Cross-Device Profiled Side-Channel Attacks using Meta-Transfer Learning

Honggang Yu, Haoqi Shan, Maximillian Panoff, and Yier Jin
Department of Electrical and Computer Engineering, University of Florida
honggang.yu@ufl.edu, haoqi.shan@ufl.edu, m.panoff@ufl.edu, yier.jin@ece.ufl.edu

***Abstract*—Deep learning (DL) based profiling side channel analysis (SCA) pose a great threat to embedded devices. An adversary can break the target encryption engine through physical leakage of power or electromagnetic (EM) emanations collected from a profiling device. However, creating a successful DL based SCA model relies on a large amount of data. This presents a large barrier to those interested in applying DL for SCA. In this paper, we propose a novel attack mechanism that adopts *meta-transfer learning* to transfer DL networks among target devices by judiciously extracting information from a profiling device even using different side-channel sources. Supported by our method, a cross-device and/or cross-domain SCA attack becomes possible among different designs. In comparison to previous attack methodologies, we significantly reduce training costs and the number of traces (< 3 for power and < 8 for EM) required for SCA attacks on both unprotected or masked Advanced Encryption Standard (AES) implementations.**

I. Introduction

Profiled side channel analysis (SCA) are attacks typically targeting cryptographic circuits in which an adversary exploits vulnerabilities of hardware implementations to recover secret information, e.g., encryption keys. In order to apply the attack, an adversary first obtains a *profiling device* under their control and then attacks a *target device* using the same side-channel leakage. The profiling SCA is itself a type of Template Attack. Template attack generates a statistical model using the profiling device and applies this model to attack target devices [1], [2].

Recently, researchers enhanced the attack method by applying deep learning (DL) techniques, making the profiled attack more powerful. For instance, Maghrebi *et al.* [3] introduced an attack method that applies deep convolution neural networks (CNNs) to recover secret keys from either unprotected or masked Advanced Encryption Standard (AES) implementations. Picek *et al.* [4] further improved DL based profiled side channel attacks in the presence of imbalanced data. Kim *et al.* [5] demonstrated that adding noise to input power traces can help an adversary boost the performance of a DL model. However, their performance rapidly degrades if the training set, generated by traces from the profiling device, slightly deviates from the measurements from the target device.

To address this issue, researchers further proposed to leverage DL models to eliminate the cross-device variations. Das *et al* [6] trained the DL model on augmenting traces collected from multiple devices and used the model to break a target 128-bit AES encryption module. Bhasin *et al.* [7] introduced an attack method which considered the device variation during a profiled SCA attack and further evaluated the portability of the attack. Zhang *et al.* [8] proposed to utilize frequency and learning based power analysis to address the challenges caused by device variations. However, these DL based side channel attacks routinely depend on the availability of large amounts of training data, which presents a large barrier to those who are interested in using DL for SCA.

To address these challenges, one research direction is to reduce the required traces from a profiling device so that the training cost can be lowered. For example, Thapar *et al* [9] implemented a DL based SCA attack that uses transfer learning to reduce the number of profiling traces required to recover secret keys from the same or a different device. Similarly, Genevey-Metat *et al.* [10] used transfer learning to show that it can improve the efficiency of power and/or EM based SCA on various devices.

Motivated by these works, in this paper, we further extend DL based profiling attacks and propose a novel *cross-device* and *cross-domain* SCA attack. Specifically, we apply the concept of *meta-transfer learning* to transfer DL networks to the target device by judiciously extracting information from a profiling device. We call the new attack Meta-Transfer Learning based Side Channel Attack (MTL-SCA). Unlike traditional transfer learning methods, the proposed attack can make the pre-trained model of the profiling device easier to be adapted to a new target device. As a result, the proposed attack method can significantly reduce the training cost and the amount of traces from the target device. To the best of our knowledge, it is the first time to apply meta-transfer learning technique into side-channel attacks.

To summarize, we make the following contributions:

- We propose a novel *cross-device/domain* attack mechanism, named MTL-SCA, which leverages the advantages of both meta learning and transfer learning for the efficient SCA attacks from both unprotected or masked cryptographic circuits.
- We use a simple and efficient measure, known as *Pearson product-moment correlation coefficient (PPMCC)*, to evaluate the similarity across devices of the same and different types. Our MTL-SCA can be more effective if the inter-device PPMCC value is larger.
- We evaluate the proposed attack on a group of devices with different microprocessors. The experimental results demonstrate that the MTL-SCA method can recover the secret keys from AES implementations with as few

978-1-6654-3274-0/21/$31.00 ©2021 IEEE

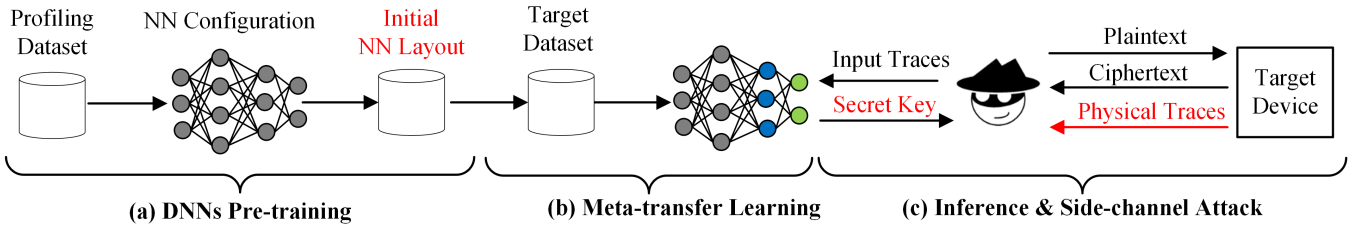


Figure 1: Overview of the proposed meta-transfer framework for profiled side-channel attacks: (a) Pre-training the DL model on the labelled profiling dataset; (b) Fine-tuning the trained DL model on target dataset using the meta-transfer learning; (c) Applying DL models on newly collected side channel traces from the target devices.

as 3 traces on the same type of microprocessors. For cross-device and cross-domain attacks, our method only requires a few hundreds of traces.

II. Preliminaries

A. Deep Neural Network

A typical Deep neural network (DNN) model consists of a cascade of multiple computational layers such as convolution, pooling and fully-connected layers, which learn to perform automatic data processing and transformation. Each hidden layer of deep neural networks has a set of neurons. These neurons are usually connected to previous hidden layer and utilize activation functions such as sigmoid, tanh and rectified linear unit (ReLU) to transform its input data into output vectors. The use of DNN models often includes two phases, *training* and *inference*. During the *training* phase, the model's parameters (e.g., weights) in each hidden layer are initialized at the beginning and then are optimized based on the training dataset. In this paper, the training dataset includes power and EM traces. During the *inference* phase, the pre-trained model is utilized to make predictions on new input data.

B. Profiled Side-channel Attack

Profiled side-channel attacks have posed a great threat to embedded devices. Adversaries assume that they have full access to an identical copy of the target device running the same cryptographic algorithms. The adversaries can collect the physical side channel leakages (e.g., power or EM) while providing the copy device with any set of plaintext and chosen keys. These captured traces can then be utilized to train DNN models. Adversaries can then use these trained models to recover secret keys from target devices. Under similar circumstances, an adversary hopes to only need one trace from the target device for efficient key recovery. However, device variation usually makes it difficult for an adversary to conduct such attacks [8]. An adversary usually need tens to thousands, if not millions, of traces to break the target device.

C. Meta-transfer Learning

Meta-transfer learning aims to leverage the advantages of both transfer learning and meta learning to boost the performance of deep neural network models [11]–[13]. Specifically, meta learning is a novel method that enables deep neural network models to converge faster so that the model can quickly learn a new task with much less labelled data [14]. Transfer learning applies the knowledge gained in a *source domain* to other different but related *target domains*. Recently, researchers have investigated the transferability of convolutional layers in DNNs and demonstrated that fine-tuning these hidden layers trained on large-scale datasets from the source domain can be utilized to learn the features of target domain with much less labelled data [15]–[19].

III. Methodology

Recent DL based cross-device attacks require a large amount of traces for training the DL model, which makes it difficult to recover secret keys from target devices, especially in real-world scenarios. To enhance existing method, we propose to apply the meta-transfer learning methodology to reveal the crucial information from the profiling device and transfer them to better recover the secret keys of the target devices. To the best of our knowledge, such a method has not been explored in side channel attacks. This novel method is named as Meta-transfer Learning based Side Channel Attacks (MTL-SCA). The overall scheme of the proposed MTL-SCA is shown in Figure 1 and the working process is presented in Algorithm 1.

The developed MTL-SCA mainly involves two stages, DNN model pre-training and meta-transfer learning. In the pre-training stage, we obtain a good initial parameters of the DNN model. Then in the meta-transfer learning stage, we will fine-tune the pre-trained model on the target dataset. Finally, we apply the trained model to recover the secret keys from the target device. Since the proposed attack exploits the advantages of both meta learning and transfer learning, our attack can break the target AES-128 device with fewer traces and lower training costs simultaneously when compared to previous works.

A. DNN Pre-training

This stage is similar to the classic DNN training phase. That is, we randomly sample an input/output pairs from the source task $\mathcal{T}_i$ and initialize the parameters θ (e.g., weights) in DNNs classifier f_θ. During the pre-training stage, the model parameters θ are optimized by minimizing the cross-entropy loss $\mathcal{L}_{\mathcal{T}_i}$ on the dataset D_i.

$$\mathcal{L}_{\mathcal{T}_i}^{D_i}\left(f_\theta\right) = \sum_{\mathbf{x}(j),\mathbf{y}(j)\sim\mathcal{T}_i} l\left(f_\theta(x), y\right) \tag{1}$$

In the proposed attack, we use physical side channel leakages including power and EM traces collected from the profiling device to pre-train the DNNs model. The resulting pre-trained parameters will serve as good initialization of the DNNs model and can help us solve the instability problem caused by the meta-learning as mentioned in [11], [14].

B. Meta-transfer Learning based SCA

Our meta-transfer learning method uses the information captured from multiple source tasks to train the target task and optimize the model's parameters effectively. Formally, we consider a classifier $f(\theta)$ with corresponding parameters θ. For each adaptation step on task $\mathcal{T}_i$, we optimize the parameters in our DNN model using a gradient based learning algorithm as follows:

$$\theta'_{\mathcal{T}_i} \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\theta) \tag{2}$$

where α is the task-level learning rate. In our method, we automatically update the parameter α using the meta-learning algorithm.

The DNN model is optimized by achieving the minimal error over the dataset from target devices. The meta-objective is defined as follows:

$$\arg\min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(\theta'_{\mathcal{T}_i}\right) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(\theta - \alpha \nabla \mathcal{L}_{\mathcal{T}_i}(\theta)\right) \tag{3}$$

where $p(\mathcal{T}$ is the task distributions.

Here we perform the meta-transfer learning which enables to transfer knowledge from source tasks to a specific target task. Note that the parameter θ can be updated using stochastic gradient descent (SGD) as follows:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}}(\theta'_{\mathcal{T}_i}) \tag{4}$$

C. Evaluation Metric

We use the guessing entropy over the test set to evaluate the performance of our MTL-SCA mechanism [5], [20]. Given a random input vector $T = [t_1, t_2, ..., t_a]$ in the attack phase, the size of the key space $|\mathcal{K}|$, the estimated probability $\hat{p}_{ij}$ for key candidates, a key guessing output can be described by the vector $g = [g_1, g_2, ..., g_{|\mathcal{K}|}]$, where g_i is given by using the following log-likelihood function:

$$g_i = \sum_{j=1}^{a} \log\left(\hat{p}_{ij}\right) \tag{5}$$

The guessing entropy in SCA can be finally computed by the average position of the secret key k^* over the test set.

IV. Experimental Results

A. Experimental Setup

We conduct experiments on AES software implementations on different microprocessors. Table I lists all experimental devices used to validate the proposed attack method. In total, we collect power and EM traces on 5 different microprocessors running the same AES encryption algorithm. We utilize

Algorithm 1 MTL-SCA: for distribution $p(\mathcal{T})$ over tasks $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_i\}$, the DNNs classifier f with parameter θ, learning rate α and β

Input: $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_i\}$, α and β
Output: Model parameters θ_M

```
//DNNs model pre-training on the source task
Randomly initialize parameter θ
Generate subset D_t from the labeled dataset D
for all D_i do
    Sample D_i = (traces, labels) batch from dataset D_t
    Update parameter θ in Eq. (1) using SGD
end for
// Meta-trasfer learning on the target task
while not done do
    Sample task batch T_i ~ p(T)
    for all batch p(T) do
        Evaluate loss ∇_θ L_{T_i(θ)} on sampled tasks
        Optimize parameter θ'_{T_i} ← θ − α∇_θ L_{T_i}(θ)
    end for
    Update parameter θ ← θ − β∇_θ Σ_{T_i~p(T)} L_T(θ'_{T_i})
end while
```

Chipwhisperer UFO target board [21] as the side channel trace acquisition platform.

This acquisition platform runs at fixed clock frequency of 7.37MHz for all our microprocessors under test. We customize the TinyAESC [22], an open-source portable version AES encryption/decryption library designed for microprocessor with reduced RAM and ROM usage. It is worth noting that the compilers may eventually generate divergent assembly code due to the architecture variation. To eliminate the influence of compilers, we inspect the binary code with reverse engineering tool and examine if the encryption procedures follow similar control flow and data flow. Further, we limit the acquisition window on specified instructions during AES encryption and a fixed key is used for our experiments. With this experimental setup, we can focus on the hardware and architecture differences while performing MTL-SCA attacks.

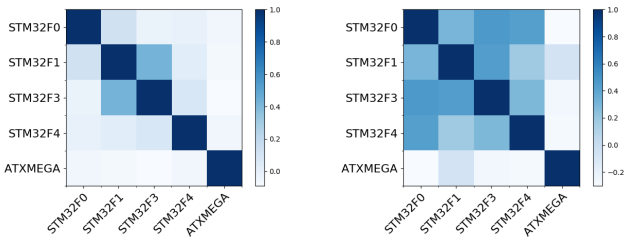
All data analyses are conducted on a server of Intel Xeon(R) E5-2623 v4 2.60GHz CPU, 128GB RAM, Ubuntu 18.04, accelerated by NVIDIA Tesla V100 GPU. The neural network architecture of the MTL-SCA consists of two convolution layers and four fully-connected layers. We use ReLU as the activation function for input and hidden layers. The DNN model is trained for 50 epochs with a batch size of 50.

B. Power and EM Trace Collection

To align all acquired traces, a trigger signal is set on a dedicated GPIO port of the target board at the beginning of each AES encryption. The trigger signal remains high during the entire AES encryption and will be pulled down once the encryption is finished. Among the entire AES encryption procedure, we only set the acquisition window into the first byte substitution layer of AES encryption. More specifically, the $sbox(k[0] \oplus p[0])$ operation occurs at fixed location on same

Dataset Abbr.	Platform	Chip Model	ISA	#Features (POIs)	#Traces
STM32F0	32-Bit Microprocessor	STM32F071RBT6	ARM Cotex-M0	700	60000
STM32F1	32-Bit Microprocessor	STM32F100RBT6	ARM Cortex-M3	700	100000
STM32F3	32-Bit Microprocessor	STM32F303RCT7	ARM Cortex-M4	700	100000
STM32F4	32-Bit Microprocessor	STM32F405RGT6	ARM Cortex-M4	700	50000
ATXMEGA	8-Bit Microcontroller	ATXMEGA128D4	AVRxm	700	50000

Table I: Profiling side channel leakage dataset (power and EM)



(a) Power (b) EM

Figure 2: Evaluation of device variations for ST and AVR microprocessors using different side-channel sources.

type of microprocessors as we apply the same clock frequency and same source code. We utilize binary analysis technique to calculate when the $sbox$ operation happens and use this time value as the acquisition offset along with the trigger signal. Therefore, we can narrow down our acquisition window and only record power and EM traces related to the operation $sbox(k[0] \oplus p[0])$ to achieve the maximum sampling rate[1]. During our experiment, Keysight MSOX4154A is used as the testing equipment with a maximum sampling rate of 2.5GSa/s. We utilize Keysight N2894A probe and Langer LF-3 probe to capture the power consumption and EM emanations from the microprocessors running AES encryption, respectively. For each acquisition, 700 data points, also called features or point of interests (POIs), of each power/EM trace are collected.

We evaluate the device variation based on these captured side channel traces through Pearson product-moment correlation coefficient (PPMCC). In our experiments, 10,000 traces for each device are collected and are used to compute the correlation coefficient. The experimental results of similarity among different types of cross devices are shown in Figure 2. Based on the quantitative analysis results, we observe that: (i) The PPMCC for the device from the same manufacturer (e.g., STM32Fx) are relatively large, illustrating that an adversary can easily recover secret keys using profiling attacks across different devices (see Figure 2 (a)). (ii) The PPMCC between devices of different architectures, e.g., STM32Fx and ATXMEGA, are much smaller, indicating that it will be challenging for an adversary to apply profiling attacks in this scenario. The similar tend can also be found in Figure 2 (b).

[1]The setting of the short acquisition window is to overcome the storage limits of commercial oscilloscopes at high sample rates.

Method	Pre-train	Fine-tune	Pre-processing	Model
DL-SCA [3]	✓	✗	✗	DNN
FL-SCA [8]	✓	✗	FFT	DNN
TL-SCA [10]	✓	✓ (TL)	✗	DNN
MTL-SCA	✓	✓ (MTL)	✗	DNN

Table II: Comparison to related work. TL - Transfer Learning, MTL - Meta-Transfer Learning, FFT - Fast Fourier Transform.

C. Case Study 1: Cross-Device Power MTL-SCA

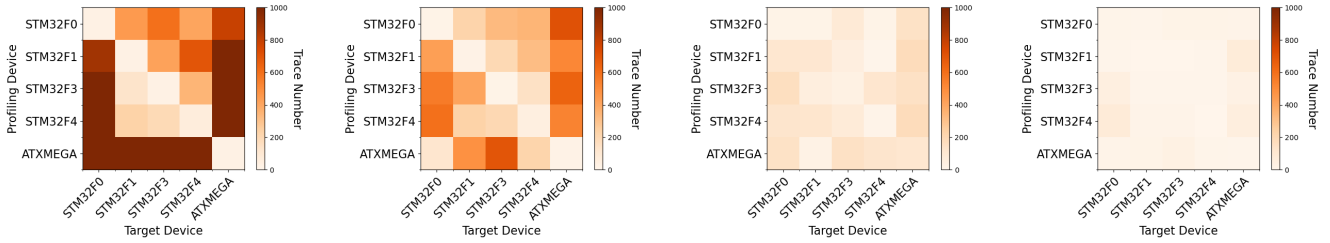
To evaluate the effectiveness of the proposed attack on the collected power traces (see Table I), we compare the attack efficacy of MTL-SCA with existing DL based attacks, including DL-SCA [3], FL-SCA [8] and TL-SCA [10]. A brief comparison between our attack and these previous attacks is shown in Table II. As we can see from the table, all four methods leverage DNN and require pre-training stages. Only TL-SCA and MTL-SCA has the fine-tune capability. As we will see soon, the MTL based fine-tuning outperforms the TL based fine-tuning.

In our experiments, we keep the DNN model architectures fixed, which means only one unified model is trained on the collected traces from the profiling devices. This model is then used to exploit all target devices. During the training stage, we randomly select 20,000 traces from the collected profiling device dataset for pre-training and 800 traces from the target device for fine-tuning using the MTL method. In the evaluation stage, we capture 10,000 traces from the target device to evaluate the effectiveness of different DL based SCA methods. Figure 3 shows the experimental results in recovering the first byte of the AES-128 key. We observe that: (i) For cross-device attacks with the same architecture, MTL-SCA can recover the secret key from the target devices using as less as 3 traces, a much better result compared to DL-SCA, FL-SCA and TL-SCA attacks. (ii) For cross-device attacks with different architectures, the DNN models fine-tuned by the MTL method can converge towards guessing entropy 0 within 40 traces, much better than previous attacks.

Further, we also observed that the proposed MTL-SCA requires much less side channel traces from the profiling device for pre-training (by 35% on average). These experimental results further demonstrate that by leveraging the advantages of both transfer learning and meta learning, our attack can converge faster while reducing the probability of overfitting.

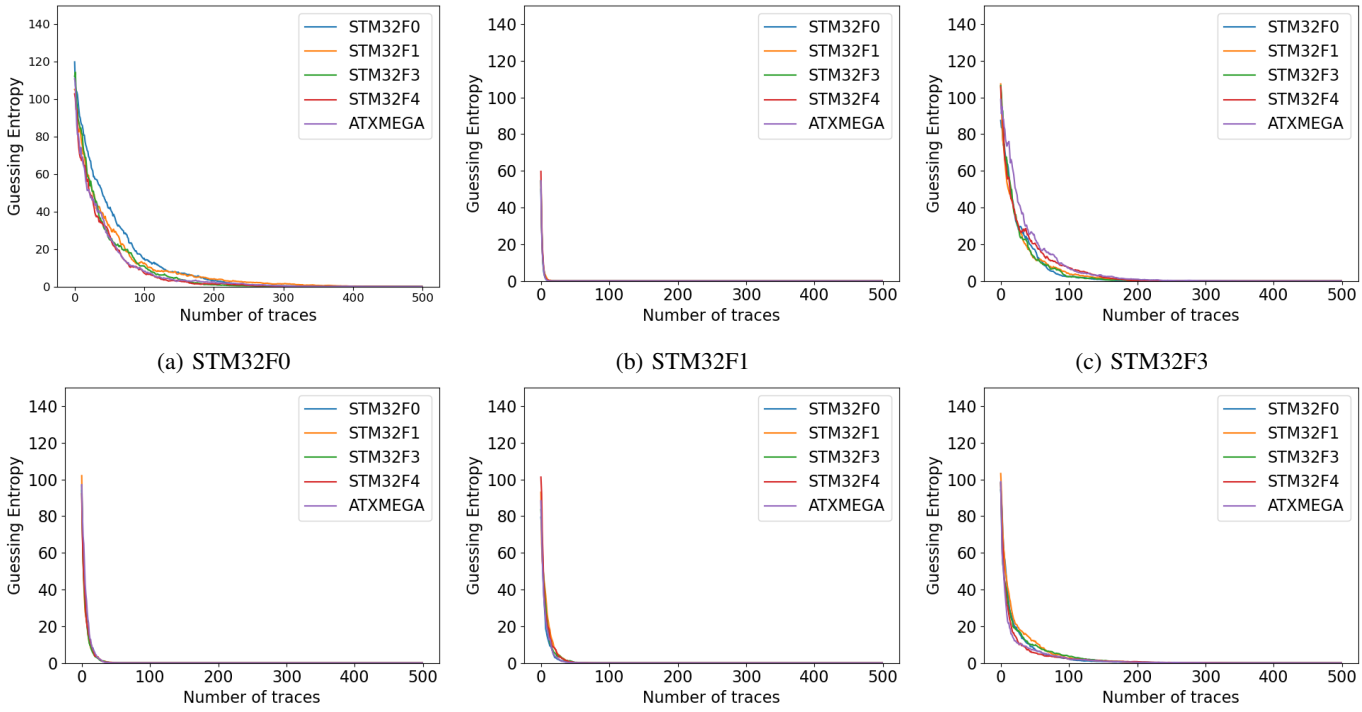
D. Case Study 2: Cross-Device EM MTL-SCA

In this section, we consider another scenario in which an adversary may only collect EM traces from the target device.



(a) DL-SCA (b) FL-SCA (c) TL-SCA (d) MTL-SCA

Figure 3: Comparisons of different cross-device side channel attacks.



(a) STM32F0 (b) STM32F1 (c) STM32F3

(d) STM32F4 (e) ATXMEGA (f) ASCAD

Figure 4: Experimental results of MTL-SCA on cross-device cross-domain attacks among different devices.

Our goal is to evaluate the effectiveness of MTL-SCA on EM traces which often have lower signal-to-noise ration. In our experiments, the DNN model is trained on the EM profiling dataset with 50,000 EM traces. Another 10,000 EM traces are collected from the target device as the testing dataset.

Similar to power side channel attacks, we implement cross-device EM side-channel attacks. Our initial experiment uses 20,000 EM traces but the DNN model cannot converge towards guessing entropy 0. By increasing the training dataset to 50,000 traces, the DNN model can recover the secret key if fine-tuned by 1,000 EM traces from the target device of the same architecture. If the target device is of different architectures, we need 1,500 EM traces to fine-tune the model. The experimentation results demonstrate that the additional noise on EM signals will influence the key guess accuracy but our method can still recover the key with much less traces.

E. Case Study 3: Cross-Device Cross-Domain MTL-SCA

To further improve the feasibility of the proposed attack, we consider a stronger threat model here. That is, the attackers may only collect noisy EM traces from the target devices but they can build a DNN model with power traces from a profiling device. We name the new attack as *cross-device and cross-domain* side-channel attack. Specifically, an adversary can pre-train the model on the power traces from the profiling device and then utilize the meta-transfer learning to fine-tune the model on *noisy* EM traces from the target device. For each target device, we try 5 different profiling devices to train the DNN model. We use 50,000 power traces for pre-training and 20,000 EM traces for fine-tuning the DNN model.

The experimental results are shown in Figure 4. We observe that: (1) For the collected datasets (see Figure 4 (a), (b), (c), (d) and (e)), our cross-device cross-domain attack can successfully recover the secret key of the target device within less than

200 EM traces (in many cases, less than 50 EM traces are needed). Take the STM32F1 device as an example, our DNNs model can break the AES key using only as few as 8 EM traces. Again, even with low signal-noise-ratio (SNR), our method can still break the AES key within a few hundred EM traces. (2) We also train our DNN model on a public masked ASCAD dataset [23] (see Figure 4 (f)) for the cross-device cross-domain attacks. Our results show that we need less than 230 EM traces to perform the side channel attacks. These experimental results further show that it is possible to implement the *cross-device* and *cross-domain* MTL-SCA. An adversary can use the proposed MTL-SCA framework to recover the secret keys even using *noisy* side-channel signals (i.e., low SNR EM traces) from target device but still achieving high attack capability.

V. Conclusions

DL based profiling side-channel attack have posed a great threat to embedded devices. An attacker can capture physical side channel leakages from a profiling device and generate the dataset for a DL model. In this paper, we present an even more powerful attack that uses *meta-transfer learning* to transfer DL networks between target devices by judiciously extracting information from a profiling device even using different side-channel sources. A new cross-device cross-domain attack was presented. Compared to previous attack methodologies, our method significantly reduces the training cost and the amount of traces from target devices for the efficient SCA attacks. In the future, we will focus on effective defense mechanisms against DL based SCA, and therefore enhance the robustness of cryptographic circuits.

Acknowledgement

This material is based on research sponsored by Intel Corp., Air Force Research Lab (AFRL) under agreement number FA8650-19-1-1741, National Institute of Standards and Technology (NIST) and Office of Naval Research (ONR) Young Investigator Program (YIP). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Intel Corp., Air Force Research Lab (AFRL), National Institute of Standards and Technology (NIST), Office of Naval Research (ONR) or the U.S. Government.

References

[1] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002*, B. S. Kaliski, ç. K. Koç, and C. Paar, Eds. Springer Berlin Heidelberg, 2003, pp. 13–28.

[2] C. Archambeau, E. Peeters, F. X. Standaert, and J. J. Quisquater, "Template attacks in principal subspaces," in *Proceedings of the 8th International Conference on Cryptographic Hardware and Embedded Systems*, ser. CHES'06. Springer-Verlag, 2006, p. 1–14.

[3] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *Security, Privacy, and Applied Cryptography Engineering*, C. Carlet, M. A. Hasan, and V. Saraswat, Eds. Cham: Springer International Publishing, 2016, pp. 3–26.

[4] S. Picek, A. Heuser, A. Jovic, S. Bhasin, and F. Regazzoni, "The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, 2018.

[5] J. Kim, S. Picek, A. Heuser, S. Bhasin, and A. Hanjalic, "Make some noise. unleashing the power of convolutional neural networks for profiled side-channel analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 3, pp. 148–179, 2019.

[6] D. Das, A. Golder, J. Danial, S. Ghosh, A. Raychowdhury, and S. Sen, "X-deepsca: Cross-device deep learning side channel attack*," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.

[7] S. Bhasin, A. Chattopadhyay, A. Heuser, D. Jap, S. Picek, and R. R. Shrivastwa, "Mind the portability: A warriors guide through realistic profiled side-channel analysis," Cryptology ePrint Archive, Report 2019/661, 2019, https://eprint.iacr.org/2019/661.

[8] F. Zhang, B. Shao, G. Xu, B. Yang, Z. Yang, Z. Qin, and K. Ren, "From homogeneous to heterogeneous: Leveraging deep learning based power analysis across devices," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[9] D. Thapar, M. Alam, and D. Mukhopadhyay, "Transca: Cross-family profiled side-channel attacks using transfer learning on deep neural networks," Cryptology ePrint Archive, Report 2020/1258, 2020, https://eprint.iacr.org/2020/1258.

[10] C. Genevey-Metat, B. Gérard, and A. Heuser, "On what to learn: Train or adapt a deeply learned profile?" Cryptology ePrint Archive, Report 2020/952, 2020, https://eprint.iacr.org/2020/952.

[11] J. W. Soh, S. Cho, and N. I. Cho, "Meta-transfer learning for zero-shot super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[12] G. I. Winata, S. Cahyawijaya, Z. Lin, Z. Liu, P. Xu, and P. Fung, "Meta-transfer learning for code-switched speech recognition," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 3770–3776. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.348

[13] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 2017, pp. 1126–1135.

[15] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2015, pp. 36–45.

[16] W. Ge and Y. Yu, "Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[17] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4068–4076.

[18] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[19] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[20] F.-X. Standaert, T. G. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *Advances in Cryptology - EUROCRYPT 2009*, A. Joux, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 443–461.

[21] NewAE Technology Inc., "Cw308 ufo target board," https://www.newae.com/products-1/NAE-CW308, 2020.

[22] kokke, "tiny-aes-c," https://github.com/kokke/tiny-AES-c,, 2017.

[23] E. Prouff, R. Strullu, R. Benadjila, E. Cagli, and C. Dumas, "Study of deep learning techniques for side-channel analysis and introduction to ascad database," Cryptology ePrint Archive, Report 2018/053, 2018, https://eprint.iacr.org/2018/053.