization (BN) layer [IS15]) and fully-connected layers (FC) do not exceed 2. Hence, we evaluate the type I diversity with the restriction of small network complexity. We select 5 architectures with high individual Acc_{label} value (*i.e.* $\geq 85\%$) to limit the impact of the outliers and preserve an overall good performance. All the architectures used for the type I diversity investigation are details in Table 9.

Table 9: Architectures and performance related to the networks used for the type I diversity (models trained with the ranking loss)

Type I diversity	$Model_1$	$Model_2$	$Model_3$	$Model_4$	$Model_5$
1^{st} **Conv. layer (+ BN)**	2 filters (size 1)	10 filters (size 5)	5 filters (size 15)	2 filters (size 1)	2 filters (size 1)
1^{st} **Pool. layer**	Avg (stride 2)	Avg (stride 5)	Max (stride 5)	Avg (stride 2)	Max (stride 2)
2^{nd} **Conv. layer (+ BN)**	-	-	-	2 filters of size 25	-
2^{nd} **Pool. layer**	-	-	-	Avg (stride 2)	-
Flatten	Yes	Yes	Yes	Yes	Yes
1^{st} **FC layer**	-	2 nodes	-	-	5 nodes
2^{nd} **FC layer**	-	-	-	-	5 nodes
Prediction layer	3 classes	3 classes	3 classes	3 classes	3 classes
Acc_{label}	94.03%	92.50%	93.80%	94.33%	89.87%
Acc_{bit}	90.62%	89.24%	90.62%	91.73%	86.66%
$\mathcal{C}_{NC}$	483.93	531.28	483.93	440.07	611.61
Training Time	$400s$	$140s$	$100s$	$60s$	$80s$

For the type I + II diversity study, we randomly generate 100 models from a range of hyperparameter selection introduced in Table 2. From the resulted pool of classifiers, we pick out those with a high individual performance (*i.e.* $Acc_{label} \geq 85\%$) such that their pairwise diversity measure (*i.e.* disagreement measure or κ-statistic) is maximized. The resulted architectures are details in Table 10.

Table 10: Architectures and performance related to the networks used for the type I + II diversity (models trained with the ranking loss)

Type I + II diversity	$Model_1$	$Model_2$	$Model_3$	$Model_4$	$Model_5$
1^{st} **Conv. layer (+ BN)**	16 filters (size 11)	4 filters (size 5)	16 filters (size 11)	32 filters (size 21)	32 filters (size 1)
1^{st} **Pool. layer**	Max (stride 4)	Avg (stride 2)	Avg (stride 6)	Avg (stride 2)	Avg (stride 2)
2^{nd} **Conv. layer (+ BN)**	64 filters (size 1)	16 filters (size 21)	32 filters (size 11)	-	-
2^{nd} **Pool. layer**	Avg (stride 6)	Avg (stride 4)	Avg (stride 2)	-	-
3^{rd} **Conv. layer (+ BN)**	-	8 filters (size 5)	64 filters (size 43)	-	-
3^{rd} **Pool. layer**	-	Max (stride 2)	Max (stride 6)	-	-
4^{th} **Conv. layer (+ BN)**	-	-	32 filters (size 11)	-	-
4^{th} **Pool. layer**	-	-	Max (stride 6)	-	-
5^{th} **Conv. layer (+ BN)**	-	-	16 filters (size 21)	-	-
5^{th} **Pool. layer**	-	-	Avg (stride 4)	-	-
Flatten	Yes	Yes	Yes	Yes	Yes
1^{st} **FC layer**	8 nodes	-	-	32 nodes	-
2^{nd} **FC layer**	8 nodes	-	-	-	-
3^{rd} **FC layer**	16 nodes	-	-	-	-
Prediction layer	3 classes	3 classes	3 classes	3 classes	3 classes
Acc_{label}	90.04%	95.43%	94.53%	92.83%	93.83%
Acc_{bit}	83.16%	93.84%	93.01%	90.16%	89.79%
$\mathcal{C}_{NC}$	704.45	358.84	393.24	500.08	512.73
Training Time	$940s$	$200s$	$400s$	$700s$	$540s$

E Bagging and Boosting Hyperparameters Selection

The hyperparameter selection is performed on *Random Forest* (RF) [Bre01] and *Convolutional Neural Networks* (CNN). Table 11 (resp. Table 12) identifies the ranges selected to configure the bagging (resp. boosting) models. Let N_{split} be the minimum number of samples required to split an internal node and the boostraping factor r denotes the number of side-channel traces n used to train a classifier (*i.e.* $n = r \cdot N_p$ with $N_p = 30,000$). For the Random Forest models, the nodes are expanded until all leaves contain less than N_{split} samples

Table 11: Range of hyperparameters selection for Bagging models

	Variables	**Values**
RF	**Bootstraping factor** (r)	$\{0.5, 0.8, 1.0\}$
	Objective function	Root Mean Square Error (RMSE)
	Number of trees	$\{5, 10, 25, 50, 100, 500, 1,000\}$
	N_{split}	$\{2, 3, 5, 10\}$
	Depth	until all leaves contain less than N_{split} samples
CNN	**Bootstraping factor** (r)	$\{0.5, 0.8, 1.0\}$
	Loss function	$\{CCE, RkL\}$
	Number of models	$\{1, 2, 3, 4, 5\}$
	Architecture	same as Section 4.1

Table 12: Range of hyperparameters selection for XGBoost models

	Variables	Values
RF	**Objective function**	Root Mean Square Error (RMSE)
	Number of trees	$\{5, 10, 25, 50, 100, 500, 1,000\}$
	Learning rate	$\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$
	Depth	until all leaves contain less than N_{split} samples
CNN-XGB	**Number of trees**	$\{5, 10, 25, 50, 100, 500, 1,000\}$
	Number of CNN	$\{1\}$
	Architecture	same as Section 4.1
	Learning rate	$\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$