ISBN 978-981-14-4787-7
Proceedings of 2020 the 10th International Workshop on Computer Science and Engineering (WCSE 2020)
Shanghai, 19-21 June, 2020, pp. 30-36
doi: 10.18178/wcse.2020.06.006

An Enhanced Convolutional Neural Network in Side-Channel Attacks and Visualization

Minhui Jin[1], Mengce Zheng[1], Honggang Hu[1+] and Nenghai Yu[1]

[1] Key Laboratory of Electromagnetic Space Information, CAS
University of Science and Technology of China, Hefei, China.

Abstract. In recent years, the convolutional neural networks (CNNs) have received a lot of interest in the side-channel community. Based on the architecture of Residual Network, we build an enhanced CNN model called attention network. To enhance the power of feature representation of the attention network, we investigate an attention mechanism - Convolutional Block Attention Module (CBAM). By CBAM, attention network can attend to the informative points of the input traces and suppresses the irrelevant points. Finally, a new visualization method, named Class Gradient Visualization (CGV) is proposed to recognize which points of the input traces have a positive influence on the predicted result of the neural networks.

Keywords: cryptography, side-channel attack, convolutional neural network, attention mechanism, visualization

1. Introduction

Side-channel attack (SCA) is a class of cryptanalytic attacks but different from traditional cryptanalysis. It exploits the physical properties such as timing, power consumption, electromagnetic (EM) emanation and even sound. Recently, Convolutional neural networks (CNNs) have been introduced as a new alternative method to SCA. In the field of SCA, CNNs have become one of the most powerful attacks. In certain situations, they are even better than traditional SCA like template attack (TA)[1], while TA is considered to be the most powerful attack from an information-theoretic point of view. Compared to the traditional SCA, due to the translation invariance, CNNs are robust to the most common countermeasures like desynchronization or masking [1][2] [3]. In [2], Cagli et al. proposed a data augmentation method and noted that CNN can deal with the traces misalignment. Kim et al. addressed how to improve the performance of the neural network by adding the artificial noise to input traces in [1]. Benadjila et al. proposed a CNN model based on VGG and addressed the problems of selecting hyperparameters in [3]. In [4], Timon introduced a Sensitivity Analysis to reveal the secret key and PoIs of the input trace which is similar to the gradient visualization. Perin et al. evaluated the neural network using a backward propagation path method in [5].

In SCA, there exist the environment noises in the measurements. Traditional SCA and CNNs are all expected to focus on the informative points of the traces as far as possible. Because the irrelevant points will introduce extra noises and cause a worse performance of attacks. So in this paper, we propose an enhanced CNN model and introduce an attention mechanism - Convolutional Block Attention Module. CBAM can make the model to focus on the important points and suppressing unnecessary points. Finally, we propose a visualization technique called Class Gradient Visualization to verify the effectiveness of the new CNN model.

[+] Corresponding author. Tel.: +86 0551-63606377.
E-mail address: hghu2005@ustc.edu.cn.

2. Background

In this paper, we use the upper-case letter X to denote random variable. The lower-case letter $\vec{x}$ denotes the realization of X. The i-th observation of a random variable X is denoted by $\vec{x}_i$. For the encryption algorithm, there is $Z = f(P, K)$, where f is the cryptographic primitive and Z is the target sensitive variable. P denotes the public variable and K is the secret key. We denote k^* as the secret key of the cryptographic algorithm.

For the profiled attack, they set the plaintexts and the secret key on the replicated device to collect sufficient measurements M. Attackers generate a model $F: \mathbb{R}^D \rightarrow \mathbb{R}^{|Z|}$ from the set M. Then, attackers randomly choose plaintext p_i of the target device and collect some measurements N. The secret key k^* is unknown but fixed. Attackers compute the score vector $F(N)$ of the sensitive variable Z based on model F. Then they use the Maximum Likelihood strategy to compute the predicted probability of each key candidate.

In this paper, we use the Rank to evaluate the performance of models. Given N_a attacking measurements, the key guess vector is $\vec{g} = \{g_1, g_2, \dots, g_{|K|}\}$ in descending order of the predicted probability. The key guess vector is calculated by the log form of maximum likelihood strategy, i.e.,

$$g_i = \sum_{j=1}^{N} \log(\hat{p}_{ij}) \tag{1}$$

where $\hat{p}_{ij}$ denotes the predicted probability of i-th key candidates in the j-th attacking measurement. Rank denotes the average position of k^* in $\vec{g}$. When Rank is equivalent to 0, it implies that the attack is successful.

3. Architecture of the Enhanced Network

The enhanced network is based on the architecture of Residual Network. The basic structure is shown in Fig. 1. The attention network is composed of several layers which are convolutional layers (*Conv*), pooling layers (*Pooling*), and activation functions (*Relu*). The shortcut connection ($\oplus$) connects the input and output of the basic block. Besides, the flatten layer and fully-connected (*FC*) layers are adopted. Finally, an output layer is applied to generate the predicted probability of each class. The function f is used to make the input and output of basic block to have the same dimensions and it is a convolutional operation. The basic hyperparameters in the attention network are set as follows:

- The size of the filter is set to 11 and the convolutional stride is set to 1.
- The average-pooling is adopted. The pooling stride and pooling window are both set to 2.

To make the attention network to focus on the informative points of the input measurements, we add an enhanced module - Convolutional Block Attention Module (CBAM) proposed in [6]. CBAM separately underlines feature along two dimensions: channel and spatial. Channel attention computes the weight of each feature and focuses on the meaningful intermediate features. Spatial attention computes the weight of each time points and stresses where is the informative part. As shown in Fig. 2, the attention module is inserted in the first residual block and sequentially uses channel attention module and spatial attention module.

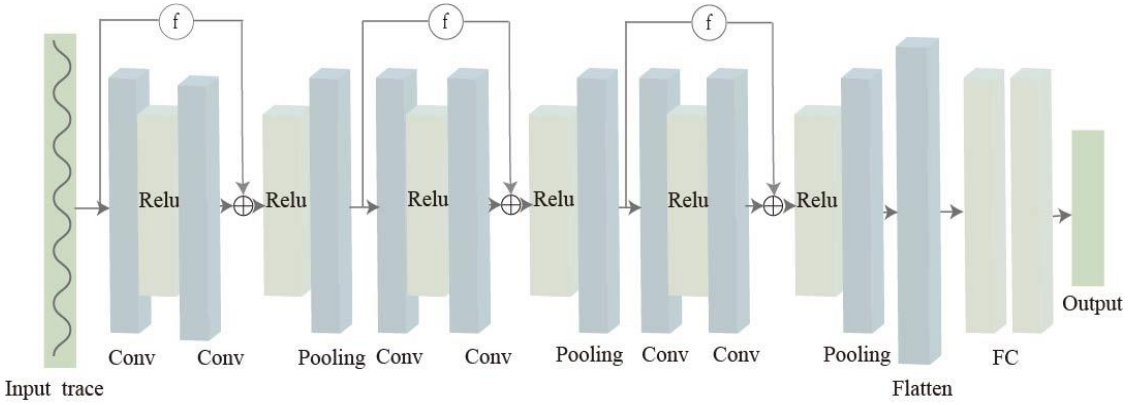


Fig. 1: The basic architecture of the attention network.

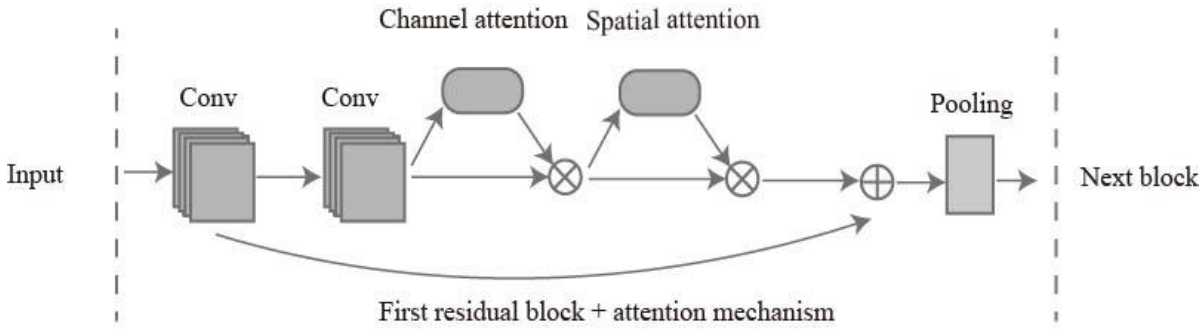


Fig. 2: Attention mechanism in attention network.

4. Performances of the Enhanced Network

In this section, we apply our attention network on different publicly available datasets. These datasets will be introduced in the Section 4.1 and the results will be shown in the Section 4.2.

4.1. Datasets

In our experiment, we use four public datasets. All of them are the implementations of the Advanced Encryption Standard (AES).

DPAcontest v4: The cryptographic algorithm of DPAcontest v4 is a masking software implementation of AES-256. We turn the masking protected implementation into the unprotected scenario. The corresponding leakage model is changed to:

$$Y(k^*) = Sbox[P_i \oplus k^*] \oplus M \tag{2}$$

where M is the mask of each cryptographic operation and the value is known. P corresponds to the plaintexts. We choose the first Sbox operation in the first round of cryptographic operation that i is set to 1.

AES_RD: The cryptographic algorithm in AES_RD is a protected software implementation of AES. The countermeasure of the algorithm applies random delay. The leakage model is defined as follows:

$$Y(k^*) = Sbox[P_i \oplus k^*] \tag{3}$$

We choose the first Sbox operation in the first round of cryptographic operation with $i = 1$.

AES_HD: The cryptographic algorithm in AES_HD is a typical class of the hardware implementations of AES-128. The leakage model is defined as follows:

$$Y(C_{i_1}, C_{i_2}, k^*) = Sbox^{-1}[C_{i_1} \oplus k^*] \oplus C_{i_2} \tag{4}$$

where C_{i_1} and C_{i_2} denote two ciphertexts and we choose $i_1 = 12$ and $i_2 = 8$.

ASCAD: The final dataset is ASCAD and the cryptographic algorithm is a masking software protected implementation of AES-128 [3]. The leakage model is defined as follows:

$$Y(k^*) = Sbox[P_i \oplus k^*] \tag{5}$$

We choose the third Sbox operation with $i = 3$.

4.2. Experiment Results

We show the performance of the attention network on four public datasets and compare it with the ASCAD network proposed in [3] and TA performed in [1]. As shown in Table 1, for the DPAcontest v4, attention network demands 3 traces which is the same to the ASCAD network. TA needs 4 traces to recover the secret key. For the AES_RD, ASCAD network needs 250 traces to make the average Rank less than 1, but the attention network only needs 170 traces. For TA, it cannot implement an efficient attack. For the AES_HD, the attention network only needs around 2100 traces to implement an efficient attack, while the ASCAD network and TA cannot recover the secret key. For the ASCAD, the attention network requires 550 traces, while the ASCAD network needs 770 traces. For traditional SCA, it did not retrieve the secret key.

Table 1: Required traces in different attack

	ASCAD network [3]	Template attacks [1]	Attention network
DPAcontest V4	3	4	3
AES_RD	250	>20 000	170
AES_HD	>25 000	>25 000	2100
ASCAD	770	>500	550

The results show that when the AES is software unprotected, CNN models and TA perform excellently. But for the protected implementations and hardware implementations, the attention network is much better than the ASCAD network and TA. It is because attention network focuses leakages and suppress the unnecessary points by using CBAM. It reduces the influence of the noises and improves performance.

5. Networks Visualization

We first introduce the principle of a new visualization method - Class Gradient Visualization. Then we use the visualization method to evaluate the effectiveness of the attention network and ASCAD network.

5.1. Class Gradient Visualization

In order to understand how CNNs work, we propose a new visualization method based on the Gradient-weighted Class Activation Mapping (Grad-CAM [7]), named Class Gradient Visualization (CGV). We compute the gradient of the features after the final pooling operation with respect to the class score. For the predicted class c, y^c denotes the class score before the softmax function. A represents the feature matrix after the final pooling operation such that $A \in \mathbb{R}^{D\times V}$. Here D denotes the number of samples and V denotes the number of features. x_i^j is an element of matrix A where $i \in \mathbb{R}^V$ and $j \in \mathbb{R}^D$. We first compute the gradient of feature matrix A with respect to the class score y^c and obtain the weights matrix W where $W \in \mathbb{R}^{D\times V}$. α_i^j denotes the weight of the j-th sample of the i-th feature:

$$\alpha_i^j = \frac{\partial\, y^c}{\partial\, x_i^j} \tag{6}$$

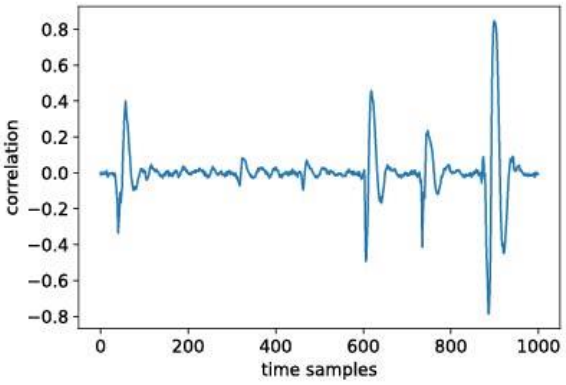
Then we compute a weight map W_{CGV}^c for class c and each element of it is computed as follows:

$$W_{CGV}^c = ReLU(\sum_i^V x_i \odot \alpha_i) \tag{7}$$

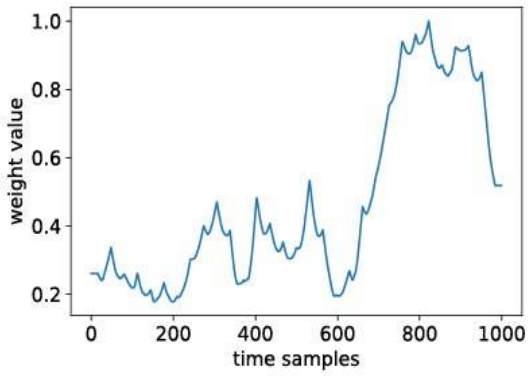
where $x_i \odot \alpha_i = (x_i^0\alpha_i^0, x_i^1\alpha_i^1, \dots, x_i^D\alpha_i^D)$. The larger value of the weight map shows that the corresponding time sample has a more important influence on the predicted class.

5.2. Visualization Results

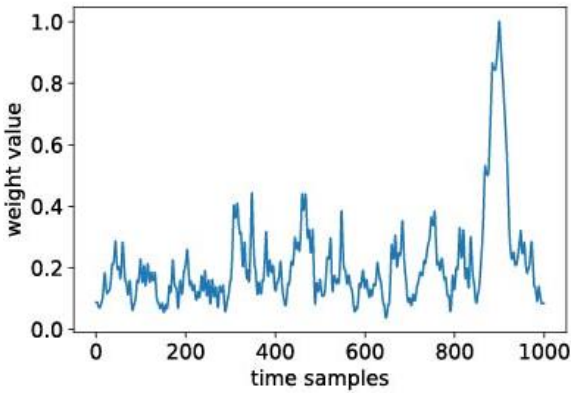
We visualize which components of the input traces have a positive influence on the final prediction by the CGV method in the trained networks. We exploit CPA analysis to find the real leakages of the traces. As is shown in Fig. 3. The visualization weight implies that attention network takes more attention to the leakages than ASCAD. For AES_RD, The region that the attention network focuses on is consistent in the region of leakages in CPA analysis, while the ASCAD network learns other areas (see Fig. 4). As for AES_HD, ASCAD network attends to the information of this area but still learns the other area. But the attention network mainly focuses on the area of leakages in the CPA analysis(see Fig. 5). For ASCAD, we find that the ASCAD network almost learns the whole trace. Attention network is similar to ASCAD network but it mainly focuses on three regions corresponding to the leakages of CPA analysis (see Fig. 6). By the CGV visualization method, it implies that the attention network takes more attention to the leakages compared to the ASCAD network.



(a)

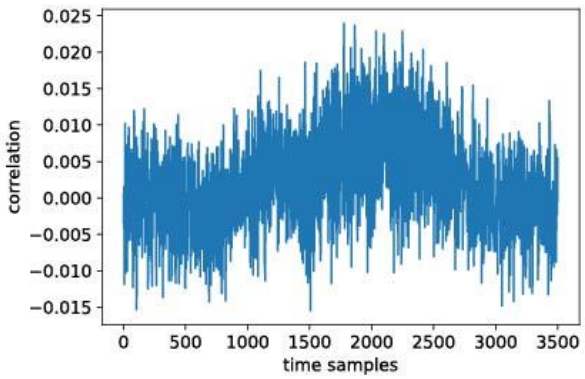


(b)

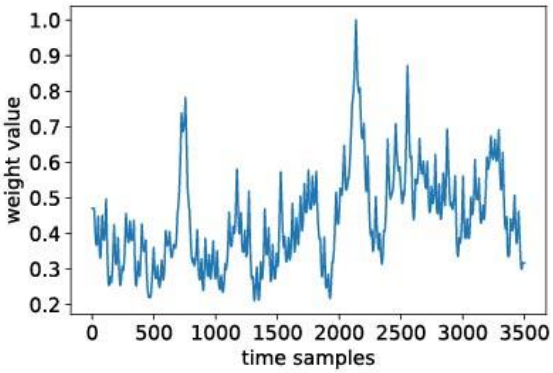


(c)

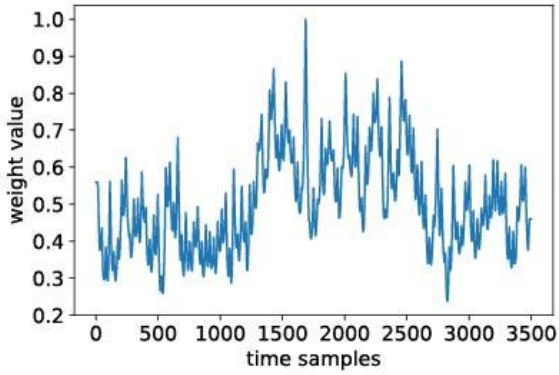
Fig. 3: (a) CPA analysis in DPAcontest v4; (b) Weight visualization of the ASCAD network in DPAcontest v4; (c) Weight visualization of the attention network in DPAcontest v4



(a)



(b)



(c)

Fig. 4: (a) CPA analysis in AES_RD; (b) Weight visualization of the ASCAD network in AES_RD; (c) Weight visualization of the attention network in AES_RD.

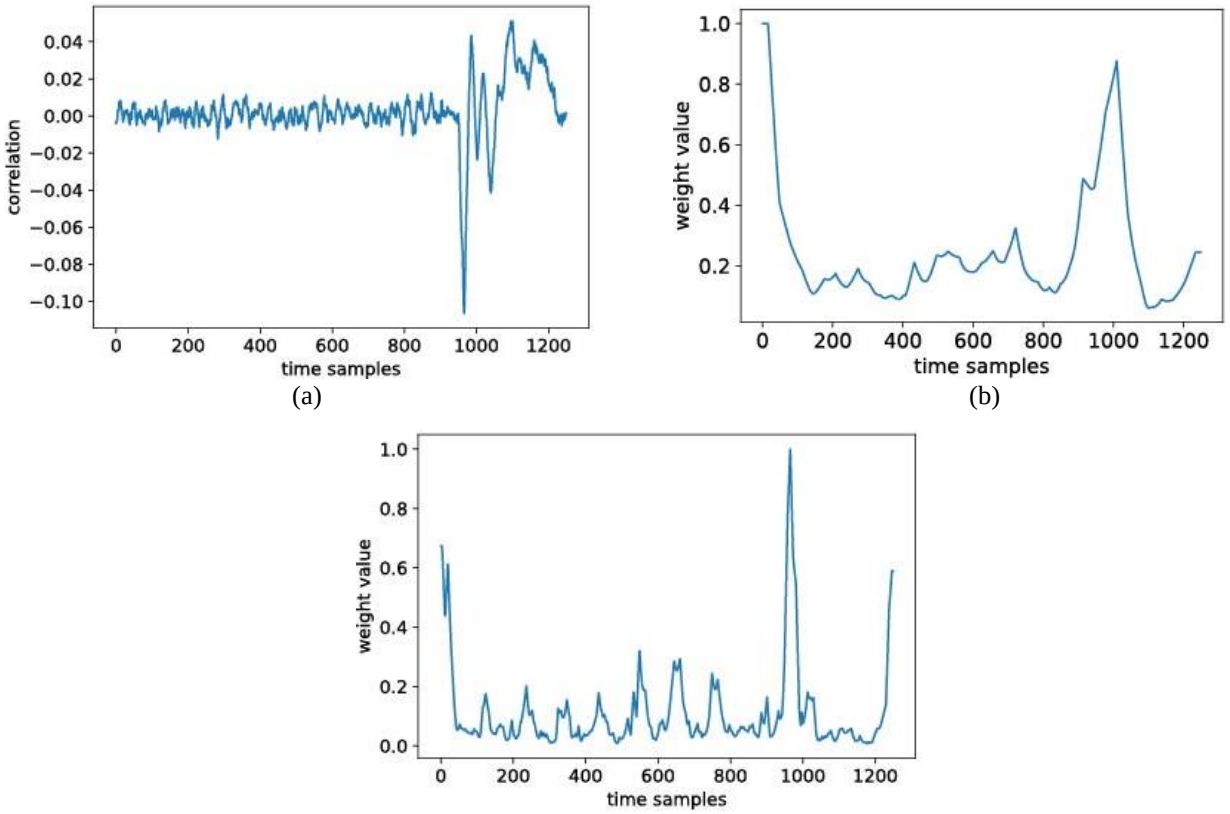


Fig. 5: (a) CPA analysis in AES_HD; (b) Weight visualization of the ASCAD network in AES_HD; (c) Weight visualization of the attention network in AES_HD.

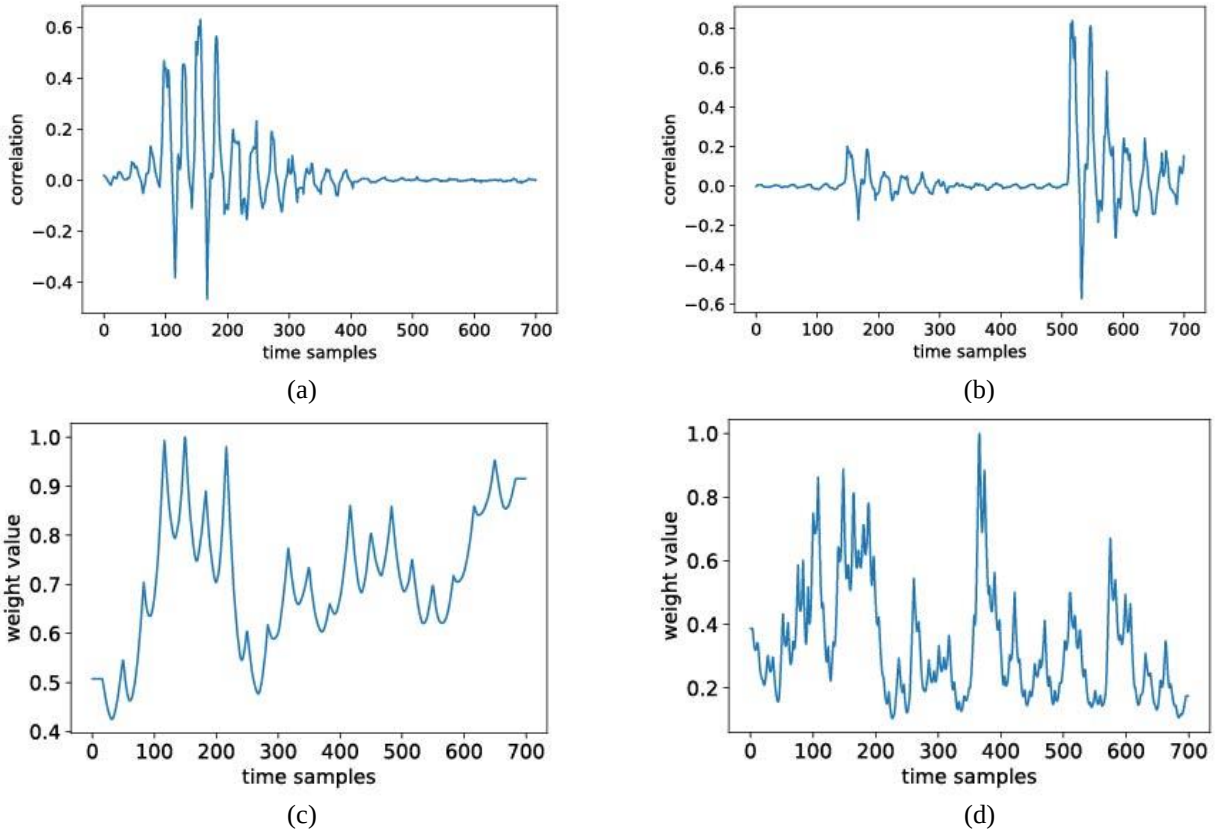


Fig. 6: (a) CPA analysis of masks in ASCAD; (b) CPA analysis of outputs of masked Sbox in ASCAD; (c) Weight visualization of the ASCAD network in ASCAD; (d) Weight visualization of the attention network in ASCAD

6. Conclusions

In this paper, we build an enhanced CNN model. We reduce the effect of the irrelative points by introducing an attention mechanism - Convolutional Block Attention Module (CBAM). Besides, we propose a new visualization method named Class Gradient Visualization. Through it, we can observe which points of the input traces have a positive influence on the final prediction. We use this visualization method to attention network and ASCAD network. The results show that the attention network takes more attention to the important leakage compared to the ASCAD network.

7. Acknowledgements

The authors would like to thank Information Science Laboratory Center of USTC for the hardware/software services. This work was supported by National Natural Science Foundation of China (Nos. 61972370 and 61632013), Fundamental Research Funds for Central Universities in China (No. WK3480000007) and Anhui Initiative in Quantum Information Technologies under Grant AHY150400.

8. References

[1] 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*. 2019(3): 148-179.

[2] E. Cagli, C. Dumas, and E. Prouff. Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures. In: W. Fischer, et al (eds.). *International Conference on Cryptographic Hardware and Embedded Systems*. Springer. 2017, pp. 45–68.

[3] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas. Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to ASCAD Database. *Journal of Cryptographic Engineering*. 2019.

[4] B. Timon. Non-Profiled Deep Learning-based Side-Channel attacks with Sensitivity Analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*. 2019(2): 107-131.

[5] G. Perin, B. Ege, and L. Chmielewski. Neural Network Model Assessment for Side-Channel Analysis. *IACR Cryptology ePrint Archive*. 2019:722.

[6] S. Woo, J. Park, JY. Lee, and I. S. Kweon. CBAM: Convolutional Block Attention Module. In: V. Ferrari, et al (eds.). *Proc. of the European Conference on Computer Vision*. Lecture Notes in Computer Science. 2018, pp. 3–19.

[7] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In: *Proc. of the IEEE International Conference on Computer Vision*. IEEE. 2017, pp. 618–626.