Perceived Information Revisited II

Information-Theoretical Analysis of Deep-Learning Based Side-Channel Attacks

Akira Ito[1], Rei Ueno[2] and Naofumi Homma[2]

[1] NTT Social Informatics Laboratories, Nippon Telegraph and Telephone Corporation,
3–9–11 Midori-cho, Musashino-shi, Tokyo, 180-8585, Japan
akria.itoh@ntt.com

[2] Tohoku University, 2–1–1 Katahira, Aoba-ku, Sendai-shi, Miyagi, 980-8577, Japan
rei.ueno.a8@tohoku.ac.jp, naofumi.homma.c8@tohoku.ac.jp

Abstract. In conventional deep-learning-based side-channel attacks (DL-SCAs), an attacker trains a model by updating parameters to minimize the negative log-likelihood (NLL) loss function. Although a decrease in NLL improves DL-SCA performance, the reasons for this improvement remain unclear because of the lack of a formal analysis. To address this open problem, this paper explores the relationship between NLL and the attack success rate (SR) and conducts an information-theoretical analysis of DL-SCAs with an NLL loss function to solve open problems in DL-SCA. To this end, we introduce a communication channel for DL-SCAs and derive an inequality that links model outputs to the SR. Our inequality states that mutual information between the model output and intermediate value, which is named the latent perceived information (LPI), provides an upper bound of the SR of a DL-SCA with a trained neural network. Subsequently, we examine the conjecture by Ito et al. on the relationship between the effective perceived information (EPI) and SR and clarify its valid conditions from the perspective of LPI. Our analysis results reveal that a decrease in NLL correlated with an increase in LPI, which indicates that the model capability to extract intermediate value information from traces is enhanced. In addition, the results indicate that the LPI bounds the SR from above, and a higher upper bound of the SR could directly improve the SR if the selection function satisfies certain conditions, such as bijectivity. Finally, these theoretical insights are validated through attack experiments on neural network models applied to AES software and hardware implementations with masking countermeasures.

Keywords: Profiled side-channel attacks · Perceived information · Success Rate · Deep learning · Information theory

1 Introduction

1.1 Background

Deep-learning-based side-channel attack. Deep-learning-based side-channel attacks (DL-SCAs) have been an active research topic in the field of cryptographic implementation [MHM14, MPP16, CDP17, PHJ+19a, HHGG20, RWPP21, UXT+22, PPM+23, TUX+23, SM23] because of their effectiveness. Profiled DL-SCA can achieve high performance in a key-recovery SCA on AES as well as in side-channel assisted chosen-ciphertext attacks on post-quantum key encapsulation mechanisms, even if the implementation is protected using masking and random delay [ZBHV19,WAGP20,ZBHV21,LZC+21,UXT+22,TUX+23]. An attacker using a DL-SCA requires less assumption/prior knowledge about the target implementation compared to who uses other SCAs. For example, conventional SCAs, such as correlation power analysis and template attacks [CRR02], require modifications to the attack algorithm and preprocessing of traces based on countermeasures employed in the target implementation. Conversely, in DL-SCA, a neural network (NN) model effectively counteracts these measures by utilizing a sufficient number of traces during the NN training. A thorough investigation of the theory and practice of DL-SCAs is imperative to comprehend potential threats posed by SCAs to cryptographic implementations.

Performance metrics of DL and SCA. In a DL-SCA, selecting an objective function, referred to as a loss function, is essential because it directly affects the training efficacy of the model (i.e., its attack performance). Performance metrics considered for SCAs, such as success rate (SR), are not always consistent with those used in machine learning, such as accuracy, recall, and F1 score. Picek et al. demonstrated that SCAs can succeed even when models exhibit 0% accuracy [PHJ+19b]. In multi-class classification problems in machine learning, *negative log-likelihood (NLL)*, often referred to as *categorical cross-entropy* in deep learning contexts, is frequently utilized as a loss function because it is considered as a surrogate loss for enhancing accuracy [MMZ23]. However, conventional machine learning measures such as accuracy do not consistently evaluate performance in SCAs; therefore, it remains unclear if NLL is adequate as a loss function in SCAs.

Loss functions for DL-SCAs. To address this discrepancy, specialized loss functions for DL-SCAs were developed in several studies. These include a *cross-entropy ratio (CER) loss* function, which makes the NLL of the correct key smaller than the average NLL of other key candidates [ZZN+20], and a *ranking loss* function, which employs ranking learning to train the model and increase the rank of the correct key [ZBD+20]. Previous studies reported that these functions perform better than NLL, particularly in attack scenarios where imbalanced data labeled with Hamming weight (HW) or Hamming distance (HD) are used, and/or where the number of traces available for training is very limited. However, as reported in [ISUH21, ZBD+20], models trained with NLL also achieve comparable or superior performance compared to those trained with specialized loss functions when a correction term related to a binomial distribution is incorporated in the model output or when sufficient training traces can be acquired. While the NLL was not originally designed for SCAs, there is no clarity regarding why they are effective in DL-SCA contexts. Numerous studies that focus on DL-SCAs employ NLL for model training; yet, a theoretical analysis of the relationship between a decrease in NLL and SCA performance remains an important open research question.

Mutual information (MI), perceived information (PI), and success rate (SR). Masure et al. presented the pioneering theoretical analysis of model training using NLL in DL-SCA [MDP20]. They proved that the NLL asymptotically converges to a cross-entropy (CE) function as the number of traces utilized approaches infinity. Furthermore, they demonstrated that *perceived information (PI)* can be derived from CE. Given a model, the PI is a lower bound on the mutual information (MI) $I(Z; \boldsymbol{X})$ between a side channel trace $\boldsymbol{X}$ and an intermediate value Z. MI can be estimated using PI by identifying model parameters that minimize CE. Besides, de Chérisey et al. [dCGRP19] presented an inequality that upper-bounds the SR by MI, thereby making MI crucial for evaluating the achievable SR. Further, when the PI equals the MI, the distribution modeled by the NN is equal to the true distribution, thereby yielding an optimal distinguisher for the most potent attack [HRG14]. Note also that the computation and precise estimation of MI is usually quite difficult; this is a reason why, in practice, we require an alternative metric, which is computable and is theoretically valid for the evaluation of SCA.

Conjecture on PI–SR inequality. In [MDP20], Masure et al. revealed the significance of identifying parameters that minimized NLL, which is approximately equivalent to CE in SCAs. However, the NLL loss function cannot attain its minimum value during model training. In fact, parameters that minimize the NLL may not exist depending on the model architecture, which makes it crucial to understand the relationship between the SR and non-minimized NLLs from both theoretical and practical perspectives. Given this context, Masure et al. [MDP20] conjectured that an inequality similar to the one proposed by de Chérisey et al. may exist between the PI and SR; that is, the PI is an upper bound of the SR for a given model. If this inequality holds, the upper bound of the SR for the model can be evaluated from the PI (or equivalently, NLL), thereby allowing the rapid prediction of model performance during training without resorting to computationally intensive SR estimation through key recovery. Therefore, validating this inequality has significant practical implications.

Effective CE/PI (ECE/EPI) and conjecture on EPI–SR inequality. Ito et al. [IUH22b] constructed a counterexample that disproved the conjecture by Masure et al. [MDP20]. They demonstrated that the SR remained invariant to variations in the inverse temperature β of the softmax function in the model output, whereas CE/NLL changed. Their findings indicated that an increase in β results in an unbounded increase in the CE without changing the SR; therefore, a high NLL, which is approximately equal to the CE, does not inherently signify an unsuccessful attack. Further, they introduced *effective CE (ECE)* as the CE minimized with respect to β and defined *effective PI (EPI)* using ECE. ECE and EPI were defined to solve the uncertainty of SR. Accordingly, Ito et al. conjectured a similar inequality that upper-bounds SR by EPI and demonstrated its experimental validity. However, this conjecture is yet to be theoretically substantiated. Proving this inequality would not only explain how NLL loss reduction enhances DL-SCA performance but also theoretically validate the estimation of SR from NLL loss. Further, the theoretical foundations would aid the study on DL-SCA in practical aspects to improve its performance and develop countermeasures. Therefore, clarifying the relationship among PI, EPI, and SR is non-trivial in this research domain.

1.2 Our contributions

In this paper, we establish a new theoretical foundation for DL-SCA by exploring the relationship between the SR and NLL in DL-SCA. To this end, we introduce a new communication channel model for DL-SCA and derive an inequality that relates the model outputs to the SR. Subsequently, we proposed a new metric, named *latent perceived information (LPI)*, according to this inequality. Further, we identified conditions under which the EPI–SR inequality conjecture holds from an LPI perspective. In addition, we formally analyzed how training a model with NLL helps improve the SR. This study makes three contributions, as presented below.

Information-theoretical analysis of DL-SCA. Although previous studies examined DL-SCA from an information-theoretic perspective, DL-SCA was not modeled as a communication channel. In this study, we establish a new communication channel model for DL-SCA. Based on this communication channel, we introduce LPI as a new metric theoretically connected to SR, which plays an essential role in our analysis. Further, we derive an information-theoretic inequality that links the model outputs to the SR. Namely, we formally prove that LPI is an upper bound of SR, as represented by the *LPI–SR inequality*.

Revealing conditions when EPI–SR inequality holds. As the LPI–SR inequality is formally proven, the EPI–SR inequality is also valid if the LPI and EPI are equivalent. We identify conditions under which the EPI–SR inequality holds using the relationship between the LPI and EPI. In addition, we demonstrate that the conjecture of the EPI–SR inequality holds when the distribution modeled by an NN is well-calibrated with the inverse temperature and matches the true distribution of the intermediate value given the model output. We further demonstrate that this condition is satisfied in practice through experimental attacks on AES software and hardware implementations with masking countermeasures. Our theorems and experiments would also validate the use of EPI for SR evaluation through the inequality.

Revealing why the NLL loss function enhances the performance of DL-SCA. Finally, we theoretically explain why training a model with NLL positively affects the SR of SCAs. To this end, we show that a decrease in NLL yields an approximate increase in both the EPI and LPI. Our inequality suggests that the upper bound on SR increases if LPI increases. If the selection function satisfies certain conditions such as bijectivity, then this increase in the upper bound of the SR will boost the actual SR directly, thereby enhancing the effectiveness of the attack.

Remark 1. Some of our findings (e.g., an NLL decrease improves the DL-SCA performance) may have been *experimentally* confirmed so far. Our contribution includes establishing a *theoretical* foundation of DL-SCA, which has rarely been discussed formally.

Remark 2. The paper mainly focuses on *profiled* DL-SCA, while our arguments would be applicable to non-profiled *collision DL-SCAs* [SM23, IUTH23].

1.3 Paper organization

Section 2 introduces the mathematical notations used in this paper and the DL-SCA. Section 3 presents the information-theoretical analyses on DL-SCA. Section 4 presents the relationship among PI, EPI, and LPI and shows the condition when the EPI–SR inequality holds. Section 5 formally demonstrates why the NLL loss is adequate for DL-SCA. Section 6 demonstrates the experimental DL-SCAs on masked AES implementations to verify the validity of our theoretical analysis. Finally, Section 7 concludes the paper.

2 Preliminaries

2.1 Notations

A calligraphic letter (e.g., $\mathcal{X}$) represents a set, a lowercase variable (e.g., x) represents an element of the corresponding set (i.e., $x \in \mathcal{X}$), and an uppercase variable (e.g., X) represents a random variable over the corresponding set (i.e., X for $\mathcal{X}$), unless otherwise defined. Let $\Pr(A)$, p, and q represent the probability of an event A, true density or mass function, and probability density or mass function represented by an NN, respectively. For example, the true probability mass function of discrete random variables X and Y is given by $p_{X,Y}(x, y) = \Pr(X = x, Y = y)$. We may omit the subscripted random variables if the random variables of the probability distribution are obvious. For example, we may write $p(x, y)$ for $p_{X,Y}(x, y)$. The conditional probability distribution is denoted by $p_{X|Y}(x \mid y) = p(x \mid y) = p(x, y)/p(y)$ if $p(y) \neq 0$; otherwise, $p_{X|Y}(x \mid y) = 0$. Let $\mathbb{E}$ represent the expectation. For example, $\mathbb{E}_X f(X)$ represents the expectation of $f(X)$ in terms of X, where $f : \mathcal{X} \to \mathbb{R}$ represents a (measurable) function. Let $H(X) = -\mathbb{E}_X \log p_X(X)$ represent an entropy function, where log is the binary logarithm. The MI between X and Y is defined as $I(X; Y) = H(X) - H(X \mid Y) = \mathbb{E}_{X,Y} \log p_{X,Y}(X, Y)/(p_X(X)p_Y(Y))$. A sequence of m random variables/vectors independently sampled from a distribution of X is represented by independent copies as $X^m = (X_1, X_2, \ldots, X_m)$.

2.2 Overview of profiled DL-SCA

This study focuses on SCAs on block ciphers, particularly AES. The DL-SCA consists of profiling and attack phases. During the profiling phase, an NN is trained to model the conditional distribution corresponding to the device leakage characteristics. Let $\mathcal{S}_p = \{\, (\boldsymbol{X}_i, Z_i) \mid 1 \leq i \leq m_{\mathsf{pro}} \,\}$ be a training dataset used in the profiling phase, where $\boldsymbol{X}_i$, Z_i, and $m_{\mathsf{pro}} \in \mathbb{N}$ represent the i-th side-channel trace of the i-th observation, corresponding intermediate value (e.g., first-round Sbox output in typical SCAs on software AES implementation), and the number of traces used in the profiling phase, respectively. We assume that $\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_{m_{\mathsf{pro}}}$ and $Z_1, Z_2, \dots, Z_{m_{\mathsf{pro}}}$ are independent and identically distributed (i.i.d.) random variables over $\mathcal{X} = \mathbb{R}^{n_\ell}$ and $\mathcal{T} = \{0,1\}^n$, respectively. Here, $n_\ell \in \mathbb{N}$ is the number of sample points in a trace, and $n \in \mathbb{N}$ is the bit length of the intermediate value (in the case of AES, $n = 8$). Let $\theta \in \mathbb{R}^{n_\theta}$ represent the NN parameters, where n_θ denotes the dimension of the parameters. The profiling phase estimates adequate model parameters $\hat{\theta}$ using the training dataset $\mathcal{S}_p$. Optimal parameters are obtained by solving the minimization problem of the CE loss function, which is defined as

$$\mathrm{CE}(q_\theta) = -\mathbb{E}_{Z,\boldsymbol{X}} \log q_\theta(Z \mid \boldsymbol{X}) = -\int \sum_z p_{Z,\boldsymbol{X}}(z, \boldsymbol{x}) \log q_\theta(z \mid \boldsymbol{x})\, d\boldsymbol{x}, \tag{1}$$

where Z and $\boldsymbol{X}$ are the random variables of label z and trace $\boldsymbol{x}$, respectively, and q_θ represents the conditional probability distribution modeled by the NN with parameters θ.

In Equation (1), $\mathrm{CE}(q_\theta)$ takes the minimum value if and only if $p_{Z|\boldsymbol{X}} = q_\theta$ [Bis06, GBC16]. We can model the true distribution p if we can determine the optimal parameters $\hat{\theta}$ that make $\mathrm{CE}(q_{\hat{\theta}})$ sufficiently small; however, we cannot calculate Equation (1) because it contains the integral and summation of the unknown probability distribution p. Therefore, we approximate $\mathrm{CE}(q_\theta)$ using the training data $\mathcal{S}_p$ via the Monte Carlo method as

$$\mathrm{CE}(q_\theta) \approx L(q_\theta) = -\frac{1}{m_{\mathsf{pro}}} \sum_{i=1}^{m_{\mathsf{pro}}} \log q_\theta(Z_i \mid \boldsymbol{X}_i). \tag{2}$$

The approximation of CE in Equation (2) is called the NLL. According to the law of large numbers, the NLL converges almost surely to $\mathrm{CE}(q_\theta)$ as $m \to \infty$ for fixed q_θ.

In the attack phase, we estimate the secret key k^* of the target device by using the trained model. Let $\mathcal{S}_a = \{\, (\boldsymbol{X}_i, T_i) \mid 1 \leq i \leq m_{\mathsf{atk}} \,\}$ be the dataset used in the attack phase, where $\boldsymbol{X}_i$, T_i, and $m_{\mathsf{atk}} \in \mathbb{N}$ represent the side-channel trace at the i-th observation, and corresponding plaintext or ciphertext, and the number of attack traces, respectively. During the attack phase, we calculate the NLL for each hypothetical key candidate $k \in \mathcal{K}$ using the intermediate value $Z_i^{(k)}$ calculated from T_i as

$$L^{(k)}(q_{\hat{\theta}}) = -\frac{1}{m_{\mathsf{atk}}} \sum_{i=1}^{m_{\mathsf{atk}}} \log q_{\hat{\theta}}(Z_i^{(k)} \mid \boldsymbol{X}_i).$$

Here, $Z_i^{(k)}$ is given as the output of selection function (e.g., $Z_i^{(k)} = \mathrm{Sbox}(T_i \oplus k)$). The correct key is estimated as the key candidate with the lowest NLL value, which is equivalent to approximating and comparing the results.

$$\mathrm{CE}^{(k)}(q_{\hat{\theta}}) = -\mathbb{E} \log q_{\hat{\theta}}(Z^{(k)} \mid \boldsymbol{X}),$$

for each key candidate k. It has been proven that such a DL-SCA is optimal (i.e., it maximizes the SR) if we have $q_{\hat{\theta}} = p_{Z|\boldsymbol{X}}$ [IUH21, IUH22b].

Hereafter, we simply denote the number of traces by m instead of m_{atk} and m_{pro}.

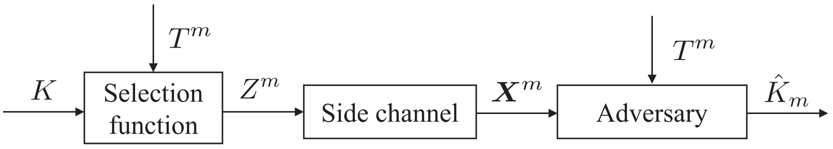


Figure 1: Communication channel model of SCA by de Chérisey et al. [dCGRP19].

2.3 Conventional communication channel model and MI–SR inequality

Figure 1 shows a communication channel model developed by de Chérisey et al. [dCGRP19] that represents an SCA, where K, $T^m = (T_1, T_2, \dots, T_m)$, $Z^m = (Z_1, Z_2, \dots, Z_m)$, $\boldsymbol{X}^m = (\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_m)$, and $\hat{K}_m$ represent the secret key byte, tuple of m plaintext/ciphertext bytes, secret intermediate values targeted by SCA, side-channel traces, and secret key byte estimated using m traces, respectively. In the "Adversary," the attacker uses an arbitrary distinguisher including the optimal one (e.g., DL-SCA using $q_{\hat{\theta}} = p_{Z|\boldsymbol{X}}$). Let $\mathrm{SR}_m = \Pr(K = \hat{K}_m)$ denote the SR of the optimal SCA with m traces. Based on this model, de Chérisey et al. proved the following theorem, which states that the optimal SR is bounded from above by the MI between the side-channel trace and intermediate value.

Theorem 1 (MI–SR inequality [dCGRP19])**.** *In the communication channel shown in Figure 1, the SR of an optimal SCA with m traces, defined as $\mathrm{SR}_m = \Pr(\hat{K}_m = K)$, is bounded from above as*

$$\xi(\mathrm{SR}_m) \leq mI(Z; \boldsymbol{X}), \tag{3}$$

where $\xi : [0, 1] \to \mathbb{R}_{\geq 0}$ is defined as

$$\xi(r) = H(K) - H_2(r) - (1 - r)\log(2^n - 1). \tag{4}$$

Here, $H_2 : [0, 1] \ni r \mapsto -r \log r - (1 - r) \log(1 - r) \in [0, 1]$ is a binary entropy function with a definition of $0 \log 0 = 0$.

In [dCGRP19], de Chérisey et al. demonstrated experimentally that this theorem can be used for a precise estimation of the achievable SR for an optimal SCA. However, this inequality is used only to evaluate the optimal SCA, and it cannot be used to evaluate a DL-SCA with the model parameters $\hat{\theta}$. In fact, an adversary in a communication channel is supposed to use an optimal distinguisher. In addition, neither the communication channel nor Equation (3) includes the model parameter $\hat{\theta}$. Thus, it is impossible to discuss the SR of DL-SCA with the model parameters $\hat{\theta}$. This motivated us to develop a new communication channel model for discussing the SR of DL-SCAs.

3 Information-theoretical analyses on DL-SCA

3.1 Overview

We model the DL-SCA as a communication channel. Based on this communication channel, we then derive an inequality, in which the SR of the DL-SCA is bounded from above by the MI between the output of the NNs and the intermediate value Z. Finally, we discuss the relationship between our inequality and the MI–SR inequality reported by de Chérisey et al. and show that our inequality is more useful for evaluating the performance of DL-SCAs. Our major technical contribution here includes the establishment of the communication channel model of DL-SCA and the discovery of the LPI–SR inequality as a new upper bound of SR, rather than the development of new proof techniques.

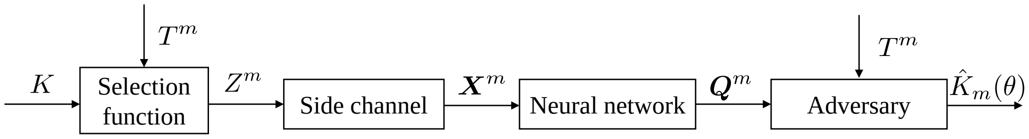


Figure 2: Communication channel of DL-SCA.

3.2 Communication channel model of DL-SCA

Figure 2 shows the proposed communication channel model of DL-SCA for a given NN model. This study focuses on attacks against a block cipher such as AES. We describe the definition of random variables and the meanings of this model.

- $m \in \mathbb{N}$ represents the number of traces used in the attack phase.
- $\theta \in \mathbb{R}^{n_\theta}$ represents the model parameters of the neural network, where n_θ represents the number of dimensions of the parameters.
- K and $\hat{K}_m(\theta)$ represent the correct and estimated partial secret keys, respectively. They belong to the key space $\mathcal{K} = \{0,1\}^n$, where n represents the bit length of the secret key (i.e., $n = 8$ for AES). Here, the estimated key is expressed as $\hat{K}_m(\theta)$ to explicit that it depends on the NN parameters θ and the number of traces m.
- $T^m = (T_1, T_2, \ldots, T_m)$ represents the tuple of m plaintexts/ciphertexts corresponding to the attack traces, where $T_1, T_2, \ldots, T_m$ denote n-bit plaintext/ciphertext random variables. In this study, they are assumed to be sampled from the uniform distribution and i.i.d.
- $Z^m = (Z_1, Z_2, \ldots, Z_m)$ represents the tuple of m intermediate values corresponding to the attack traces. Each intermediate value is given by $Z_i = \phi(K, T_i), i \in \{1, 2, \ldots, m\}$ with a selection function $\phi : \mathcal{K} \times \mathcal{T} \to \mathcal{Z}$. Here, the selection function can be any function. For example, we frequently have $\phi(K, T_j) = \mathrm{Sbox}(K \oplus T_j)$ in attacking software AES implementations.
- $\boldsymbol{X}^m = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m)$ represent the tuple of m side-channel traces. Each trace $\boldsymbol{X}_i$ represents a random variable over an n_ℓ-dimensional Euclidean space $\mathbb{R}^{n_\ell}$. These random variables are assumed to be i.i.d.
- $\boldsymbol{Q}^m = (\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_m)$ represents a tuple of the outputs of the NN for the attack traces. For each i, $\boldsymbol{Q}_i$ represents a vector and is given by $\boldsymbol{Q}_i = f_\theta(\boldsymbol{X}_i)$, where $f_\theta : \mathcal{X} \to \mathcal{Q} = \mathbb{R}^{n_q}$ represents the NN with the model parameters θ, and n_q represents the number of output classes. For example, if the NN predicts the probability of intermediate values (i.e., $n_q = 2^n$), the output $\boldsymbol{Q}_i$ represents a 256-dimensional vector consisting of the outputs of the softmax function as $\boldsymbol{Q}_i = (q_\theta(0 \mid \boldsymbol{X}_i), q_\theta(1 \mid \boldsymbol{X}_i), \ldots, q_\theta(2^n - 1 \mid \boldsymbol{X}_i))$ for each i, where $n = 8$ for AES. Further, if the NN predicts the probability of the HW of the intermediate value, it is an $(n+1)$-dimensional vector given by $\boldsymbol{Q}_i = (q_\theta(0 \mid \boldsymbol{X}_i), q_\theta(1 \mid \boldsymbol{X}_i), \ldots, q_\theta(n \mid \boldsymbol{X}_i))$ for each i.

In Figure 2, “Adversary” estimates the secret key by using all outputs $\boldsymbol{Q}^m$ of the NN and the plaintexts/ciphertexts T^m. In [dCGRP19], de Chérisey et al. implied that a Markov chain $(K, T^m) \to (Z^m, T^m) \to (\boldsymbol{X}^m, T^m) \to (\boldsymbol{Q}^m, T^m) \to (\hat{K}_m(\theta), T^m)$ holds. This communication channel model does not depend on how the NN is trained or used during an attack. This means that, for example, the model is valid even if any loss function, such as the CER or ranking loss, is used as the loss function during training.

Based on the communication channel illustrated in Figure 2, we define a new information-theoretical metric named the LPI, which plays an essential role in our analysis and has desirable properties for evaluating DL-SCA.

Definition 1 (Latent Perceived Information (LPI))**.** Let θ represent the model parameters of a neural network and $q_\theta(Z \mid \boldsymbol{X})$ represent the conditional probability distribution modeled by the NN. In the communication channel in Figure 2, the LPI of the model is defined as

$$\mathrm{LPI}(q_\theta) \coloneqq I(Z; \boldsymbol{Q}) = H(Z) - H(Z \mid \boldsymbol{Q}).$$

3.3 Relationship between the SR and model outputs

We prove the following theorem, which states that an inequality between $\boldsymbol{Q}$ and SR holds on the communication channel in Figure 2, which we call the LPI–SR inequality.

Theorem 2 (LPI–SR inequality)**.** *In the communication channel shown in Figure 2, the SR with m traces using the NN with model parameters θ, defined as $\mathrm{SR}_m(\theta) = \Pr(\hat{K}_m(\theta) = K)$, is bounded above as*

$$\xi(\mathrm{SR}_m(\theta)) \leq mI(Z; \boldsymbol{Q}) = m\mathrm{LPI}(q_\theta), \tag{5}$$

where ξ denotes the function defined in Equation (4).

Proof. The proof is based on Fano's inequality [CT06, Theorem 2.10.1] similarly to the MI–SR inequality [dCGRP19, Lemma 2]. In their proof, they used the data processing inequality [CT06, Theorem 2.8.1] on the Markov chain of $(K, T^m) \to (Z^m, T^m) \to (\boldsymbol{X}^m, T^m)$ represented in Figure 1, given as

$$I(K, T^m; \boldsymbol{X}^m, T^m) \leq I(Z^m, T^m; \boldsymbol{X}^m, T^m).$$

In this proof, we alternatively focus on the Markov chain of $(K, T^m) \to (Z^m, T^m) \to (\boldsymbol{X}^m, T^m) \to (\boldsymbol{Q}^m, T^m)$ in Figure 2, implying that the corresponding data processing inequality is

$$I(K, T^m; \boldsymbol{Q}^m, T^m) \leq I(Z^m, T^m; \boldsymbol{Q}^m, T^m). \tag{6}$$

According to the relationship between MI and entropy, the left-hand side of Equation (6) can be written as

$$\begin{aligned} I(K, T^m; \boldsymbol{Q}^m, T^m) &= H(K, T^m) - H(K, T^m \mid \boldsymbol{Q}^m, T^m) \\ &= H(K) + mH(T) - H(K \mid \boldsymbol{Q}^m, T^m) \\ &= H(K) + mH(T) - H(K \mid \boldsymbol{Q}^m, T^m, \hat{K}_m(\theta)) \\ &\geq H(K) + mH(T) - H(K \mid \hat{K}_m(\theta)). \end{aligned}$$

The equation $H(K \mid T^m, \boldsymbol{Q}^m) = H(K \mid T^m, \boldsymbol{Q}^m, \hat{K}_m(\theta))$ holds because $\hat{K}_m(\theta)$ is a deterministic function of T^m and $\boldsymbol{Q}^m$. Using Fano's inequality, we obtain

$$H(K \mid \hat{K}_m(\theta)) \leq H_2(\mathrm{SR}_m(\theta)) + (1 - \mathrm{SR}_m(\theta)) \log(2^n - 1).$$

Thus, it holds

$$I(K, T^m; \boldsymbol{Q}^m, T^m) \geq H(K) + mH(T) - H_2(\mathrm{SR}_m(\theta)) - (1 - \mathrm{SR}_m(\theta)) \log(2^n - 1). \tag{7}$$

The right-hand side of Equation (6) can be written as

$$\begin{aligned} I(Z^m, T^m; \boldsymbol{Q}^m, T^m) &= H(Z^m, T^m) - H(Z^m, T^m \mid \boldsymbol{Q}^m, T^m) \\ &= H(T^m) + H(Z^m \mid T^m) - H(Z^m \mid \boldsymbol{Q}^m, T^m) \\ &= mH(T) + I(Z^m; \boldsymbol{Q}^m \mid T^m). \end{aligned} \tag{8}$$

By combining Equations (6) to (8) with [IUH22a, Lemma 4.2], we conclude that

$$H(K) - H_2(\mathrm{SR}_m(\theta)) - (1 - \mathrm{SR}_m(\theta)) \log(2^n - 1) \leq I(Z^m; \boldsymbol{Q}^m \mid T^m) \leq mI(Z; \boldsymbol{Q}),$$

as required. □

Theorem 2 states that an inequality similar to the one proved by Cherisey et al. is applicable to DL-SCAs. On the left-hand side of Equation (5), $\xi(\mathrm{SR}_m(\theta))$ represents the entropy required to achieve a given $\mathrm{SR}_m(\theta)$ in the DL-SCA using an NN model parameterized by θ with m traces. For example, if the bit length of the secret key is eight bits, an SR of 100% in the DL-SCA requires eight bits of key entropy. Consequently, we deduce $\xi(\mathrm{SR}_m(\theta)) = \xi(1) = 8$, as anticipated. Meanwhile, $mI(Z; \boldsymbol{Q})$ quantifies the information concerning the intermediate value that the NN model can potentially extract from the m traces. Therefore, this inequality indicates that, to achieve a certain SR, information about intermediate values potentially extracted by the NN (i.e., $mI(Z; \boldsymbol{Q})$) must surpass the information required to achieve it (i.e., $\xi(\mathrm{SR}_m(\theta))$).

In Equation (5), the MI $I(Z; \boldsymbol{Q})$ measures the extent of information regarding the intermediate value that can potentially be retrieved from the model output. This concept is intrinsically linked to PI and EPI, as described in Section 4.

The LPI–SR inequality is not constrained by the specifics of the NN models used. This is agnostic to the loss function, model architecture during training, and the distinguisher applied in the attack. Although this study primarily addresses scenarios in which the activation function of the output layer in the NN model is a softmax function, the inequality remains valid even with different activation functions. Consequently, the modeled target should not be strictly a probability distribution; it may also be a regression model (e.g., in SCAs, a regression model predicting labels, such as HW and HD, from traces can be employed). This distinction markedly separates it from the LPI–SR inequality for the ranking loss in DL-SCA, which depends directly on the distinguisher. In addition, this inequality provides an upper bound for the SR, which delineates the limit of DL-SCA performance. This characteristic is useful for security evaluation.

3.4 Difference from MI–SR inequality

We discuss the differences between the MI–SR inequality and our LPI–SR inequality to demonstrate that Equation (5) is more effective in evaluating DL-SCA performance.

In [dCGRP19], de Chérisey et al. established the inequality $\xi(\mathrm{SR}_m) \leq mI(Z; \boldsymbol{X})$ for the SR of side-channel attacks, not limited to DL-SCA. Here, SR_m represents the probability of successful attacks by any SCA using m traces. The function $\xi : [0, 1] \to \mathbb{R}_{\geq 0}$ converts the SR to entropy. Thus, $\xi(\mathrm{SR}_m)$ on the left-hand side represents the entropy of the key required for a certain SR by *any* SCA. Conversely, $mI(Z; \boldsymbol{X})$ quantifies the amount of information regarding intermediate values retrieved from m side-channel traces, where any extraction method is acceptable. This implies that, to achieve a given SR, the information about the intermediate values extractable from m traces $mI(Z; \boldsymbol{X})$ must exceed the necessary information of the secret key $\xi(\mathrm{SR}_m)$.

Further, this inequality must hold for DL-SCA. Let $\mathrm{SR}_m(\theta)$ denote the SR with model parameter θ, and $\xi(\mathrm{SR}_m(\theta)) \leq mI(Z; \boldsymbol{X})$ must hold. However, this inequality may be too loose to evaluate the attack performance (i.e., SR) of a specific model because the right-hand side does not depend on the model parameters. For example, if θ represents random values in the initial learning phase, the SR is assumed to be $\mathrm{SR}_m(\theta) \approx 2^{-n}$, which is as high as a random guess, regardless of the number of traces m. However, the right-hand side increases monotonically with m when $I(Z; \boldsymbol{X}) > 0$ because the MI $I(Z; \boldsymbol{X})$ is a constant independent of the model parameters. Therefore, the MI–SR inequality suggests that the attack can succeed with high probability (e.g., $\mathrm{SR}_m(\theta) = 1$) given a sufficient number of traces, even for ineffective model parameters. Hence, the MI–SR inequality is unsuitable for evaluating DL-SCA models.

In contrast, $\mathrm{LPI}(q_\theta) = I(Z; \boldsymbol{Q})$ on the right-hand side of Equation (5) depends on the model parameters. For example, if the model parameters are unsuitable for the attack, the NN output is likely to be a random number with little relevance to the intermediate values. In such cases, the MI between Z and $\boldsymbol{Q}$ is approximately zero. Thus, $\xi(\mathrm{SR}_m(\theta)) \leq mI(Z; \boldsymbol{Q}) \approx 0$, followed by $\mathrm{SR}_m(\theta) \approx 2^{-n}$.[1] This suggests that the LPI–SR inequality is more suitable for evaluating DL-SCA than the MI–SR inequality.

In addition, the difficulty in MI estimation may differ significantly between $I(Z; \boldsymbol{X})$ and $\mathrm{LPI}(q_\theta) = I(Z; \boldsymbol{Q})$. The NN output $\boldsymbol{Q}$ has a lower dimensionality than $\boldsymbol{X}$, and therefore, the LPI estimation is dimensionally reduced to extract information on intermediate values compared with the MI estimation. Therefore, estimating $\mathrm{LPI}(q_\theta)$ is simpler than estimating $I(Z; \boldsymbol{X})$. The methodology for estimating LPI is detailed in Section 4.5.

4 Relationship with other perceived information metrics

4.1 Overview

In this section, we first review the PI and its related information quantity, EPI. Next, we explain the relationship among PI, EPI, and LPI and demonstrate that SR can be estimated when PI and EPI are equal to LPI. Further, we examine when PI and EPI are equal to LPI in terms of the conditional probability distributions. Finally, we propose an LPI estimation method based on the adjustment of NN outputs. We introduce our method by explaining its basic concept from the perspectives of EPI and confidence calibration.

4.2 Review of PI

Definition of PI. In EUROCRYPT 2011 [RSVC+11], Renauld et al. defined PI as follows:

Definition 2 (Perceived Information (PI))**.** Let θ represent the model parameters of an NN and $q_\theta(Z \mid \boldsymbol{X})$ represent the conditional probability distribution modeled by this model. The PI of the model is defined as

$$\mathrm{PI}(q_\theta) \coloneqq H(Z) + \mathbb{E} \log q_\theta(Z \mid \boldsymbol{X}) = H(Z) - \mathrm{CE}(q_\theta).$$

PI has two important properties.

(i) PI can be approximated by NLL. We obtain $\mathrm{PI}(q_\theta) = H(Z) - \mathrm{CE}(q_\theta)$ using the CE $\mathrm{CE}(q_\theta) = -\mathbb{E} \log q_\theta(Z \mid \boldsymbol{X})$. Further, the CE can be approximated using NLL $L(q_\theta)$ when the number of traces m is sufficiently large, implying that $\mathrm{PI}(q_\theta) \approx H(Z) - L(q_\theta)$. In other words, a decrease in the value of the NLL loss function during training is equivalent to an increase in PI.

[1] Note that ξ takes a global minimum value of 0 at 2^{-n} [IUH22a, Lemma 5.1]. This is deduced from the fact that the entropy required for a random guess of an n-bit key is zero.

(ii) PI is a lower bound of MI. Using the conditional entropy $H(Z \mid \boldsymbol{X}) = -\mathbb{E} \log p(Z \mid \boldsymbol{X})$, the MI between the trace and intermediate value is given by $I(Z; \boldsymbol{X}) = H(Z) - H(Z \mid \boldsymbol{X})$, where $p(Z \mid \boldsymbol{X})$ represents the true conditional distribution of the intermediate value Z given trace $\boldsymbol{X}$. According to the non-negativity of the Kullback–Leibler (KL) divergence $D_{\mathrm{KL}}(p_{Z|\boldsymbol{X}} \parallel q_\theta) = \mathbb{E} \log \frac{p(Z|\boldsymbol{X})}{q_\theta(Z|\boldsymbol{X})} \geq 0$ [CT06, Theorem 2.6.3], we have

$$H(Z \mid \boldsymbol{X}) = -\mathbb{E} \log p(Z \mid \boldsymbol{X}) \leq -\mathbb{E} \log q_\theta(Z \mid \boldsymbol{X}) = \mathrm{CE}(q_\theta).$$

Therefore, MI is bounded from below as $I(Z; \boldsymbol{X}) \geq H(Z) - \mathrm{CE}(q_\theta) = \mathrm{PI}(q_\theta)$.

Conjecture on the PI–SR inequality. These properties suggest an intuitive interpretation: the PI represents the amount of information of the intermediate value that the NN can extract from the trace [RSVC$^+$11]. To explain this, we consider the following properties. A decrease in the value of the NLL loss function increases the amount of information that an NN can extract (i.e., the PI). Then, the PI becomes equal to the MI when the NN can retrieve the information most successfully (i.e., when the PI is at its maximum). Meanwhile, when the model training completely fails (i.e., when NLL loss is large), the PI is approximately equal to or less than zero, and the inequality $I(Z; \boldsymbol{X}) \geq \mathrm{PI}(q_\theta)$ becomes meaningless. This can be considered as the model that does not extract any information about intermediate values from the traces. Based on these observations, we hypothesize that the PI represents the amount of information that the model can extract. Thus, Masure et al. [MDP20] conjectured that $\mathrm{SR}_m(\theta)$ is bound above by $\mathrm{PI}(q_\theta)$ as

$$\xi(\mathrm{SR}_m(\theta)) \leq m\mathrm{PI}(q_\theta),$$

which we call the PI–SR inequality in this study. Yet, this is not proven formally. Indeed, this is not always true, as described in Section 4.3.

4.3 Review of EPI

Transformation by inverse temperature. Ito et al. [IUH22b] showed that the conjecture on the PI–SR inequality is not always true by constructing a counterexample. They used the fact that a transformation of the softmax function using the inverse temperature $\beta > 0$ does not change the SR, whereas the PI varies depending on β. For the conditional probability distribution $q_\theta(Z \mid \boldsymbol{X})$ modeled by the NN, the transformed conditional probability distribution $q_\theta^{(\beta)}(Z \mid \boldsymbol{X})$ with an inverse temperature $\beta > 0$ is given by

$$q_\theta^{(\beta)}(z \mid \boldsymbol{x}) = \frac{(q_\theta(z \mid \boldsymbol{x}))^\beta}{\sum_{z'} (q_\theta(z' \mid \boldsymbol{x}))^\beta}. \tag{9}$$

Ito et al. showed that $\mathrm{CE}(q_\theta^{(\beta)}) \to \infty$ and $\mathrm{PI}(q_\theta^{(\beta)}) \to -\infty$ hold as $\beta \to \infty$ [IUH22b, Prpoposition 3]; which indicates that PI can take a negative value and be arbitrarily small by changing β. Hence, we can create a counterexample for the PI–SR inequality conjecture using a sufficient large β because the SR is invariant to this transformation for $\beta > 0$. Further, this suggests that the idea of PI expressing the amount of information that a model can retrieve is incorrect.

Definition of ECE and EPI. Ito et al. defined ECE and EPI to rectify the problem of the inverse temperature of PI. Their basic idea was calibrating the CE and PI to β for solving their uncertainty for an identical SR.

Definition 3 (Effective CE (ECE) and Effective PI (EPI) [IUH22b])**.** Let $q_\theta^{(\beta)}$ represent the probability distribution transformed with the inverse temperature $\beta \geq 0$ defined in

Equation (9). Using the same notation as Definition 2, ECE and EPI of q_θ are defined as

$$\mathrm{ECE}(q_\theta) \coloneqq \min_{\beta\geq 0} \mathrm{CE}(q_\theta^{(\beta)}) = \min_{\beta\geq 0} -\mathbb{E}\log q_\theta^{(\beta)}(Z \mid \boldsymbol{X}),$$
$$\mathrm{EPI}(q_\theta) \coloneqq \max_{\beta\geq 0} \mathrm{PI}(q_\theta^{(\beta)}) = \max_{\beta\geq 0} \Big(H(Z) + \mathbb{E}\log q_\theta^{(\beta)}(Z \mid \boldsymbol{X}) \Big) = H(Z) - \mathrm{ECE}(q_\theta),$$

respectively.[2]

Conjecture on EPI–SR inequality. Similar to Masure et al., Ito et al. also conjectured that the SR is bounded above by the EPI as

$$\xi(\mathrm{SR}_m(\theta)) \leq m\mathrm{EPI}(q_\theta),$$

which we refer to as EPI–SR inequality in this paper. Ito et al. [IUH22b] experimentally confirmed its validity and precision; that is, EPI enables precise SR evaluation through the inequality reported by de Chérisey et al. However, this is yet to be proven formally.

4.4 Relationship among PI, EPI, LPI, and MI

The LPI–SR inequality was formally proven; therefore, if LPI and EPI/PI are equivalent, the EPI/PI–SR inequality is valid. We discuss the relationship among PI, EPI, LPI, and MI, and describe them when the above two conjectures hold.

First, we have the following inequality among the PI, EPI, LPI, and MI. LPI is a lower bound of MI, tighter than PI and EPI.

Theorem 3 (Order of PI, EPI, LPI, and MI)**.**

$$\mathrm{PI}(q_\theta) \leq \mathrm{EPI}(q_\theta) \leq \mathrm{LPI}(q_\theta) \leq I(Z; \boldsymbol{X}). \tag{10}$$

Proof. From this definition, $\mathrm{PI}(q_\theta) \leq \mathrm{EPI}(q_\theta)$ holds. We have $\mathrm{LPI}(q_\theta) = I(Z;\boldsymbol{Q}) \leq I(Z;\boldsymbol{X})$ according to the data processing inequality on the Markov chain of $Z^m \to \boldsymbol{X}^m \to \boldsymbol{Q}^m$ in Figure 2. We then demonstrate that $\mathrm{EPI}(q_\theta) \leq I(Z;\boldsymbol{Q})$ holds. As $\mathrm{EPI}(q_\theta) = H(Z) - \min_{\beta\geq 0}\mathrm{CE}(q^{(\beta)})$ and $\mathrm{LPI}(q_\theta) = I(Z;\boldsymbol{Q}) = H(Z) - H(Z \mid \boldsymbol{Q})$ hold, it suffices to show that

$$\min_{\beta\geq 0} \mathrm{CE}(q^{(\beta)}) \geq H(Z \mid \boldsymbol{Q}).$$

We have $H(Z \mid \boldsymbol{X}) \geq H(Z \mid \boldsymbol{Q})$ from $I(Z;\boldsymbol{Q}) \leq I(Z;\boldsymbol{X})$. This indicates that $\mathrm{CE}(q^{(\beta)}) \geq H(Z \mid \boldsymbol{X})$ holds, regardless of the value of β, which is followed by

$$\mathrm{CE}(q_\theta^{(\beta)}) \geq H(Z \mid \boldsymbol{X}) \geq H(Z \mid \boldsymbol{Q}),$$

for any β. This completes the proof. □

From Theorem 3, we have the following corollary.

Corollary 1.

$$\mathrm{CE}(q_\theta) \geq \mathrm{ECE}(q_\theta) \geq H(Z \mid \boldsymbol{Q}) \geq H(Z \mid \boldsymbol{X}).$$

From Theorem 3, the conjectures of the PI–SR and EPI–SR inequalities hold if the equality in Equation (10) holds. We first prove Theorem 4, which states that the condition for the EPI–SR inequality holds.

[2] In the original paper, these are defined using inf and sup; however, in this paper, they are defined using min and max for simplicity, as $\mathrm{CE}(q_\theta^{(\beta)})$ always has a global minimum in terms of $\beta \geq 0$.

Theorem 4 (Equality condition between EPI and LPI)**.** $\mathrm{EPI}(q_\theta) = \mathrm{LPI}(q_\theta)$ *holds if and only if*

$$D_{\mathrm{KL}}\Big(p(Z \mid \boldsymbol{X}) \,\Big\|\, q_\theta^{(\beta')}(Z \mid \boldsymbol{X})\Big) = D_{\mathrm{KL}}\Big(p(Z \mid \boldsymbol{X}) \,\Big\|\, p(Z \mid f_\theta(\boldsymbol{X}))\Big),$$

where D_{KL} *represents the KL divergence and* $\beta' = \arg\min_{\beta \geq 0} \mathrm{CE}(q_\theta^{(\beta)})$*; that is,* $q_\theta^{(\beta')}$ *represents the probability distribution used to calculate* $\mathrm{EPI}(q_\theta)$*.*

Proof. Recall that $\mathrm{LPI}(q_\theta) = H(Z) - H(Z \mid \boldsymbol{Q})$ and $\mathrm{EPI}(q_\theta) = H(Z) - \min_{\beta > 0} \mathrm{CE}(q_\theta^{(\beta)})$. The condition $\mathrm{LPI}(q_\theta) = \mathrm{EPI}(q_\theta)$ is equivalent to

$$H(Z \mid \boldsymbol{Q}) = \min_{\beta \geq 0} \mathrm{CE}(q_\theta^{(\beta)}) = \mathrm{CE}(q_\theta^{(\beta')}). \tag{11}$$

$H(Z \mid \boldsymbol{Q}) = -\mathbb{E} \log p(Z \mid \boldsymbol{Q}) = -\mathbb{E} \log p(Z \mid f_\theta(\boldsymbol{X}))$ holds because of the definition of conditional entropy, and therefore, Equation (11) is equivalent to

$$\begin{aligned} &\mathbb{E} \log p(Z \mid f_\theta(\boldsymbol{X})) = \mathbb{E} \log q_\theta^{(\beta')}(Z \mid \boldsymbol{X}) \\ &\Leftrightarrow \mathbb{E} \log \frac{p(Z \mid \boldsymbol{X})}{p(Z \mid f_\theta(\boldsymbol{X}))} = \mathbb{E} \log \frac{p(Z \mid \boldsymbol{X})}{q_\theta^{(\beta')}(Z \mid \boldsymbol{X})} \\ &\Leftrightarrow D_{\mathrm{KL}}\Big(p(Z \mid \boldsymbol{X}) \,\Big\|\, p(Z \mid f_\theta(\boldsymbol{X}))\Big) = D_{\mathrm{KL}}\Big(p(Z \mid \boldsymbol{X}) \,\Big\|\, q_\theta^{(\beta')}(Z \mid \boldsymbol{X})\Big), \end{aligned} \tag{12}$$

as required for the sufficiency proof. Equation (12) can be inversely transformed to Equation (11) as required for the necessity proof. This completes the proof. □

Theorem 4 states that EPI is equal to LPI if the distance between the true distribution $p(Z \mid \boldsymbol{X})$ and the distribution $p(Z \mid f_\theta(\boldsymbol{X}))$ is the same as the distance between the true distribution and the modeled probability $q_\theta^{(\beta')}(Z \mid \boldsymbol{X})$. An important case is that $p(Z \mid f_\theta(\boldsymbol{X})) = q_\theta^{(\beta')}(Z \mid \boldsymbol{X})$ holds. An intuitive explanation of this is provided below.

As stated in [IUH22b, Lemma 1], the calibration using β' reduces the CE without changing the ranks of the key candidates. The distribution $q_\theta^{(\beta')}(Z \mid \boldsymbol{X})$ represents the closest distribution to the true distribution when considering the degree of freedom (in the sense that the rank of the correct key does not change) with respect to the inverse temperature. However, in addition to the inverse temperature, other transformations that reduce the CE without changing the SCA performance (i.e., SR) may exist[3]. It is unclear whether the distribution $q_\theta^{(\beta')}(Z \mid \boldsymbol{X})$, which is optimized only with respect to the inverse temperature β, is the best distribution, in the sense that the SCA performance is invariant. In contrast, $p(Z \mid f_\theta(\boldsymbol{X}))$ represents the probability distribution of the best prediction of the intermediate value Z from the NN output $f_\theta(\boldsymbol{X})$ using all possible methods (transformations). All transformations here include transformation by inverse temperature. In other words, distribution $p(Z \mid f_\theta(\boldsymbol{X}))$ represents a distribution closer to the true distribution than distribution $q_\theta^{(\beta')}(Z \mid \boldsymbol{X})$. The equality of these distributions implies that the transformation to obtain the best predictive distribution of the intermediate value Z from the output $\boldsymbol{Q}$ of the NN model is limited to that obtained using the inverse temperatures.

[3] Ito et al. [IUH22b] mentioned this open problem as follows. *Theorem 2 states that the proposed metrics are the most appropriate for SR and GE evaluations for any conversions of probability distribution that preserve the order of key ranks. However, ECE and EPI are not guaranteed to be unique to an SR and are appropriate for SR evaluation if there is a conversion such that the SR is preserved but the order of key ranks is not preserved. ECE and EPI are truly unique in terms of the SR evaluation if there is no such conversion. An analysis of the existence of such a conversion is important for future work.* (*Theorem 2* is theirs, *the proposed metrics* are ECE and EPI, and *GE* refer to guessing entropy.) LPI is considered a more appropriate metric that covers this problem than EPI.

In Section 6, we experimentally confirm whether such a transformation is limited to the inverse temperature (i.e., the equivalence between LPI and EPI) and show that this holds approximately, which proves the conjecture on the EPI–SR inequality in the experiment.

Finally, the equality condition between the PI and LPI is that $q_\theta(Z \mid \boldsymbol{X}) = q_\theta^{(\beta')}(Z \mid \boldsymbol{X}) = p(Z \mid \boldsymbol{Q})$. This represents a situation where the trained model $q_\theta(Z \mid \boldsymbol{X})$ is equal to the distribution $q_\theta^{(\beta')}(Z \mid \boldsymbol{X})$ obtained by minimizing the CE on the inverse temperature, and it is also equal to the distribution $p(Z \mid \boldsymbol{Q})$ simultaneously. In [IUH22b], it was shown experimentally that $q_\theta(Z \mid \boldsymbol{X}) = q_\theta^{(\beta')}(Z \mid \boldsymbol{X})$ rarely holds, which implies that the PI–SR inequality may not be valid. We will also demonstrate this experimentally in Section 6.

4.5 Estimation of LPI

The objective of training NNs in the DL-SCA is to predict the intermediate value Z from the input trace $\boldsymbol{X}$. Like conventional multi-class classification by DL, the NN in the DL-SCA comprises a *feature extractor* that extracts information from the trace and a *classifier* that performs classification from the information extracted by the feature extractor. Thus far, in the architectures proposed for the DL-SCA, the dimensions of the feature extractor are often smaller than those of the output. For example, the architectures proposed by Zaid et al. [ZBHV19] and Wouters et al. [WAGP20] have CNN outputs of only about a dozen dimensions. However, the output of the NN had at least 256 dimensions (for AES) when ID leakage models were used. Therefore, in such models, the feature extractor extracts and *compresses* information from a trace, whereas the classifier uses the *compressed* information to make predictions in a high-dimensional space. If the classifier is perfectly trained, the distribution $q_{\hat{\theta}}(Z \mid \boldsymbol{X})$ matches the distribution $p(Z \mid \boldsymbol{Q})$, which provides the best predictions of the intermediate values using information extracted by the NN. However, Ito et al. [IUH22b] proved that the conjecture on the PI–SR inequality does not always hold, thereby indicating that a difference exists between PI and LPI. Further, this means that the distributions $q_{\hat{\theta}}(Z \mid \boldsymbol{X})$ and $p(Z \mid \boldsymbol{Q})$ do not coincide, and that the distribution of the classifier is not perfect.

One reason the distributions that compose the classifier are not perfect is the *overconfidence* of the model, wherein the model assigns an unreasonably high confidence level to a label that it judges as correct. In the machine-learning community, the overconfidence of the model has led to the point that the output of a model cannot be used as it is as a probability. To solve this problem, a method called *temperature scaling* has been proposed [GPSW17]; this method is similar to the conversion in ECE/EPI by Ito et al. (i.e., Equation (9)). Temperature scaling uses inverse temperature calibration for preventing the model output from reaching extreme values and ensuring that the prediction confidence matches the accuracy value. The confidence of the model matches very well with the probability that the model guesses the correct label using temperature scaling. The value of this distribution $p(Z \mid \boldsymbol{Q})$ is expected to be close to the accuracy of each class of models because the distribution $p(Z \mid \boldsymbol{Q})$ represents how well the intermediate value Z can be estimated by the output $\boldsymbol{Q}$ of the model. Therefore, temperature scaling is expected to calibrate the model output such that $q_{\hat{\theta}}(Z \mid \boldsymbol{Q}) \approx p(Z \mid \boldsymbol{Q})$. Calibrating the model output $q_{\hat{\theta}}(Z \mid \boldsymbol{X})$ like the temperature scaling is necessary to estimate the distribution $p(Z \mid \boldsymbol{Q})$ because temperature scaling has been shown to work well experimentally.

In this study, we propose a method to estimate the LPI by performing adjustments using a different NN. First, let $\gamma_{\hat{\theta}}(\boldsymbol{x})$ denote the logits of a trained NN when the input trace $\boldsymbol{x}$ is provided. This trained model is acquired during the profiling phase of the standard DL-SCA. In our approach, we train another NN g_ω to estimate Z from $\gamma_{\hat{\theta}}(\boldsymbol{x})$, where ω denotes model parameters. We use the NN g_ω to model the distribution of z from $\boldsymbol{x}$ as $\mathrm{softmax}(\gamma_{\hat{\theta}}(\boldsymbol{x}) \odot g_\omega(\gamma_{\hat{\theta}}(\boldsymbol{x})))$, where the operator $\odot$ indicates the Hadamard product (i.e., element-wise multiplication). This indicates that, similar to the temperature scaling and conversion in ECE/EPI, the proposed method involves calibrating the inverse temperature β through the Hadamard product with the trained $g_\omega(\boldsymbol{x})$. Meanwhile, the inverse temperature is determined based on the logits of the classifier using an NN, and it can be set to different values for each dimension of the logit. Consequently, $\mathrm{softmax}(\gamma_{\hat{\theta}}(\boldsymbol{x}) \odot g_\omega(\gamma_{\hat{\theta}}(\boldsymbol{x})))$ will yield the true distribution $p(Z \mid \boldsymbol{Q})$ with regard to over-confidence if model parameters ω are learned successfully. Therefore, we can estimate the LPI value empirically using $\mathrm{softmax}(\gamma_{\hat{\theta}}(\boldsymbol{x}) \odot g_\omega(\gamma_{\hat{\theta}}(\boldsymbol{x})))$ as $p(z \mid \boldsymbol{Q})$.

5 Validity of the NLL loss for DL-SCA

5.1 overview

In this section, we discuss why training NNs with NLL loss functions can be beneficial for DL-SCA from the LPI perspective. To this end, we first explain that training NNs to decrease the NLL loss approximately increases the LPI. Then, we explain that increasing LPI raises the upper bound of the SR, thereby resulting in an improvement in the SR. The explanation in this section is experimentally validated in Section 6.

5.2 DL-SCA performance and loss functions

In conventional DL-SCAs, models are trained for minimizing the NLL between the computed intermediate value from the correct key and trace. During the key estimation process, the NLL for each key candidate is calculated across multiple traces, and we identify the candidate with the smallest NLL as the correct key. This practice lends credence to using an NLL corresponding to the correct key as a loss function; however, the primary objective of training in DL-SCA should not only decrease the NLL associated with the correct key but also improve (i.e., move up) the rank of the correct key. Simply decreasing the NLL of the correct key is insufficient, and its NLL must be smaller than those of all wrong key candidates. Thus far, employing NLL as a loss function during training does not indicate that the NLL of the correct key is smaller than NLLs of wrong key candidates, raising doubts on the inherent superiority of the NLL loss function in DL-SCAs.

Several studies proposed alternative loss functions, including CER loss [ZZN+20] and ranking loss [ZBD+21], which were designed to directly increase the SR to address this issue. These functions incorporate the attack process (i.e., computing and comparing the NLL for each key candidate) during model training, thereby directly improving the rank of the correct key. This approach seems reasonable because an SCA succeeds if the rank of the correct key is lower than that of the other keys.

However, in contrast to this intuition, [ISUH21, ZBD+20] show that, by introducing a complemental term to the model output or using a sufficiently large number of traces, the performance of training using the NLL loss function is comparable to or better than that using these specialized loss functions for DL-SCA. This section explains the counterintuitive phenomena in terms of LPI. Our conclusions can be summarized as follows:

1. A decrease in the NLL is approximately equal to an increase in the LPI.

2. An increase in the upper bound of SR in Theorem 2 is expected to increase the SR value if the selection function has some good properties.

5.3 NLL decrease and LPI increase

NLL is an approximation of the CE function because it almost surely converges to CE as the number of traces approaches infinity. Therefore, the decrease in NLL is approximately equal to the decrease in CE. The definition of ECE indicates that $\mathrm{ECE}(q_\theta) \leq \mathrm{CE}(q_\theta)$.

Therefore, training the model to decrease the NLL loss implies a decrease in ECE. We prove Theorem 5 to clarify the relationship between NLL/CE and ECE.

Theorem 5 (CE minimization implies ECE minimization for an NN)**.** *Let $\Omega = \mathbb{R}^{n_\theta}$ represent the set of all possible parameters of the NN. Let q_θ represent the probability distribution modeled by the NN with the parameters θ, where we assume that the last layer of the NN is a fully-connected layer with the softmax function as its activation function. That is, q_θ is represented by $q_\theta(z \mid \boldsymbol{x}) = \mathrm{softmax}(W_\theta h(\boldsymbol{x}) + b_\theta)_z$, where $h(\boldsymbol{x})$ denotes the input to the last layer if a trace $\boldsymbol{x}$ is input to the model; W_θ and b_θ represent the weight and bias of the last layer of the model, respectively; and the subscript z of the softmax function represents the z-th dimension of the output. It holds that*

$$\arg\min_\theta \mathrm{CE}(q_\theta) \subset \arg\min_\theta \mathrm{ECE}(q_\theta).$$

Proof. It is trivial that $\arg\min_\theta \mathrm{CE}(q_\theta) \subset \arg\min_\theta \mathrm{ECE}(q_\theta)$ holds if $\arg\min_\theta \mathrm{CE}(q_\theta) = \emptyset$. We assume that $\arg\min_\theta \mathrm{CE}(q_\theta) \neq \emptyset$ holds. Let $\mathcal{Q} = \{ q_\theta \mid \theta \in \Omega \}$ represent a set of all possible distributions modeled by the NN. Let $\mathcal{Q}' = \{ q_\theta^{(\beta)} \mid \theta \in \Omega, \beta \in \mathbb{R}_{\geq 0} \}$ represent the set of all possible distributions obtained using the inverse temperature $\beta \geq 0$ for the distribution in $\mathcal{Q}$. First, we prove that $\mathcal{Q} = \mathcal{Q}'$. From this definition, we obtain $\mathcal{Q} \subset \mathcal{Q}'$. Let $q_{\theta'}^{(\beta')} \in \mathcal{Q}'$ be any distribution in $\mathcal{Q}'$, which is defined as

$$q_{\theta'}^{(\beta')}(z \mid \boldsymbol{x}) = \frac{(q_{\theta'}(z \mid \boldsymbol{x}))^{\beta'}}{\sum_{z'} (q_{\theta'}(z' \mid \boldsymbol{x}))^{\beta'}}.$$

Here, $q_{\theta'}$ is given by $q_{\theta'}(z \mid \boldsymbol{x}) = \frac{\exp(-(W_{\theta'} h(\boldsymbol{x}) + b_{\theta'})_z)}{\sum_{z''} \exp(-(W_{\theta'} h(\boldsymbol{x}) + b_{\theta'})_{z''})}$ because of the assumption that the NN employs the softmax function in the output layer. Thus, it holds that

$$q_{\theta'}^{(\beta')}(z \mid \boldsymbol{x}) = \frac{\exp(-(\beta' W_{\theta'} h(\boldsymbol{x}) + \beta' b_{\theta'})_z)}{\sum_{z'} \exp(-(\beta' W_{\theta'} h(\boldsymbol{x}) + \beta' b_{\theta'})_{z'})} = \mathrm{Softmax}(\beta' W_\theta h(\boldsymbol{x}) + \beta' b_\theta)_z.$$

Therefore, the distribution $q_{\theta''}$ corresponding to the parameter θ'' with the weight and bias of the last layer of the parameter θ' replaced by $\beta' W_{\theta'}$ and $\beta' b_{\theta'}$, respectively, is equal to $q_{\theta'}^{(\beta')}$. Therefore, we have $q_{\theta'}^{(\beta')} \in \mathcal{Q}$ for any θ' and β'; thus, $\mathcal{Q} = \mathcal{Q}'$ holds.

According to $\mathcal{Q} = \mathcal{Q}'$, it holds that

$$\min_\theta \mathrm{CE}(q_\theta) = \min_{q_\theta \in \mathcal{Q}} \mathrm{CE}(q_\theta) = \min_{q_{\theta'}^{(\beta)} \in \mathcal{Q}'} \mathrm{CE}(q_{\theta'}^{(\beta)}) = \min_{\theta'} \min_{\beta \geq 0} \mathrm{CE}(q_{\theta'}^{(\beta)}).$$

Therefore, the minimum ECE value must exist if the minimum CE value exists, and these values must be equal.

Let $\theta' \in \arg\min_\theta \mathrm{CE}(q_{\theta'})$ represent the minimum value of CE. The minimum values of CE and ECE must be equal, because

$$\mathrm{CE}(q_{\theta'}) = \mathrm{CE}(q_{\theta'}^{(1)}) = \min_{\theta''} \min_{\beta \geq 0} \mathrm{CE}(q_{\theta''}^{(\beta)}),$$

which is followed by $\theta' \in \arg\min_{\theta''} \mathrm{ECE}(q_{\theta''})$. Therefore, we conclude that $\arg\min_\theta \mathrm{CE}(q_\theta) \subset \arg\min_\theta \mathrm{ECE}(q_\theta)$, as required. □

Theorem 5 indicates that the parameters minimizing CE value also minimize ECE value. Therefore, CE can be considered as a surrogate loss for ECE. Moreover, ECE bounds the entropy $H(Z \mid \boldsymbol{Q})$ from above, thereby enlarging the LPI by minimizing CE.

5.4 Relationship between increases in the SR upper bound and SR itself

Thus far, we observed that a decrease in the NLL approximately increases the LPI, thereby yielding an increase in the *upper bound* of the SR; however, it is not the SR itself. An increase in LPI does not necessarily lead to a direct increase in the SR itself, thereby implying that an increase in LPI may not improve the attack performance (i.e., SR). We discuss why, when, and how an increase in LPI is (in)consistent with SR improvement. Consequently, we clarify that such discrepancy between LPI and SR does not occur in attacks on typical block ciphers, including AES.

We cannot derive an inequality for the lower bound of SR from the NLL, instead of an upper bound, because the NLL does not utilize information about other key candidates. A decrease in NLL is approximately equivalent to a decrease in CE, which, in turn, indicates an increase in the LPI of the model (i.e., $I(Z; \boldsymbol{Q})$). The LPI measures the amount of information regarding intermediate values that can be extracted from the trace using the model. This signifies that, for example, one bit of information about the intermediate value can be extracted per trace if the LPI is one bit. Consequently, secret key information can be extracted at one bit per trace if the selection function is bijective. Intuitively, one can assume that eight traces would sufficiently estimate the partial key. However, we can create counterexamples in which such an assumption does not hold. For example, let us consider the case that $I(Z; \boldsymbol{Q}) = 1$ and the one-bit leakage is always the MSB of the partial key. In such a case, DL-SCA will only reveal the MSB of the secret key but cannot recover the other remaining secret key bits even when many traces are used. Thus, in this case, the attack will not be successful; that is, SR must be not more than 2^{n-1} despite the value of $mI(Z; \boldsymbol{Q})$.

Another potential issue arises when the intermediate values of multiple key candidates are similar, namely, when the dummy peaks are attributed to the correlation between correct and incorrect keys. Let us consider an extreme case in which a selection function is given by a bitwise AND operation (i.e., k & t), where k represents the eight-bit partial key candidate and t represents the plaintext. The key candidates $k_1 = (11111110)_2$ and $k_2 = (11111111)_2$ have similar intermediate values. These candidates yield identical intermediate values when the LSB of the plaintext is zero. In such a scenario, achieving SR $= 1$ with a single trace is inherently difficult even if the model perfectly extracts intermediate value information from the trace (i.e., LPI is eight bits). More importantly, if there were an incorrect key that behaved the same as the correct key, it would be considerably difficult to achieve an SR of more than 1/2.

Therefore, despite the ability of the model to retrieve substantial information from intermediate values, the difficulty in key recovery depends on the form of leakage and selection functions. In other words, Theorem 2 evaluates the SR only in terms of the amount of information extracted by an NN per trace (i.e., LPI); however, it does not consider the form of leakage and selection function, whereas they have a nontrivial impact on the success of key recovery. This is one of the reasons Theorem 2 states an inequality rather than an equality. In some settings, the inequality would be loose depending on the leakage form and selection function. Note here that we just gave some theoretical counterexamples for the explanation, which may be impractical or pathological.

Contrarily, if the inequality is guaranteed to be meaningfully tight, it suffices for the SR estimation to evaluate the LPI as its upper bound, thereby validating the decrease in NLL (i.e., increase in LPI increase) to improve the SR. Most issues causing the looseness are related to the leakage form and selection function. If the selection function is bijective with respect to both plaintext and key and if the plaintexts are uniformly distributed, the aforementioned problems are mitigated. An example of such a selection function is observed in attacks on AES software implementations; namely, $Z^{(k)} = \mathrm{Sbox}(T \oplus k)$ represents a bijection for any fixed k. Actually, in [dCGRP19], de Chérisey et al. experimentally confirmed the tightness of the SR upper bound based on Fano's inequality for the case of AES. Thus, in practice, the LPI–SR inequality would be meaningfully tight, which concludes that a decrease in NLL improves the DL-SCA performance. We will experimentally confirm the validity of the above discussion in Section 6.

6 Experimental validation

6.1 Dataset and experimental setup

This section validates our theoretical analyses through experimental DL-SCAs on masked AES software and hardware implementations. We evaluate the empirical values of PI, EPI, and LPI and examine whether the relationship among them mentioned in Section 4 and each SR inequality hold experimentally.

Target software. We used a first-order Boolean masked AES software provided by ASCAD [BPS+20]. We implemented it on Atmel Xmega128D4-AU eight-bit microcontroller and acquired the side-channel traces because we required more traces than the ASCAD dataset for improving the accuracy of estimating PI, LPI, and EPI. We acquired the side-channel traces for two different keys, which correspond to profiling (i.e., NN training) and attack phases. The target microcontroller is mounted on a ChipWhisperer CW308 UFO baseboard. The ChipWhisperer CW1200 capture box generated the base clock, and the clock frequency was set to 50 MHz. The microcontroller was connected to a Keysight DSOX6004A oscilloscope at a sampling rate of 20 GSamples/s for acquiring side-channel traces. The number of traces acquired was 100,000 each for training and attack phases. We used a selection function of $Z^{(k)} = \mathrm{Sbox}(T \oplus k)$, as in many previous studies on (DL-)SCA on ASCAD and AES software implementations; it is bijective for T and k.

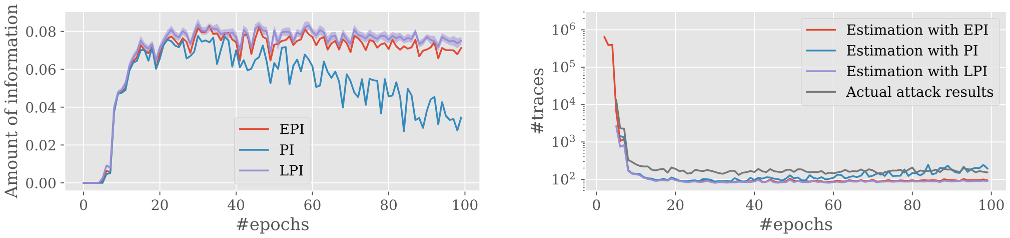
Target hardware. We used a public dataset released by Ito et al.[4], which was also used in the evaluation in [IUH22b]. The AES hardware implementation, presented in [UHA17], employs threshold implementation (TI) as a masking scheme [NRS11, RBN+15] with a byte-serial architecture. The implementation was conducted on a Xilinx Kintex-7 FPGA on a SAKURA-X board, and its power traces were acquired at a sampling rate of 455 MSa/s. The number of traces was 10,000,000 for training and 10,000,000 for the attacks. Similar to [IUH22a], we used a selection function targeting the register transition between the first and fifth bytes of the inversion output, denoted by

$$Z^{(k[1],k[5])} = \mathrm{Inv}(\Delta_f(T[1]) \oplus \Delta_f(k[1])) \oplus \mathrm{Inv}(\Delta_f(T[5]) \oplus \Delta_f(k[5])),$$

where $T[1]$ and $T[5]$ represent the first and fifth byte of plaintext, respectively; $k[1]$ and $k[5]$ represent the first and fifth byte of the secret key, respectively; Δ_f represents the isomorphic mapping from AES field (i.e., $\mathrm{GF}(2^8)$) to Canright's tower field (i.e., $\mathrm{GF}(((2^2)^2)^2)$) [Can05]; and Inv represents the $\mathrm{GF}(((2^2)^2)^2)$ inversion in Canright's Sbox (See [UHA17] for the details). We must guess two consecutive key bytes to employ an XOR-based selection function. Hence, the partial key length in the attack is $n = 16$ for the AES hardware implementation in our experiment. Although this function is not bijective, it satisfies a good property for DL-SCA called the *key independence condition* [IUH21].

Neural networks for $q_{\hat{\theta}}(Z \mid X)$ and $\gamma_{\hat{\theta}}(X)$ and its training. For evaluating the software implementation, we employ an NN architecture presented in [WAGP20] for implementing

[4] The dataset of side-channel traces is publicly available at https://github.com/ECSIS-lab/perceived_information_revisited/tree/main. The AES hardware is publicly available at https://github.com/ECSIS-lab/curse_of_re-encryption/tree/main/Masked_AES_hardware.



(a) PI values.

(b) Number of traces to achieve an SR of 90%.

Figure 3: DL-SCA on the AES software implementation.

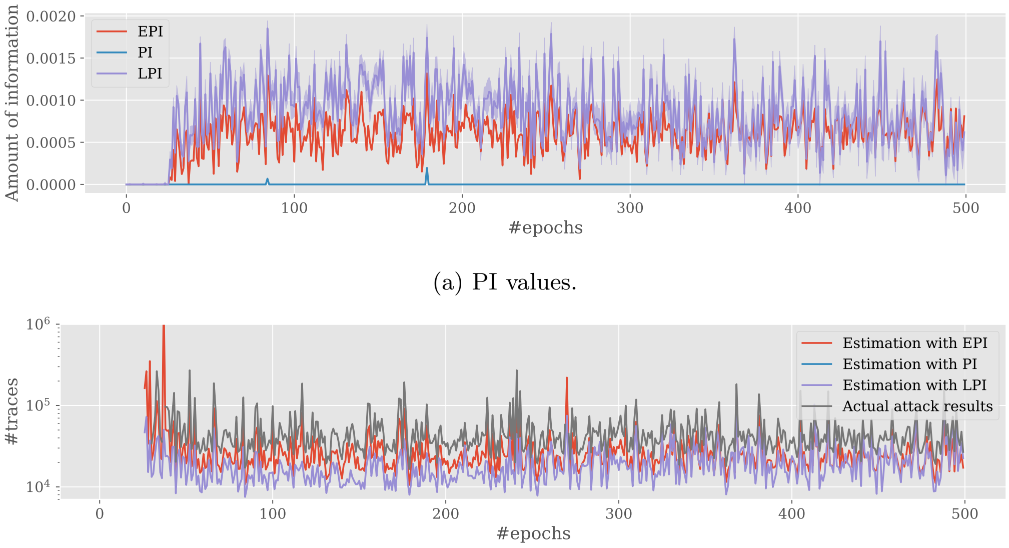
it with a random delay measure of 50 sample points. We set the learning rate, batch size, and maximum number of epochs to 1e−2, 512, and 100, respectively. We normalize the side-channel traces using `feature standardization` presented by Wouters et al. in [WAGP20]. For the hardware implementation, we used another neural network architecture proposed in [IUH22b], and set the learning rate, batch size, and maximum number of epochs to 1e−2, 1000, and 500, respectively. We normalize the side-channel traces using `feature scaling between -1 and 1` proposed in [WAGP20].

Neural network for g_ω and its training. We use the method described in Section 4.5 to estimate LPI. The architecture of the NN g_ω is a multilayer perceptron consisting of two layers with 16 and 256 units. We used batch normalization for the first layer and SELU as the activation function. The activation function of the last layer was set as a sigmoid function. We set the learning rate, batch size, and maximum number of epochs as 1e−3, 512, and 100, respectively. For training, we split the test data in half and used them as the training and validation sets for model g_ω. From the trained model $\gamma_{\hat{\theta}}$ and g_ω, we calculated softmax$(\gamma_{\hat{\theta}}(\boldsymbol{x}) \odot g_\omega(\gamma_{\hat{\theta}}(\boldsymbol{x})))$ and its NLL loss value, and it was used to update the model parameter ω and estimate the LPI value. The training of g_ω was repeated ten times, and the LPI was estimated using the model at the epoch with the smallest validation loss for each training. The mean and variance of the LPI estimates were evaluated in all trials.

6.2 Evaluation result

Figures 3 and 4 report (a) the estimated PI, EPI, and LPI values during training on the AES software and hardware implementations and (b) the empirical and estimated numbers of traces to achieve an SR of 90% using the model and inequalities, respectively. Precisely, Figures 3(a) and 4(a) show the estimated PI, EPI, and LPI values during training, where the horizontal axis represents the number of epochs and the vertical axis represents the amount of information. Figures 3(b) and 4(b) show the empirical and estimated numbers of traces to achieve an SR of 90% by the model at each epoch. The blue, red, and purple curves correspond to the number of traces required to achieve the SR estimated from the PI, EPI, and LPI values, respectively, using the corresponding inequality. The grey line denotes the empirical number of traces in an actual attack using the model.

We focus on the results of AES software implementation. From Figure 3(a), we observe that the PI value increases over the first 20 epochs and decreases subsequently. The values of EPI and LPI do not (relatively) decrease after 20 epochs. Therefore, after 20 epochs, the PI indicates a degradation in the attack performance of the model, whereas the EPI and LPI indicate a slight change in attack efficacy. In fact, Figure 3(b) indicates that the number of traces estimated from PI exceeds the number of traces in the actual attack at the final epoch, thereby disproving the conjecture on the PI–SR inequality in experiment,



(a) PI values.

(b) Number of traces to achieve an SR of 90%.

Figure 4: DL-SCA on the AES hardware implementation.

in addition to the theoretical argument. In contrast, the number of traces estimated from the EPI and LPI are smaller compared to those for the actual attack, and they correlate proportionally. The attack performance can be estimated from the LPI and EPI values through SR inequalities. The results show that PI < EPI < LPI holds, and the difference between LPI and EPI is negligibly small. Therefore, the EPI is appropriate for predicting attack performance, and the EPI–SR inequality is valid.

Next, we focus on the results of the AES hardware implementation, which was consistent with the results of the AES software implementation. The major difference was that the PI was smaller than zero in almost all epochs (the graph shows negative values of PI as zero). We expect an attack failure from the PI perspective (even with an unbounded number of traces); however, EPI and LPI indicate a successful attack because these values are greater than zero, even in early epochs. In fact, Figure 4(b) shows that the attack performance is predicted correctly from EPI and LPI using their corresponding inequalities. Therefore, we confirmed the validity of the EPI–SR inequality.

Finally, we confirm the relationship between the NLL loss function and SR. Figures 3(a) and 4(a) show that EPI and LPI increase as the learning progresses, even with NLL used as the loss function. This is because the NLL function is a surrogate loss of EPI and LPI, which leads to an increase in their values. The increase in LPI raises the upper bound of SR from Theorem 2, thereby contributing to the performance improvement.

7 Concluding remarks

This study analyzed the relationship between NLL and model attack performance in a DL-SCA. We developed a communication channel model for DL-SCA by defining LPI. Based on this model, we derived and proved the LPI–SR inequality. Then, we discussed the relationship among LPI, PI, and EPI to validate the conjecture on the EPI–SR inequality through the LPI–SR inequality, and we commented on the similarity between LPI and EPI. For the practical computation of LPI, we proposed a method to estimate LPI using an NN. We proved that an increase in the PI contributes to an increase in EPI and LPI, as in Theorem 5, which suggests that learning to decrease NLL increases the LPI and the upper bound of SR. Finally, we conducted an experimental DL-SCA on masked AES software and hardware implementations to validate the correctness of our analyses. In consequence, the LPI and EPI would be appropriate and valid metrics to evaluate the SCA performance (i.e., SR) for a given device in practice.

The experiments in this study were conducted for cases in which the selection function has good properties, such as bijection. Major practical ciphers, including AES, are included in this scope. However, the generality of our analysis for other cases is unclear, and the improvement in attack performance caused by an increase in the upper bound of the SR attributed to the decrease in NLL might be proven for limited cases. In the future, the appropriateness of using NLL as a loss function, even when the selection function does not have such good properties, should be investigated. Further, clarifications regarding the accuracy of the proposed LPI estimation method and the development of better alternatives should be pursued in the future.

References

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

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

[Can05] D. Canright. A very compact S-box for AES. In *International Workshop on Cryptographic Hardware and Embedded Systems*, volume 3659 of *Lecture Notes in Computer Science*, pages 441–455. Springer, 2005.

[CDP17] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures. In *Cryptographic Hardware and Embedded Systems CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 45–68. Springer, 2017.

[CRR02] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In *International Workshop on Cryptographic Hardware and Embedded Systems*, LNCS, pages 13–28, 2002.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.

[dCGRP19] Eloi de Chérisey, Sylvain Guilly, Olivier Rioul, and Pablo Piantanida. Best information is most successful: Mutual information and success rate in side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(2):49–79, 2019.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[GPSW17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330, 2017.

[HHGG20] Benjamin Hettwer, Tobias Horn, Stefan Gehrer, and Tim Güneysu. Encoding power traces as images for efficient side-channel analysis. In *2020 IEEE*

International Symposium on Hardware Oriented Security and Trust (HOST), pages 46–56, 2020.

[HRG14] Annelie Heuser, Olivier Rioul, and Sylvain Guilley. Good is not good enough - deriving optimal distinguishers from communication theory. In *CHES*, pages 55–74. Springer, 2014.

[ISUH21] Akira Ito, Kotaro Saito, Rei Ueno, and Naofumi Homma. Imbalanced data problems in deep learning-based side-channel attacks: Analysis and solution. *IEEE Transactions on Information Forensics and Security*, 16:3790–3802, 2021.

[IUH21] Akira Ito, Rei Ueno, and Naofumi Homma. Toward optimal deep-learning based side-channel attacks: Probability concentration inequality loss and its usage. Cryptology ePrint Archive, Report 2021/1216, 2021. https://ia.cr/2021/1216.

[IUH22a] Akira Ito, Rei Ueno, and Naofumi Homma. On the success rate of side-channel attacks on masked implementations: Information-theoretical bounds and their practical usage. Cryptology ePrint Archive, Report 2022/576, 2022. https://eprint.iacr.org/2022/576.

[IUH22b] Akira Ito, Rei Ueno, and Naofumi Homma. Perceived information revisited: New metrics to evaluate success rate of side-channel attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(4), 2022.

[IUTH23] Ito, Rei Ueno, Rikuma Tanaka, and Naofumi Homma. Formal analysis of non-profiled deep-learning based side-channel attacks. Cryptology ePrint Archive, Paper 2023/1563, 2023. https://eprint.iacr.org/2023/1563.

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

[MDP20] 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, 2020.

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

[MMZ23] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[MPP16] Houssem Maghrebi, Thibault Portigliatti, and E. Prouff. Breaking cryptographic implementations using deep learning techniques. *Security, Privacy, and Applied Cryptography Engineering (SPACE)*, 10076:3–26, 2016.

[NRS11] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of nonlinear functions in the presence of glitches. *Journal of Cryptology*, 24(2):292–321, 2011.

[PHJ+19a] Stjepan Picek, Annelie Heuser, Alan Jovic, Shivam Bhasin, and Francesco Regazzoni. The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, (1):209–237, 2019.

[PHJ+19b] 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, Issue 1:209–237, 2019.

[PPM+23] Stjepan Picek, Guilherme Perin, Luca Mariot, Lichao Wu, and Lejla Batina. SoK: Deep learning-based physical side-channel analysis. *ACM Computing Surveys*, 55(11):1–35, 2023.

[RBN+15] Oscar Reparaz, Begül Bilgin, Svetla Nikova, Benedikt Gierlichs, and Ingrid Verbauwhede. Consolidating masking schemes. In *Advances in Cryptology—CRYPTO 2015*, pages 764–783, 2015.

[RSVC+11] Mathieu Renauld, François-Xavier Standeart, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and side-channel attacks for nanoscale devices. In *Advances in Cryptology—Eurocrypt 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 109–128, 2011.

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

[SM23] Marvin Staib and Amir Moradi. Deep learning side-channel collision attack. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(3):422–444, 2023.

[TUX+23] Yutaro Tanaka, Rei Ueno, Keita Xagawa, AKira Ito, Junko Takahashi, and Naofumi Homma. Multiple-valued plaintext-checking side-channel attacks on post-quantum KEMs. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(3), 2023.

[UHA17] Rei Ueno, Naofumi Homma, and Takafumi Aoki. Toward more efficient DPA-resistant AES hardware architecture based on threshold implementation. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*, volume 10348 of *Lecture Notes in Computer Science*, pages 50–64, 2017.

[UXT+22] Rei Ueno, Keita Xagawa, Yutaro Tanaka, Akira Ito, Junko Takahashi, and Naofumi Homma. Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 1:296–322, 2022.

[WAGP20] Lennert Wouters, Victors Arribas, Benedikt Gierlichs, and Bart Praneel. Revisiting a methodology for efficient CNN architectures in profiling attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(3):147–168, 2020.

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

[ZBD$^+$21] 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, 2021.

[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, 2019.

[ZBHV21] Gabriel Zaid, Lilian Bossuet, Amaury Habrard, and Alexandre Venelli. Efficiency through diversity in ensemble models applied to side-channel attacks – a case study on public-key algorithms –. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2021(3):60–96, 2021.

[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, 2020.