Differential Metric based Deep Learning Methodology for Non-Profiled Side Channel Analysis

Gonella Vijayakanthi, Jaganath Prasad Mohanty, Ayas Kanta Swain, Kamalakanta Mahapatra
Department of Electronics and Communicaton Engineering, National Institute of Technology, Rourkela, India
{swaina, kkm}@nitrkl.ac.in

***Abstract*—Power Side-Channel analysis recovers sensitive information not only from physical proximity to a device but also from basic knowledge of sample leaked data collection. With minimum mean squared error metric, power analysis using a deep learning test case increase confidence level of proper identification of leaked data. Comparison with state-of-the art technology in this work shows improved performance in the non-profiled SCA category of detection. The deep learning technique aids in calculating the average loss gradient values and the loss values, both being calculated by taking the traces in mathworks implementation as the training data and the MSB values of the intermediate values as the training labels to reveal the expected secret key. Moreover iterative training of some machine learning techniques with different FPGA boards implementing cryptographic designs increased the accuracy of leakage detection at an earlier stage to a better extent.**

***Index Terms*—Side Channel Analysis; Machine Learning; Loss Gradient; Multi-Layer Perceptron; Differential Deep Learning Analysis; AES Cryptography.**

I. Introduction

Deep Learning (DL) architectures are based on NN (neural networks). In DL applications, deep neural networks are preferred. Based on forward propagation and backward propagation,the parameters can be tuned and a deep learning model can be trained. Unlike traditional approaches for Side channel attacks, machine learning techniques have been used in this investigation, eventually tagging with deep learning models, because of their respective advantages and usage in complex data. Deep learning techniques were used in Side channel attacks [7] where the model was trained with raw traces as training data, and MSB values of the intermediate values as the training labels [12].

In 2011, Gabriel et.al. [1] used Machine learning techniques for Side channel attacks and concluded that choice of parameters of an algorithm affects the accuracy of an attack. With similar trend in 2018, Naila Mukhtar et.al. analysed power traces leaked from FPGA implementation of Elliptic Curve Cryptography algorithm [13]. Simultaneously, in many comparative studies, the usage of different types of neural networks for Side channel attacks have been observed [17-24]. Later on, studies have been done on the implementation of Deep learning based techniques for Side channel attacks of a masked AES to reveal the secure key [14]. Further, secret key from a masked AES was revealed along with the mask location by using techniques related to deep learning [15].

Quite a lot of research has been done in the pretence of profiled power side channel analysis [10, 22], but rarely any for the non-profiled metrics [15,27], hence the necessity of research in this direction. A summarized view of the ongoing work on deep learning based side channel attacks on a few platforms are circumscribed in the Table 1.

TABLE I: Comparison with existing state of art technology

	Algorithms	*Masked*	*Traces*	*Technique*	*Methods*
[17]	AES	✓	DPA data-set	Deep Learn-ing(DL)	Accurate Profiling
[18]	AES	✗	FPGA	Unsupervised Learning	LSTM Auto-encoder
[20]	AES	✗	AVR micro-controller	DL MLP	DTW-PCA-MLP
[23]	AES	✓	FPGA	DL assisted	DNN-DVS
[25]	AES	✓	Microprocessor	Meta Trans-fer Learning	DNN-MTL

The main contribution of this work includes

- briefly discussing the **design of a non-profiled based DL-SCA attack** which uses less traces from a dataset to recover the key,
- calculating **the average loss gradient values and the loss values** with the aid of mean squared error statistical metric,
- Designing **tandem model attacks which uses 33.5 percent fewer traces** on average than single-model DL-SCAs to recover the key

The paper is structured as follows. Section II presents the background effort going around in this domain. The implementation aspects of the differential deep learning analysis, initiating from power tracing, to storing efficient data and extracting features are discussed in section III along with training neural networks and calculating the loss values from trained network to a certain level of accuracy. Section IV concludes the work in progress and trails interim future headway targeted in this direction.

II. Preliminaries

A Multi-Layer Perceptron (MLP) is a NN which is composed of several perceptron units. A bias value can be added

to the summation of the individual products of inputs and their respective weights. In the NN, $x_1..x_n$ is the input data taken as a vector which are individually connected to the hidden layer with the weights $w_1..w_n$. Based on equation 1, z is the output which is obtained after using an activation function 'f', used to define how the weighted sum of the input is transformed to the output.

$$z = f(b + \sum_{i=1}^{m}[w_i x_i]) \tag{1}$$

The equation 1 is used to calculate the value of the output layer of a neural network based on the bias value (b), Activation function (f), weights being w_i and input values being x_i respectively as shown. Rectified Linear Unit (ReLU) is an activation function used in neural networks to define the output by activating the neurons based on it's functionality which affects the output. It is defined as 0 when the input is negative and remains the same as the input if it is positive.

$$R(z) = max(0, z) \tag{2}$$

The output obtained by z is the predicted output. The deviation of this predicted value from the desired value is to be observed. This is usually done by calculating the loss function. The loss function can be calculated by some methods namely mean square error, mean absolute error etc. Calculation of loss using mean square error(MSE), used in this endeavor, is shown in equation 3.

$$MSE = \sum_{i=1}^{n} \frac{(e_i - y_i)^2}{n} \tag{3}$$

The equation 3 is used to calculate MSE (mean square error) value of y, for n number of values of y_i, where i ranges from 1 to n. Here e_i is the estimated value and y_i is the original value. The main goal of dealing with NN is to reduce this loss value. Firstly, the weights have to be modified to reduce the loss value. Similarly the bias values should also be modified for the reduction of loss. This is done by using an optimizer. Updating of the weights and the bias values is done during back propagation. Therefore, after the back propagation, new weights are proposed and then the forward propagation takes place. This is an iterative process and one cycle of forward propagation and backward propagation is called an epoch. Learning rate (a) is factor which is used to justify the rate at which the weights are updated.

Now a days deep learning techniques are in use in every field for various applications due to advancement in technology. Deep learning techniques are also in use to attack many cryptographic devices. A method of using deep learning architecture for finding the secret key from the given power traces and input data is discussed and implemented in this work, using python in Google Colaboratory by using the libraries required. The general idea behind the real time analysis of the endeavor can be understood from Fig. 1.

A. Deep learning based Side Channel Attack

After almost a decade of initial investigation on Side channel attacks (SCA), machine learning techniques have been used to reveal the secret data. A kernel based powerful learning technique Least Squares Support Vector Machine (LS-SVM) was used for the attack and the results strongly depicted that the parameters of the algorithm used have a strong impact on the accuracy of the result [8]. Pre-processing the input data has shown a considerable accuracy in results depicted in [9]. The usage of Deep Learning techniques for side channel attack began as they were more effective while working on the side channel analysis of a masked algorithm in comparison to any other machine learning techniques [4,5,11]. Further, without any pre-processing of the input data, deep learning techniques can produce the secret data unlike any traditional machine learning approaches [10].

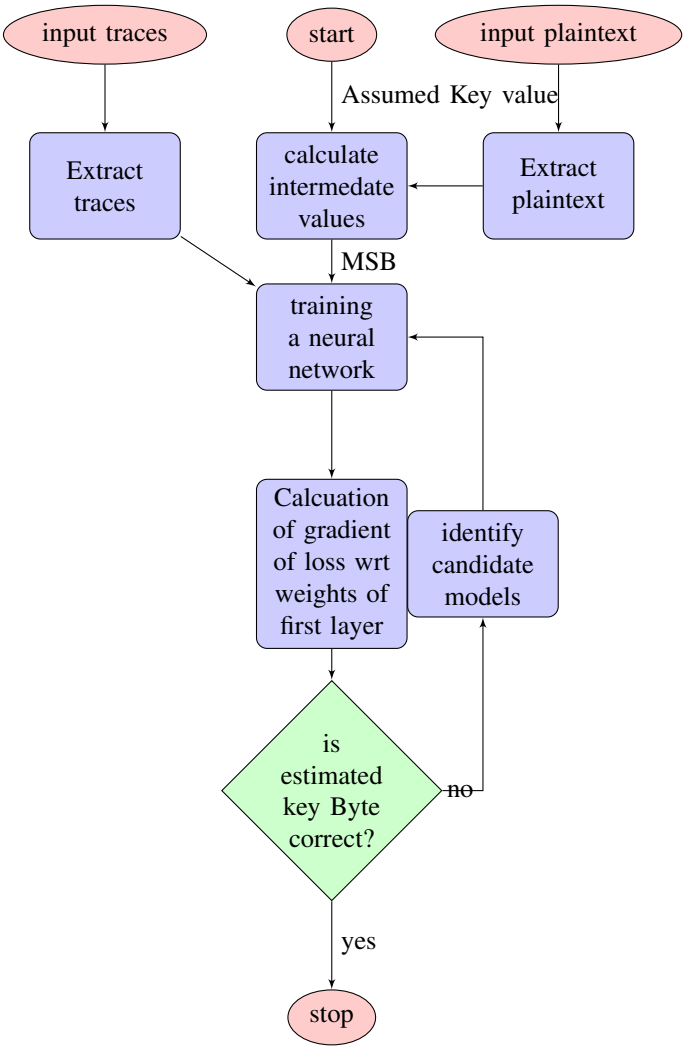

Fig. 1: Flow chart of Differential Deep Learning Analysis.

B. Differential Deep Learning Analysis (DDLA)

In this analysis, a deep neural network is developed and trained, and based on the DL metrics the secret key is revealed. The input traces are given to the network which act as training data. MSB of the intermediate value calculated is given as the training label for the network. For each key value ranging

from 0 to 255, the network is trained and the key value for which the network gives the best metrics is the desired key value. The byte number of the key revealed is the byte of the key value being used to calculate the MSB of the intermediate value.

DDLA has been done with the inputs being the power traces and the MSB of the intermediate values. A python code has been written and using keras and tensorflow libraries a multi-layer perceptron NN has been developed using Google Collaboratory platform, where the byte number of the MSB of the intermediate values is the byte number of the secret key used. The training data being the power traces have been taken in array format from the correlation power analysis performed. The training labels being the MSB values of the intermediate values are also taken from the analysis done.

III. Procedural Analysis and Discussion

A few steps for calculating average loss gradient values and loss values with the aid of mean squared error statistical metric, calculated by taking the traces in mathworks implementation as training data and MSB values of the intermediate values as training labels to reveal the key, are discussed briefly in this section. Fig. 1 shows a flowchart of the procedural analysis done in various steps.

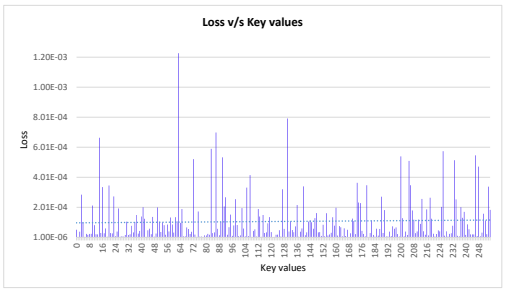

Fig. 2: Loss v/s Key values obtained during Differential Deep learning Analysis.

TABLE II: Hyperparameter setting with necessary values

Optimizer	*Learning rate (alpha)*	*No. of epochs*	*Layers*	*Loss*
Adamax	0.01	50	8 hidden layers + 1 input layer + 1 Output layer	Mean Squared Error

Fig. 2 shows the minimum loss values obtained during back propagation step, performed while training the NN with 50 epochs versus the key byte values ranging from 0 to 255. The key byte value for which the loss value is minimum is the desired key byte. The essential hyper parameter values for calculating loss gradient with an optimizer can be viewed in Table II.

Fig. 3 presents the average of gradients of weights of the first layer obtained during the back propagation step, performed while training the neural network versus the key byte

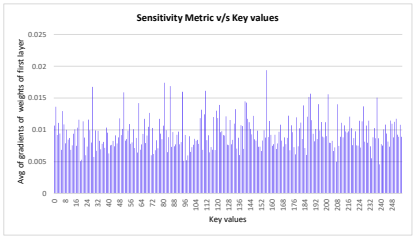

Fig. 3: Avg .of gradients of weights of first layer v/s Key values obtained during Differential Deep learning Analysis.

values ranging from 0 to 255. The correlation power analysis was done on the traces taken from [16] which were collected with an input and secret key while performing AES algorithm. The average loss gradient values and the loss values calculated by taking the traces in the MATLAB implementation as the training data and the MSB values of the intermediate values as the training labels. They were plotted across the corresponding key values ranging from 0 to 255. This depicts that the last byte of the key is 121 in decimal which means 79 in hexadecimal.

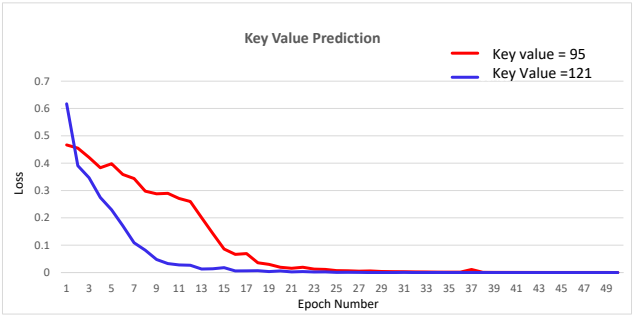

Fig. 4: Loss v/s Epoch number for key values 121 and 95 (decimal notation) .

Fig. 4 depicts the loss values during each epoch for the key 121 and key 95 in decimal notation, plotted with respect to the epochs. It was observed that the loss values approached the minimum value at best for key 121, while for other key values the loss change with respect to epochs was not as desired. From the loss values and the sensitivity metric obtained, the key value of 121 in decimal was less, with higher value of the sum of the gradients of weights of first layer.

The Table III draws a brief comparison to the related work done in the domain of deep learning based Side channel analysis with the proposed research. In [13], the algorithm used for attacking is Elliptic Curve Cryptographic algorithm where the power analysis attacks have been done using machine learning classification algorithms namely, Support Vector Machine, Random Forest, naive Bayes and Multilayer perceptron. They concluded that the accuracy of finding the secret data is high while using the machine learning algorithms, and MLP architecture needs more number of traces to be more accurate. In [14], the attack was done on PRESENT ultra-weight algorithm and the side channel analysis has been

TABLE III: Comparison with profiled and non-profiled attacks

	Algorithms	*Non-Profiled*	*Technique*	*Methods*
Our Proposal	AES 128	✓	Deep Learning	DDLA, MLP
[13]	ECC	✗	Machine Learning	Pre-processing SVM, MLP
[14]	PRESENT	✗	Supervised Learning	DL-LA, MLP, CNN
[15]	AES	✓	Deep Learning	DDLA, MLP, CNN
[27]	AES - 128	✓	Deep Learning	Preprocessing with Auto-encoders

done using deep learning leakage assessment (DL-LA). They have worked on aligned and misaligned traces to reveal the data and found that DL-LA needs less number of traces compared to that of the others considered by them.

An algorithm proposed by Timon [15] based on deep learning technique has been implemented to reveal the secret key showing the task being done towards applying deep learning techniques in the Side channel analysis domain, in non-profiled phase. Another work [27] in the non-profiled attack environment used preprocessing with autoencoders to reduce noise, and improves SCA with DDLA. Comparatively, our effort carries out in continuation to Timon's work, a tandem model attack which uses 33.5 percent fewer traces on average than single-model DL-SCAs to recover the key.

IV. Conclusion & Future prospects

The main advantage of involving Deep learning is the ability to reveal data being masked, a countermeasure used for Side channel attacks. In this investigation, AES-128 algorithm was implemented and correlation power analysis was performed on the traces. Deep learning techniques were used for Side channel analysis to reveal secured information. With the research carried out, further investigation can be done to draw vital conclusion on the efficiency of an algorithm and the traces used. The algorithm can be used to reveal data from the traces which are not even aligned. Using Deep learning techniques, novel methods can be used to countermeasure data from the attack rather than misaligning the traces, in non-profiled phase.

References

[1] Hospodar, G., Gierlichs, B., De Mulder, E. et al. Machine learning in side-channel analysis: a first study. Journal of Cryptographic Engineering 1, 293, 2011. doi: 10.1007/s13389-011-0023-x.

[2] Owen Lo , William J. Buchanana and Douglas Carsonb, "Power analysis attacks on the AES- 128 S-box using differential power analysis (DPA) and correlation power analysis (CPA)," Journal of cyber security technology, 2017

[3] Mark Zhao and G. Edward Suh, "FPGA-Based Remote Power Side-Channel Attacks," IEEE Symposium on Security and Privacy, 2018.

[4] Sunghyum Jin, Suhri Kim, HeeSeok Kim Seokhie Hong, Recent advances in deep learning-based side-channel analysis ETRI Journal 2020-04-06, journal-article. doi: 10.4218/etrij.2019-0163

[5] Shijie Song, Kaiyan Chen and Yang Zhang, "Overview of Side Channel Cipher Analysis Based on Deep Learning ,"Journal of Physics,2019

[6] François Durvaux and Marc Durvaux, "SCA-Pitaya: Practical and Affordable Side Channel Attack Setupfor Power Leakage–Based Evaluations," Digital Threats: Research and Practice Vol. 1, No. 1, Article 3,2020.

[7] R. Gilmore, N. Hanley, and M. O'Neill, "Neural network based attack on a masked implementation of AES", in Proc. IEEE Int. Symp. Hardw. Orient. Secur. Trust (HOST), Washington, DC, USA, pp. 106–111, May 2015.

[8] Z. Martinasek and V. Zeman, "Innovative method of the power analysis", Radioengineering 2 , no. 2, 586–594, 2013

[9] C Rebeiro, M Mondal and D Mukhopadhyay, "Pinpointing Cache Timing Attacks on AES,"23rd International Conference on VLSI Design, 2010, pp. 306-311, doi: 10.1109/VLSI.Design.2010.29.

[10] Yu, Weize & Kose, Selcuk." A Lightweight Masked AES Implementation for Securing IoT Against CPA Attacks". IEEE Transactions on Circuits and Systems I:pp. 1-11.2017, doi: 10.1109/TCSI.2017.2702098.

[11] Sasdrich, Pascal, Bilgin Begül , Hutter Michael & Marson, Mark." Low-Latency Hardware Masking with Application to AES." IACR Transactions on Cryptographic Hardware and Embedded Systems.2020, 300-326. doi: 10.46586/tches.v2020.i2.300-326.

[12] H. Wang, S. Forsmark, M. Brisfors and E. Dubrova, "Multi-Source Training Deep-Learning Side-Channel Attacks," 2020 IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL), 2020, pp. 58-63, doi: 10.1109/ISMVL49045.2020.00-29.

[13] Mukhtar N, Mehrabi M A, Kong Y, Anjum A, " Machine Learning based Side Channel Evaluation of Elliptic Curve Cryptographic FPGA Processor," Applied Science 2019. doi: 10.3390/app9010064

[14] Moos, T., Wegener, F., Moradi, A. (2021). DL-LA: Deep Learning Leakage Assessment: A modern roadmap for SCA evaluations. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2021(3), 552–598

[15] Timon Benjamin, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis", IACR Transaction on Cryptography. Hardware Embedded Systems 2 ,no. 4, 107–131,2019.

[16] Side Channel Attack: Differential Power Analysis (DPA) on AES encryption algorithm to deduce secret keys: github.com/GaPhil/dpa.

[17] Maghrebi H, Portigliatti T, Prouff E, (2016) Breaking Cryptographic Implementations Using Deep Learning Techniques. In: Carlet C., Hasan M., Saraswat V. (eds) Security, Privacy, and Applied Cryptography Engineering. SPACE 2016. Lecture Notes in Computer Science, vol 10076. Springer, Cham. https://doi.org/10.1007/978-3-319-49445-6

[18] Ramezanpour, K.; Ampadu, P.; Diehl, W. SCAUL: Power Side-Channel Analysis with Unsupervised Learning. arXiv e-Prints 2020, arXiv:2001.05951

[19] Hettwer, B.; Gehrer, S.; Guneysu, T. Applications of machine learning techniques in side-channel attacks: A survey. J. Cryptogr. Eng. 2019

[20] Golder, A.; Das, D.; Danial, J.; Ghosh, S.; Sen, S.; Raychowdhury, A. Practical Approaches toward Deep-Learning-Based Cross-Device Power Side-Channel Attack. IEEE Trans. Very Large Scale Integr. (Vlsi) Syst. 2019, 27, 2720–2733

[21] Jin, S.; Kim, S.; Kim, H.; Hong, S. Recent advances in deep learning-based side-channel analysis. ETRI J. 2020, 42, 292–304

[22] Libang, Z.; Xinpeng, X.; Junfeng, F.; Zongyue, W.; Suying, W. Multi-label Deep Learning based Side Channel Attack. In Proceedings of the 2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), Piscataway, NJ, USA, 16–17 December 2019; p. 6

[23] Yu, W.; Chen, J. Deep learning-assisted and combined attack: A novel side-channel attack. Electron. Lett. 2018, 54, 1114–1116

[24] Wang, H.; Brisfors, M.; Forsmark, S.; Dubrova, E. How Diversity Affects Deep-Learning Side-Channel Attacks. In Proceedings of the 5th IEEE Nordic Circuits and Systems Conference, NORCAS 2019: NORCHIP and International Symposium of System-on-Chip, SoC 2019, Helsinki, Finland, 29–30 October 2019; IEEE Circuits and Systems Society (CAS). Tampere University: Tampere, Finland, 2019

[25] Honggang Yu, Haoqi Shan, Max Pano., and Yier Jin, "Cross-Device Profiled Side-Channel Attacks using Meta-Transfer Learning," Design Automation Conference (DAC), 2021

[26] Méndez Real, M.; Salvador, R. Physical Side-Channel Attacks on Embedded Neural Networks: A Survey. Preprints 2021, 11, 6790. https://doi.org/10.3390/app11156790

[27] D. Kwon, H. Kim and S. Hong, "Non-Profiled Deep Learning-Based Side-Channel Preprocessing With Autoencoders," in IEEE Access, vol. 9, pp. 57692-57703, 2021, doi: 10.1109/ACCESS.2021.3072653.