X-DeepSCA: Cross-Device Deep Learning Side Channel Attack*

Debayan Das[1], Anupam Golder[2], Josef Danial[1], Santosh Ghosh[3], Arijit Raychowdhury[2], Shreyas Sen[1]
[1]School of Electrical and Computer Engineering, Purdue University, USA
[2]School of Electrical and Computer Engineering, Georgia Institute of Technology, USA
[3]Intel Labs, Hillsboro, Oregon, USA
{das60,jdanial,shreyas}@purdue.edu,anupamgolder@gatech.edu,arijit.raychowdhury@ece.gatech.edu

Abstract

This article, for the first time, demonstrates Cross-device Deep Learning Side-Channel Attack (X-DeepSCA), achieving an accuracy of > 99.9%, even in presence of significantly higher inter-device variations compared to the inter-key variations. Augmenting traces captured from multiple devices for training and with proper choice of hyper-parameters, the proposed 256-class Deep Neural Network (DNN) learns accurately from the power side-channel leakage of an AES-128 target encryption engine, and an N-trace ($N \leq 10$) X-DeepSCA attack breaks different target devices within seconds compared to a few minutes for a correlational power analysis (CPA) attack, thereby increasing the threat surface for embedded devices significantly. Even for low SNR scenarios, the proposed X-DeepSCA attack achieves ~ 10× lower minimum traces to disclosure (MTD) compared to a traditional CPA.

CCS Concepts

• **Security and privacy → Embedded systems security**; **Side-channel analysis and countermeasures**.

Keywords

Side-channel Attacks, Profiling attacks, Cross-device Attack, Deep Learning, Neural Networks.

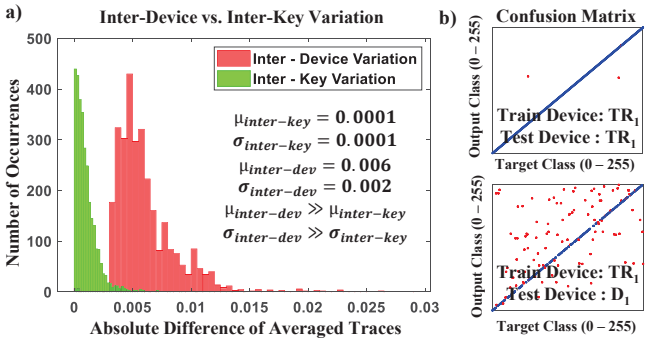

Figure 1: (a) Histogram plot showing that the mean of device-to-device variations of the power traces is significantly higher than the mean of key-to-key (class) variations for one device. 40 traces with 3000 time samples each were used in each case. (b) Training with one device (TR_1), the 256-class DNN is able to classify unseen test traces from the same device (TR_1) accurately as seen from the confusion matrix, while it does not generalize for other devices and misclassifies many test traces from a different device (D_1).

1 Introduction

In today's computing and communication systems, cryptographic algorithms are designed to provide integrity and confidentiality of data. The mathematical security of these implementations depend on the secrecy of a short *key*, which provides a computational advantage to the communicating parties over the adversary. Hence, a brute-force attack on these algorithms can only succeed with negligible probability. Side-channel analysis (SCA) is a form of cryptanalytic attack which breaks the secret key of an embedded device by utilizing the unintended 'side-channel' leakage emanating from the physical implementation of the cryptographic algorithm. These side-channel leakages can be obtained by monitoring the power consumption of the device running the algorithm [4, 5, 25], electromagnetic radiations [6, 16] during the cryptographic operations, processing time [3], cache hits/misses, and so on.

This article focuses on the power SCA attacks. Non-profiled power SCA attack techniques include differential and correlational power analysis (DPA/CPA), which have been utilized to break many real-world encryption devices [1, 9, 30]. Profiled power SCA attacks comprise of two stages: profiling and attack [2, 8, 27]. In the profiling phase, multiple traces from an identical device are collected by varying sub-keys (part of the cryptographic key), and a model is built. During the attack stage, the model is utilized to classify each sub-key of the device under attack.

In recent years, various machine-learning (ML) techniques have been evaluated to perform profiling power SCA attacks [13, 20, 29]. Although successful attacks have been shown, these ML techniques require pre-processing of the traces with proper time-alignment. In 2017, Cagli et al. [11] proposed a deep-learning based approach utilizing convolutional neural networks (CNNs) to provide an end-to-end profiling strategy, even in the presence of trace misalignments. Masking-based countermeasures were also shown to be broken using neural networks [15, 26]. Deep learning based SCA attacks does not require extensive statistical analysis to identify the points of leakage, in contrast to the template attacks. Also, as the dimensions of the data increase, ML SCA attacks become more prominent compared to the template attacks [20]. Deep Learning (DL) based SCA is still a new research paradigm [14] and all the

*This work was supported in part by the National Science Foundation (NSF) under Grant CNS 17-19235, and in part by Intel Corporation. Anupam Golder is also supported by the National Science Foundation (NSF) under Grant CNS 16-24810 - Center for Advanced Electronics through Machine Learning (CAEML) and its industry members.

DAC 2019, Las Vegas, NV, USA

ACM ISBN 978-1-4503-6725-7/19/06…$15.00
https://doi.org/10.1145/3316781.3317934

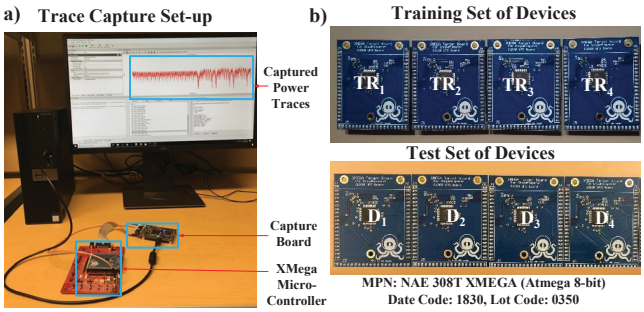

Figure 2: (a) Trace Capture Set-up using the Chipwhisperer platform. (b) Traces are captured from multiple Atmega microcontroller devices (TR_{1-4}) for training a DNN so that the model is able to generalize to any other target device (D_{1-4}).

previous works till date have focused on evaluating and improving the attack on the same device which has been used to train the neural network.

This work, for the first time, demonstrates a Cross-Device Deep Learning based Side-Channel Attack (X-DeepSCA) using a 256-class DNN. Figure 1(a) shows the measured cross-device variations in the form of a histogram (red plot) of the absolute difference between the samples at the same time index of the averaged traces from 2 different devices (TR_1, D_1) running the same software implementation of AES-128. For the device TR_1, the green curve shows the histogram of the variation between 2 different key bytes (classes). We see that the *inter-device variations for the same key are significantly higher than the inter-key variations of the same device*, which makes the cross-device attack particularly challenging. The confusion matrices in Figure 1(b) show that although the test accuracy on the same device (DNN trained with device TR_1 and tested with unseen traces from the same device) is very high (red dots represent the misclassified key bytes), the accuracy on a different test device (D_1) is significantly lower. Hence, training with one profiling device overfits to that particular device leakage and may not be able to generalize well to other devices.

Hence, in this work, we augment traces from multiple profiling devices (Figure 2(b)) and build a DNN architecture to perform cross-device deep-learning based power side-channel analysis (X-DeepSCA) attack. In addition, we analyze the individual class (key byte) accuracies and demonstrate the practicality of an N-trace ($N \leq 10$) X-DeepSCA attack to achieve $> 99.9\%$ success of attack. Finally, we study the effect of varying SNR scenarios, and show that the X-DeepSCA attacks require $\sim 10\times$ lower number of traces to attack (minimum traces to disclosure: MTD) than the traditional correlation power analysis (CPA) attacks [10].

In summary, the key contributions of this work are:

- A combination of designing the appropriate 256-class DNN with proper choice of the hyperparameters to prevent overfitting, utilizing traces from multiple devices (TR_{1-4}) during training, coupled with the proposed N-trace attack leads to the first successful demonstration of a cross-device deep-learning SCA (X-DeepSCA) attack.
- Using the Keras library with a Tensorflow backend [21], we show that the single-trace X-DeepSCA attack using the DNN model achieves an average accuracy of $> 99.9\%$ for all the test devices (D_{1-4}) under attack using 200K total traces for the training (Sec. 3).
- Further, we investigate the individual class accuracies by introducing a measure of entropy, leading to the proposed N-trace X-DeepSCA attack to guarantee $> 99.9\%$ attack success with $N \leq 10$ encryptions (Sec. 4).
- Finally, we show that the X-DeepSCA attack performs $> 10\times$ better in terms of MTD, with different signal-to-noise ratio (SNR) scenarios, reducing the time of attack from minutes to seconds (Sec. 5).

Table 1: Overview of the Related Works on Profiling Attacks

Train/Test Scenario	Profiling SCA Attacks	Classifier	
Train and Test with the same Device	[11], [15], [17], [18], [19], [20], [26], [27], [29]	SVM, RF, FCN, CNN, Statistical TA	
Train with one device and Test on different Devices	[7], [22], [23]	PCA/LDA, MIA, Statistical TA	**No Cross-device Machine Learning SCA exists**
Training with multiple devices, test with different devices	[24]	Statistical TA	
	This Work*	**DNN**	

**First Cross-device Deep-Learning Side-Channel Attack*

2 BACKGROUND & RELATED WORK

Template-based profiling power SCA attacks are extremely powerful as they can potentially break the encryption key within a few encryption traces [7, 27]. Recently, machine learning (ML) based profiling attacks have been studied extensively [13, 17–20, 29]. These ML-based attacks use supervised learning models like the support vector machine (SVM), Self-Organizing Map (SOM) or Random Forest (RF) for classification.

Deep neural networks (DNNs) have generated significant interest in the recent years. It has been shown that the clock-jitter based countermeasures against power/EM SCA can be broken using a convolutional neural network (CNN) [11, 12, 14]. Also, masking based countermeasures have been shown to be broken with neural networks [15, 26].

A summary of the related works is shown in Table 1. Most of the existing works [11, 17–20, 26, 27, 29] on profiling attacks have tested their attack on the same device used for the template generation. [7, 22, 23] have evaluated cross-device template-based attacks (TA) using statistical multivariate analysis, Principal Component Analysis (PCA), Mutual Information Analysis (MIA) and Linear Discriminant Analysis (LDA). [24] showed a multi-device profiling using statistical TA.

However, none of the ML-based works have focused on the cross-device attacks yet. Also, the previous works based on neural networks (NNs) have evaluated their models with the same device used for training. We have seen in Figure 1(a), the inter-device variation is typically much higher than the inter-key (or inter-class) variations. Hence, a NN model evaluated against the same device may not necessarily work well on a different target device. This work shows the first cross-device profiling attack using a deep neural network (DNN).

To train a neural network, the typical leakage models used for the power consumption are the Hamming Weight (HW) model (9-class classification), and the identity (ID) model (256-class classification) [14]. In this work, we use the identity model for 256-class classification and train our DNN to learn the leakage information

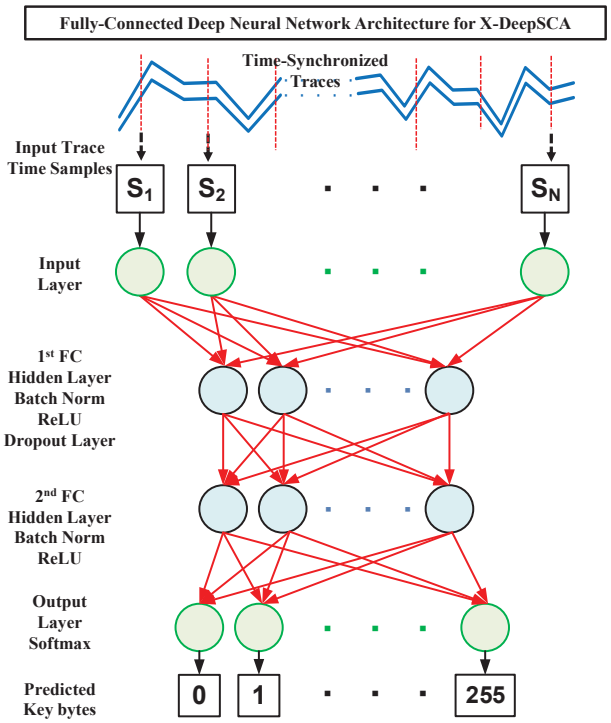

Figure 3: Architecture of the proposed Fully Connected DNN for X-DeepSCA. The input layer consists of N = 500 neurons. The 1^{st} fully-connected (FC) hidden layer consists of 200 hidden neurons, followed by Batch Normalization, Rectified Linear Unit (ReLU) activation, and a dropout layer. The 2^{nd} hidden layer is similar without the dropout layer. Finally, the output layer has 256 neurons for predicting the correct key byte utilizing the softmax function. If the traces are not aligned in time, a convolutional layer as the input layer would be required. In this work, we use the Fully Connected DNN as the traces captured from the Chipwhisperer are time-aligned.

accurately. For all the analyses shown in this work, the attacks are performed on the 1st key byte of the AES-128 encryption engine.

Also, most of the **previous NN models** have been **evaluated on the available DPA v4 contest dataset**, or the newly published ASCAD database [12] which, to the best of our knowledge, **do not contain traces from multiple devices**. Hence, to evaluate our cross-device attack, we built a new database by capturing traces from multiple devices using the Chipwhisperer platform (Figure 2(a)). Separate sets of Atmega microcontrollers (Figure 2(b)) running AES-128 are used for profiling and testing the X-DeepSCA attacks.

3 SINGLE TRACE X-DeepSCA ATTACK

In this section, we evaluate a single-trace X-DeepSCA attack. A 256-class classifier is necessary to perform a single-trace cross-device SCA attack (X-DeepSCA). However, designing a 256-class classifier is significantly more difficult compared to the HW-based 9-class classifier. Hence, choice of the hyperparameters like the learning

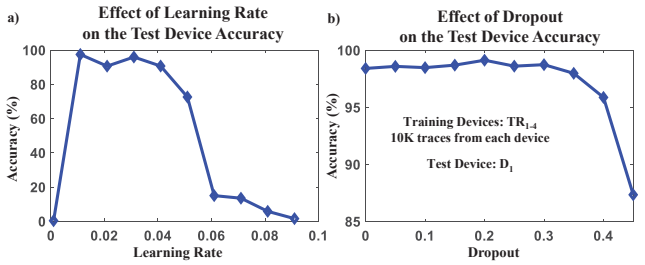

Figure 4: Effect of Model Hyper-parameters on the Test Accuracy (on the test device D_1 after training with TR_{1-4}): (a) Learning rate (LR) of ~ 0.01 provides the maximum test accuracy, and higher LR leads to overfitting of the DNN reducing the test accuracy. (b) Lower dropout shows higher accuracy which implies that the data gathered from the microcontroller devices has sufficient electronic noise which helps generalize to unseen data. Dropout higher than 0.3 reduces the accuracy.

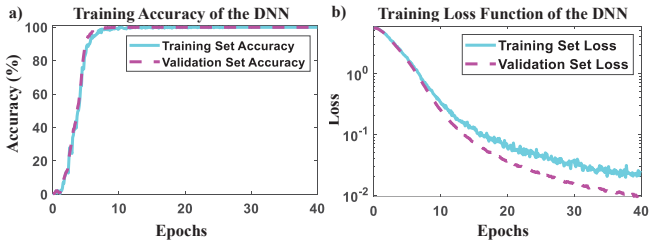

Figure 5: (a) Training and Validation Accuracy of the DNN reaches $\sim$ 100% within 25 epochs and does not show any overfitting. (b) Loss function of the DNN for both the training/validation sets. Note that training and validation have been performed with data from all the 4 devices (TR_{1-4}).

rate, number of hidden neurons, dropout, are extremely critical to prevent overfitting or underfitting.

3.1 DNN Architecture

Figure 3 shows the architecture of the proposed fully-connected (FC) DNN for the X-DeepSCA attack. Note that, for our work, the traces collected from the Chipwhisperer platform are time-synchronized and hence use of a convolutional layer is not necessary. Although the captured traces from the AES-128 encryption engine (clocked at 7.37 MHz) had 3000 time samples (ADC sampling frequency of 29.48 MHz) for an entire encryption operation, it was initially fed to the DNN and verified that the network learns accurately from the points of leakage (cross-verified using a CPA attack) within the first 200 time samples for the 1st key byte under attack. After this verification[1], to reduce the model complexity (and the time for training the DNN), only the first 500 time samples from each power trace were fed to the DNN.

The first FC layer of the DNN consists of 200 neurons, and increasing the number of hidden neurons may lead to overfitting. Batch normalization layer [28] and the dropout layers provide regularization to prevent overfitting and encourage generalization to

[1]It is also worth noting that the DNN model can also serve as a leakage assessment tool for cryptographic devices.

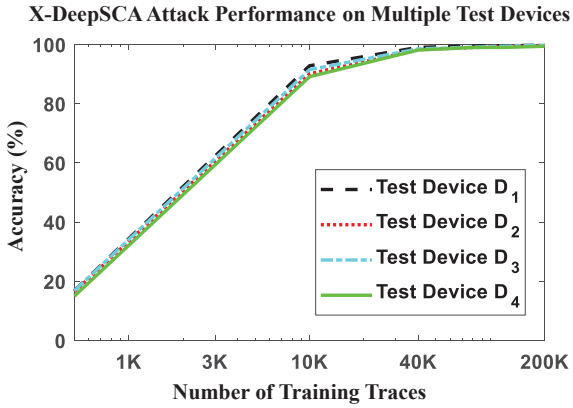

Figure 6: Attack Accuracy on the test devices (D_{1-4}) with the DNN model trained with varying number of training traces gathered from the 4 training devices, where each of them (TR_{1-4}) contributed equally .

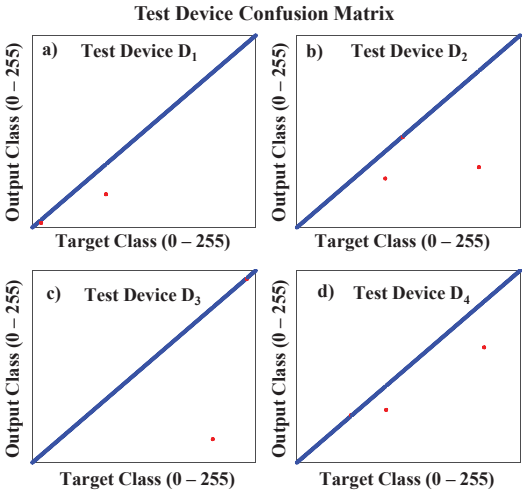

Figure 7: Confusion matrix for each of the test devices (D_{1-4}). At most 3 key bytes out of the 256 (~ 99% overall accuracy) are getting misclassified for each of the test devices, with the DNN model trained with 10K traces from each of the 4 training devices (TR_{1-4}).

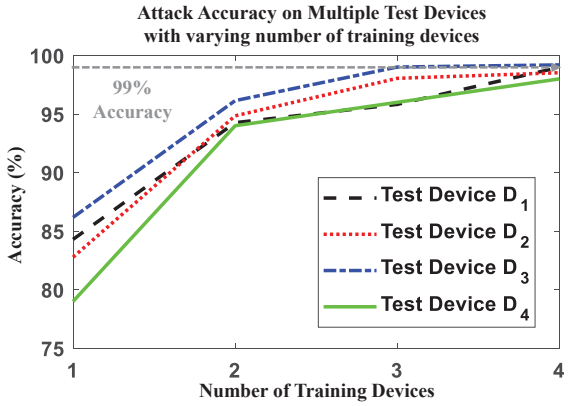

Figure 8: Effect of augmenting traces from Multiple Devices during training: As the number of devices is increased, the DNN model generalizes well to new devices (D_{1-4}) and hence the accuracy improves and reaches 99% with 4 training devices (TR_{1-4} - 10K traces each).

unseen data. The Rectified Linear Unit (ReLU) is used as the non-linear activation function to learn non-linear mappings from the input to the output. The second FC layer is similar without the dropout layer, and is finally followed by the output layer with 256 neurons, which predicts the correct key byte in a single trace utilizing the softmax function. The loss function used was categorical cross-entropy, optimized with the Adam algorithm and with a batch size of 32.

Figure 4(a, b) shows the effect of some of the hyper-parameters of the DNN model on the accuracy of a different test device. Figure 4(a) shows that a learning rate of 0.01 provides the maximum test accuracy, while a higher learning rate could lead to overfitting resulting in reduced test device accuracy. From Figure 4(b), we see that even in case of low dropout, the test accuracy remains high, which implies that the data gathered from the *real-world devices has sufficient electronic noise*. However, dropout more than 30% leads to reduced classification accuracy.

To train the DNN, for all our experiments (unless otherwise mentioned), 10K traces (equally distributed for all the 256 possible values for the 1^{st} key byte (classes) with a fixed plaintext) from each of the four devices were augmented together, and 20% of the total number of traces were kept for validation of the DNN during the profiling phase.

3.2 Performance Analysis of Single-Trace X-DeepSCA Attack

Figure 5(a,b) shows the training and validation accuracies of the DNN. We can see that the DNN model reaches an accuracy of > 99.9% within 25 epochs and also that the training and validation loss approach 0. The validation set accuracy remains almost same as that of the training set, implying that the DNN model is not overfitting. Note that the validation loss is lower since the dropout layer is present during training and not for the validation.

Figure 6 shows the performance of the trained DNN model on the test devices (D_{1-4}) with varying number of training traces, drawn equally from all the four devices (TR_{1-4}) reserved for training. The X-DeepSCA attack on all the 4 test devices shown reaches 99% accuracy with 40K training traces, and > 99.9% with 200K training traces in total(drawn equally from each of TR_{1-4}).

Note that for the test devices, traces are collected for different keys to evaluate the accuracy of all the classes (key bytes). Figure 7(a-d) shows the confusion plots on the test devices (D_{1-4}) after training with 40K traces (10K from each of the 4 training devices). As expected, for all the test devices, we see that at most 3 key bytes are misclassified (marked in red, outside the diagonal line) out of the all 256 different key bytes.

Figure 8 shows the effect of augmenting traces from multiple devices (with 10K traces each) for training the DNN. We see that with only 1 training device, the accuracy on a test device goes to ~ 80%, while it increases to ~ 99% after augmenting traces from

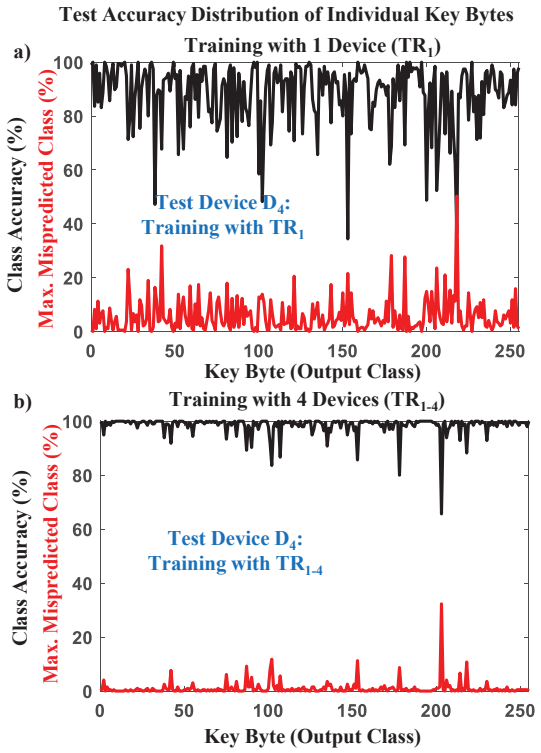

Figure 9: Individual Key Byte (Class) Accuracy Distribution for the test device D_4 (showed the worst average accuracy out of the D_{1-4}): The black plot represents the accuracy for each key byte, and the red plot denotes the maximum percentage of a particular class misprediction for the test device D_4. (a) With 1-device training (TR_1 - 10K traces), for some of the key bytes the black and red curves overlap, while (b) with 4-device (TR_{1-4} - 10K traces each) training, there is a significant reduction in the measure of entropy and an N-trace attack would be able to predict even the lowest accuracy key byte with high success probability (refer to Fig. 10).

all the 4 training devices with only 10K traces captured from each device.

4 N-TRACE X-DeepSCA ATTACK

In the previous section, we have shown that a single-trace X-DeepSCA attack with an accuracy of > 99.9% (averaged over all the 256 classes) can be performed on a test device, with 200K training traces (equally from each of the devices TR_{1-4}) used to build the DNN model. In this section, we analyze the individual class (key byte) accuracies to evaluate the practicality of a single-trace attack.

4.1 Individual Key Byte Accuracy

Figure 9(a, b) shows the individual key byte (class) accuracies and the percentage of the misclassified key byte with the highest occurrence in prediction (for every key byte class) for the test device D_4 (as it showed the lowest accuracy of the 4 test devices and poses the worst case scenario for an attacker). The separation between the class accuracy ($K_{pred} = K_{target}$) and the maximum percentage of the mispredicted class (the particular key byte which is wrongly

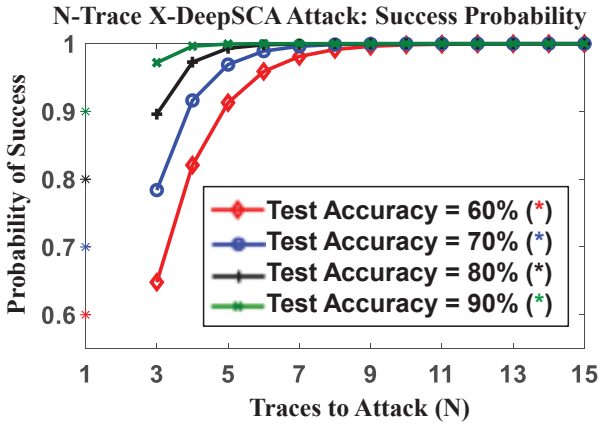

Figure 10: N-trace X-DeepSCA Attack: Number of traces required by an attacker to achieve a confidence of 99.9%. Even for classes with low accuracies, the N-trace X-DeepSCA attack would reveal the correct key byte within $N \leq 10$ traces, as long as the individual class accuracy remains higher than the % of maximum mispredicted key byte for that class.

predicted maximum number of times - $K_x \neq K_{target}$) gives a measure of the entropy ($\eta_{K_{target}}$) of the X-DeepSCA attack, as shown in Eqn. 1,

$$\eta_{K_{target}} = 1 - \left[\frac{|K_{pred} = K_{target}|}{|K_{total}|} - \frac{argmax(|K_{pred} = K_x| : K_x \neq K_{target})}{|K_{total}|} \right] \quad (1)$$

where, K_{pred} represents the predicted key byte, K_{target} is the target key byte, K_x is any other key byte (mispredicted) which has the maximum occurrence for the K_{target} class, and $|K_{total}|$ denotes the total number of queries (traces) for that particular K_{target} class.

Figure 9 shows that training with 4 devices has significantly lower entropy ($\eta_{K_{target}}$) compared to 1-device training. Also, we see from Figure 9 that although the test device D_4 achieves an average accuracy of > 99% (most of the key bytes can be broken with a single-trace), as seen from Figure 6, 8, the minimum accuracy of few key byte drops below 80%. Hence, although the single-trace attack will succeed on most key bytes, it may not work for a few key bytes, and a multi-trace attack is required.

4.2 Success Probability of the N-trace X-DeepSCA attack

Using the concept of majority voting, we propose an N-trace X-DeepSCA attack. The number of encryption traces required to gather in order to achieve a confidence (probability of success) of 99.9% can be mathematically derived, as shown in Eqn. 2 (valid for $N \geq 3$):

$$Pr(Maj(N) = K_{target}) = \sum_{x=2}^{N} Pr(x) = \sum_{x=2}^{N} \binom{N}{x} p^x (1-p)^{N-x} \frac{^{255}P_{N-x}}{255^{N-x}} \quad (2)$$

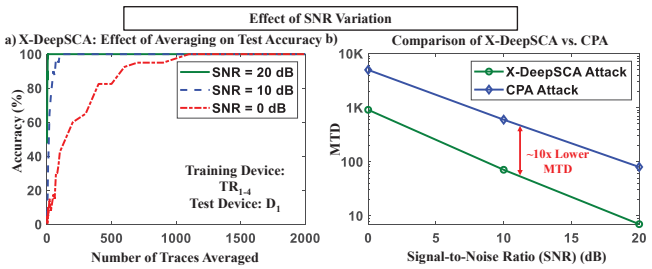

Figure 11: (a) Number of traces (averaged) required for a successful X-DeepSCA attack (with $>$ 99.9% accuracy) in different SNR scenarios. For SNR=20dB, averaging with less than 10 traces is sufficient to achieve $>$ 99.9% accuracy, while it requires $\sim$ 100 traces for SNR=10dB, and $\sim$ 1000 traces for SNR=0 dB to be averaged over to achieve the 99.9% accuracy. (b) Comparison of the X-DeepSCA attack with traditional CPA attack shows a lower MTD for X-DeepSCA for all SNRs.

where, $Pr(Maj(N) = K_{target})$ gives the probability of a successful target key recovery using the majority voting with N traces, p is the single-trace test accuracies for each class (key byte value), P represents the permutation operator. Note that the underlying assumption of Eqn. 2 is that the class accuracy and the class misprediction distributions are uniform, and there is no overlap between them for any of the individual key bytes. Hence, as seen from Figure 9(b), majority voting works as the entropy is reduced, and even for the lowest accuracy key byte (with 70% accuracy, $p = 0.7$), N-trace X-DeepSCA attack achieves an accuracy (success probability) of 99.9% with $N \leq 10$ encryptions, as shown in Figure 10 (derived from Eqn. 2).

5 DISCUSSIONS

5.1 X-DeepSCA Attack: Effect of SNR Variation

Now, we evaluate the effect of varying Signal-to-Noise Ratio (SNR) on the efficacy of the X-DeepSCA attack. Figure 11(a) shows the number of traces required to average for a successful X-DeepSCA attack with $>$ 99.9% accuracy on the test device D_1 using the training set with TR_{1-4} (10K traces each). Figure 11(b) shows that the number of traces required to retrieve the correct key byte of the AES-128 engine under attack is $\sim$ 10$\times$ lower than the traditional CPA attack (at the 1^{st} round S-box output using Hamming Weight(HW) leakage model) for different levels of SNR.

5.2 Future Work

For the future scope of this work, the efficacy of the proposed X-DeepSCA attacks can be further improved if we can guarantee that the accuracy of each key byte and the mispredicted classes for that key byte are uniformly distributed. This could be achieved by ensuring that the DNN has minimum bias during a misclassification, which would lower the number of traces (N) required for a successful N-trace X-DeepSCA attack. Overall, the proposed attack can put a huge dent to the security of embedded devices.

6 CONCLUSIONS

For the first time, this work shows a Cross-device Deep Learning based Side-Channel Analysis (X-DeepSCA) attack. Utilizing multiple (4) devices for training a fully-connected DNN, results showed that an average accuracy of 99.9% can be achieved with all the 4 test devices using 200K training traces, showing the possibility of a single-trace attack. However, deeper analysis utilizing the proposed measure of entropy revealed that few individual key bytes had lower accuracies, and hence an N-trace X-DeepSCA attack ($N \leq 10$) is proposed to break the key with $>$ 99.9% confidence. Finally, we show that for varying SNR scenarios, the proposed X-DeepSCA attack achieves $\sim$ 10$\times$ lower MTD, which breaks the target devices within seconds compared to a few minutes for the traditional CPA attack, increasing the threat surface significantly.

References

[1] A. Moradi et al. 2012. Black-Box Side-Channel Attacks Highlight the Importance of Countermeasures. In *CT-RSA 2012*. 1–18.
[2] C. Rechberger et al. 2005. Practical Template Attacks. In *Information Security Applications*. 440–456.
[3] D. Brumley et al. 2003. Remote Timing Attacks Are Practical. In *USENIX Symposium - Volume 12 (SSYM'03)*. Berkeley, CA, USA, 1–1.
[4] D. Das et al. 2017. High efficiency power side-channel attack immunity using noise injection in attenuated signature domain. In *2017 IEEE HOST*. 62–67.
[5] D. Das et al. 2018. ASNI: Attenuated Signature Noise Injection for Low-Overhead Power Side-Channel Attack Immunity. *IEEE TCAS-I* (2018), 1–12.
[6] D. Das et al. 2018. *STELLAR: A Generic EM Side-Channel Attack Protection through Ground-Up Root-cause Analysis*. Technical Report 620. https://eprint.iacr.org/2018/620
[7] D. Montminy et al. 2013. Improving cross-device attacks using zero-mean unit-variance normalization. *Journal of Cryptographic Engineering* 3, 2 (June 2013).
[8] D. Oswald et al. 2011. Breaking Mifare DESFire MF3ICD40: Power Analysis and Templates in the Real World. In *CHES 2011*. 207–222.
[9] D. Oswald et al. 2013. Side-Channel Attacks on the Yubikey 2 One-Time Password Generator. In *Research in Attacks, Intrusions, and Defenses*. 204–222.
[10] E. Brier et al. 2004. Correlation Power Analysis with a Leakage Model. In *CHES*. Number 3156. 16–29.
[11] E. Cagli et al. 2017. *Convolutional Neural Networks with Data Augmentation against Jitter-Based Countermeasures – Profiling Attacks without Pre-Processing –*. Technical Report 740. https://eprint.iacr.org/2017/740
[12] E. Prouff et al. 2018. *Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to ASCAD Database*. Technical Report 053.
[13] G. Hospodar et al. 2011. Machine learning in side-channel analysis: a first study. *Journal of Cryptographic Engineering* 1, 4 (Oct. 2011), 293.
[14] G. Perin et al. 2018. Lowering the Bar: Deep Learning for Side-Channel Analysis (White-Paper). *Blackhat 2018* (July 2018), 15.
[15] H. Maghrebi et al. 2016. *Breaking Cryptographic Implementations Using Deep Learning Techniques*. Technical Report 921.
[16] Karine Gandolfi et al. 2001. Electromagnetic Analysis: Concrete Results. In *CHES 2001*. 251–261. https://link.springer.com/chapter/10.1007/3-540-44709-1_21
[17] L. Lerman et al. 2013. A Time Series Approach for Profiling Attack. In *Security, Privacy, and Applied Cryptography Engineering*. 75–94.
[18] L. Lerman et al. 2014. Power Analysis Attack: An Approach Based on Machine Learning. *Int. J. Appl. Cryptol.* 3, 2 (June 2014), 97–115.
[19] L. Lerman et al. 2015. A machine learning approach against a masked AES. *Journal of Cryptographic Engineering* 5, 2 (June 2015), 123–139.
[20] L. Lerman et al. 2018. Template attacks versus machine learning revisited and the curse of dimensionality in side-channel analysis: extended version. *Journal of Cryptographic Engineering* 8, 4 (Nov. 2018), 301–313.
[21] M. Abadi et al. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium*. USENIX Association, Savannah, GA, 265–283.
[22] M. O. Choudary et al. 2018. Efficient, Portable Template Attacks. *IEEE TIFS* 13, 2 (Feb. 2018), 490–501.
[23] M. Renauld et al. 2011. A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In *EUROCRYPT 2011*. 109–128.
[24] N. Hanley et al. 2014. Empirical evaluation of multi-device profiling side-channel attacks. In *2014 IEEE Workshop on Signal Processing Systems (SiPS)*. 1–6.
[25] P. Kocher et al. 1999. Differential Power Analysis. In *CRYPTO' 99*. 388–397.
[26] R. Gilmore et al. 2015. Neural network based attack on a masked implementation of AES. In *2015 IEEE HOST*. 106–111.
[27] S. Chari et al. 2003. Template Attacks. In *CHES 2003*. 13–28.
[28] S. Ioffe et al. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]* (Feb. 2015).
[29] T. Bartkewitz et al. 2013. Efficient Template Attacks Based on Probabilistic Multi-class Support Vector Machines. In *Smart Card Research and Advanced Applications*. Springer Berlin Heidelberg, 263–276.
[30] T. Eisenbarth et al. 2008. On the Power of Power Analysis in the Real World: A Complete Break of the KeeLoq Code Hopping Scheme. In *CRYPTO 2008*. 203–220.