ChinaRxiv

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-201807.00049

Postprint: A Novel Template Attack Method for Cryptographic Chips Based on VGGNet Convolutional Neural Networks

Authors: Guo Dongxin, Chen Kaiyan, Zhang Yang, Xiaoyu Zhang, Li Jianlong

Date: 2018-07-09T00:00:00+00:00

Abstract

To address the intractable problems of traditional template analysis in practical attacks, this paper focuses on the VGGNet network model, which exhibits excellent feature extraction capabilities in the image recognition domain, and proposes a novel template attack method based on the VGGNet network model. To prevent signal quality from exerting a significant impact on model accuracy, correlation-based power analysis methods are employed to evaluate the quality of the collected side-channel signals; to accommodate the dimensional characteristics of side-channel signal data, the network model structure is appropriately adjusted; during the network training process, issues such as slow gradient descent, gradient vanishing, and overfitting are specifically addressed, and five-fold cross-validation is adopted to validate the trained model. Experimental results demonstrate that the test success rate of the VGGNet-based model reaches 92.3%, representing a 7.7% improvement over traditional template attack effectiveness.

Full Text

Preamble

New Template Attack Method for Encryption Chips Based on VGGNet Convolutional Neural Network

Guo Dongxin[1], Chen Kaiyan[1]†, Zhang Yang[1], Zhang Xiaoyu[2], Li Jianlong[1]
[1]Equipment Simulation Training Center, Shijiazhuang Campus, Army Engineering University, Shijiazhuang 050003, China [2]Unit 78090, PLA, Chengdu 610036, China

Abstract: To address the intractable problems of traditional template analysis in practical attacks, this paper investigates the VGGNet network model, which

demonstrates excellent feature extraction capabilities in image recognition, and proposes a novel template attack method based on the VGGNet architecture. To prevent signal quality from significantly impacting model accuracy, we employ correlation power analysis to evaluate the quality of collected side-channel signals. To accommodate the dimensional characteristics of side-channel signal data, we make appropriate adjustments to the network structure. During network training, we specifically address issues such as slow gradient descent, gradient vanishing, and overfitting, and validate the trained model using five-fold cross-validation. Experimental results demonstrate that the VGGNet-based approach achieves a success rate of 92.3%, representing a 7.7% improvement over traditional template attacks.

Keywords: VGGNet model; encryption chip; template attack; power analysis

0 Introduction

Traditional template attacks suffer from limitations in feature point selection and numerical instability during matrix inversion. Neural network-based template analysis overcomes these constraints by enabling unrestricted feature selection and avoiding explicit matrix inversion. In recent years, deep learning has advanced rapidly, yielding numerous effective feature extraction architectures. To improve template attack efficiency, this work adopts the VGGNet deep neural network model proposed in 2014. Compared to conventional Gaussian-based template attacks, VGGNet' s deep hierarchical structure provides superior feature extraction capabilities. After adaptive structural modifications, we apply it to side-channel template analysis to enhance attack effectiveness.

The main contributions of this paper are: (a) We analyze the implementation principles of convolutional neural networks, detail the VGGNet-16 architecture and parameter configurations, and perform adaptive structural adjustments; (b) To ensure signal quality does not adversely affect model construction, we validate side-channel signal quality; (c) During VGGNet-16 training, we address underfitting through polynomial expansion and transformation of training data to increase feature differentiation across classes, mitigate overfitting via dropout and L2 regularization, and accelerate gradient descent through data normalization; (d) Using data collected from the same device and batch, we conduct comparative experiments between Gaussian-based template analysis and VGGNet-based template analysis, demonstrating that the proposed method achieves higher matching success rates, improving by 7.7%.

1.1 Category Feature Conversion

Category feature conversion transforms non-numeric feature categories into numeric representations. In this work, we convert nine distinct Hamming weight

models into numeric categories to unify data labeling formats and facilitate machine learning. Since nine different data types exist, we assign labels 1 through 9 corresponding to Hamming weights 0–8. Each category is represented as a 9-dimensional one-hot vector, where the position corresponding to the category label is set to 1 and all other positions to 0, as shown in Table 1 .

1.2 Polynomial Features

Polynomial features generate additional feature attributes by applying predefined polynomial operations to feature data, enabling more effective classification. Assuming input features (a , a) with maximum polynomial degree 2, the resulting polynomial features become (1, a , a , a ^{2}, a a , a ^{2}). During network training, polynomial features increase inter-class differentiation, effectively addressing underfitting. As illustrated in Figure 1 [Figure 1: see original paper], data1 and data2 represent power consumption traces for adjacent Hamming weights. At point X=141, the maximum difference between original data is 0.97, while after polynomial transformation, this difference increases to 3.69, significantly enhancing class separability and facilitating efficient template construction.

1.3 Stochastic Gradient Descent

We first introduce definitions relevant to stochastic gradient descent algorithms [4].

Definition 1: If function f at point a(x, y, z) has the following limit along direction l (with direction angles , ,):

$$\frac{\partial f}{\partial l} = \lim_{\Delta s \to 0} \frac{f(x+\Delta x, y+\Delta y, z+\Delta z) - f(x,y,z)}{\Delta s}$$

then this limit is called the directional derivative of f at point p along direction l.

The projection of the gradient onto direction l yields the directional derivative. The directional derivative formula is:

$$\frac{\partial f}{\partial l} = \frac{\partial f}{\partial x}\cos\alpha + \frac{\partial f}{\partial y}\cos\beta + \frac{\partial f}{\partial z}\cos\gamma$$

When the direction aligns with the gradient, the cosine value equals 1 and the directional derivative reaches its maximum:

$$\frac{\partial f}{\partial l} = \mathbf{G}\cdot\mathbf{l} = |\mathbf{G}|\cos(\mathbf{G},\mathbf{l})$$

where $\mathbf{G} = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)$. The gradient direction represents the direction of maximum change rate, and its magnitude represents the maximum rate of change.

Definition 2: For function z = f(x, y) with continuous first-order partial derivatives in region D, for each point P(x, y) D, the vector:

$$\text{grad} f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$$

is called the gradient of z = f(x, y) at point P.

Using the Nebla operator $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$, the gradient can be simplified as $\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$.

Gradient descent, also known as steepest descent, is an optimization algorithm that iteratively finds optimal values during model fitting. The descent direction follows the negative gradient, with step size decreasing as the solution approaches the target. The iteration formula is:

$$x_{k+1} = x_k + \alpha s_k$$

where x is the value to be optimized, s is the negative gradient direction, and α is the step size.

Stochastic gradient descent extends gradient descent. In deep learning, the cost function decomposes into the sum of per-sample costs. As training sets grow, computing full gradients becomes computationally expensive. SGD' s core insight is that the gradient is an expectation that can be approximated using small samples. Specifically, at each iteration, we uniformly sample a mini-batch from the training set. The gradient estimate becomes:

$$g = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L(x^{(i)}, y^{(i)}; \theta)$$

where m is the mini-batch size and L is the per-sample loss. The parameter update follows:

$$\theta \leftarrow \theta - \epsilon g$$

where ϵ is the learning rate.

1.4 Underfitting and Overfitting

If a model cannot achieve low error on a specific training set, it underfits the data. If a large gap exists between training and test error, the model overfits. To quantitatively assess these conditions, we introduce model capacity—the ability of a model to fit various functions. Generally, lower-capacity models struggle to fit training sets, while higher-capacity models fit them easily. During training, we address underfitting by adding features and polynomial preprocessing, and mitigate overfitting through regularization.

1.5 Regularization and Dropout

Regularization addresses overfitting by reducing model complexity, typically by adding a regularization term to the loss function. The primary forms are L1 and L2 regularization:

$$L_1 = \alpha \sum |w|, \quad L_2 = \alpha \sum w^2$$

where α is the regularization coefficient and w represents model weights.

Dropout is another regularization technique that randomly removes neurons from hidden layers during training, using only a subset of neurons to update weights and biases. Since averaging predictions across different networks trained on the same data typically reduces overfitting, dropout exploits this principle by training different model architectures in each iteration. This weakens interdependencies between neurons and enhances generalization, thereby improving classification accuracy. Figure 2 [Figure 2: see original paper] illustrates this principle, showing a complete network on the left and the dropout-modified network on the right.

2 Model Analysis and Adjustment

Karen Simonyan and Andrew Zisserman' s VGGNet model [2] achieved excellent performance in the 2014 ILSVRC competition, demonstrating superior feature extraction capabilities. Evolved from AlexNet, VGGNet employs smaller convolution kernels and deeper networks. The competition results confirmed VGGNet' s exceptional feature acquisition ability, as shown in Figure 3 [Figure 3: see original paper].

The authors designed six variants (A, A-LRN, B, C, D, E), with model D performing best. This work adopts VGGNet-D, a 16-layer network comprising 13 convolutional layers and 3 fully connected layers. To enhance feature extraction, all convolution kernels use 3×3 receptive fields. The architecture includes: 64 feature maps in convolutions 1-2, 128 in convolutions 3-4, 256 in convolutions 5-7, and 512 in convolutions 8-13. Five subsampling layers with scaling factor 2 are positioned between convolutions 2-3, 4-5, 7-8, 10-11, and 13-FC1. The first two fully connected layers contain 4096 neurons each, while the third layer' s neuron count depends on classification categories (1000 for ImageNet).

2.2 VGGNet Model Structure Adjustment

To adapt VGGNet for side-channel analysis, we make the following modifications: First, since side-channel signals are one-dimensional unlike

two-dimensional images and lack channels, we adjust all convolution kernels from 3×3 to 2×1 to increase feature sensitivity. Correspondingly, subsampling layers transition from 2D to 1D processing regions. Second, because side-channel signal features exhibit subtle differences, we accelerate model construction and improve classification accuracy by categorizing data into nine Hamming weight classes, requiring the output layer to contain nine neurons. The adjusted architecture is shown in Figure 4 [Figure 4: see original paper].

---

3 Experiments

We implement AES-128 on an AT89C52 microcontroller, triggering on the AddRoundKey operation in the first round. Random plaintext is transmitted via RS-232 during encryption, and side-channel power signals are captured using a Tektronix DPO4032 oscilloscope.

3.1 Side-channel Signal Quality Assessment

To prevent model underfitting due to poor signal quality, data validation and cleaning are essential. We target the first 8 key bits of AES, using correlation power analysis [6] to attack 15, 50, 100, and 200 power traces, as shown in Figures 5 [Figure 5: see original paper] through 8 [Figure 8: see original paper]. The horizontal axis represents 256 possible key guesses (showing only 128 due to Hamming weight symmetry [7]), while the vertical axis shows correlation coefficients. As trace count increases, correlation coefficients for incorrect keys decrease significantly. The correct key (0x2B, marked as X:44) exhibits the highest correlation coefficient with good distinguishability, confirming adequate signal quality.

3.2 Hamming Weight Model Verification

To verify Hamming weight model correlation on the AT89C52, we partition traces at the point of maximum correlation from CPA. The attack targets the first-round AddRoundKey operation, partitioning intermediate values from the XOR of plaintext and key into nine Hamming weight categories (0–8), as shown in Figure 9 [Figure 9: see original paper]. Each trace contains 100,000 points, necessitating localization of maximum correlation points before template construction. We use CPA for this localization, as shown in Figure 10 [Figure 10: see original paper]. Multiple positions exhibit high correlation; we record point indices in descending order of correlation coefficient magnitude to ensure optimal feature extraction.

3.3 Comparative Experiment

We compare traditional Gaussian-based template attacks [9] with our VGGNet-based approach using data from the same device and batch. The training set

comprises 9,000 traces (1,000 per Hamming weight), and the test set contains 1,800 traces (200 per class). Test data remains unseen during training. We employ five-fold cross-validation during neural network training, partitioning training data into five subsets, each serving once as validation set while the others train the model. After training, we evaluate the final model on the test set.

We select 100 feature points using the method from Section 3.2 and apply polynomial feature squaring from Section 2.2 to enhance inter-class differences, effectively addressing underfitting. Results are presented in Figure 11 [Figure 11: see original paper] and Table 2 .

Figure 11 shows that VGGNet-based template attacks achieve higher matching success rates than traditional Gaussian-based attacks across most Hamming weight categories. Lower matching rates for weights 1–2 and 6–7 indicate reduced distinguishability in these classes.

Table 2 reveals that traditional template attacks require less construction time as they involve no parameter tuning, whereas VGGNet-based attacks demand more time for iterative parameter updates and potential restructuring if overfitting occurs. Attack times are comparable. However, the average matching success rate improves from 84.6% for traditional methods to 92.3% for VGGNet, representing a 7.7% enhancement.

---

4 Conclusion

This paper proposes a novel template attack method based on the VGGNet architecture, applying a modified deep neural network to side-channel analysis. Leveraging deep learning' s strong feature extraction capabilities, we improve template attack success rates. By collecting power side-channel signals from an AT89S52 microcontroller running AES-128 and conducting comparative experiments, we demonstrate that VGGNet-based template attacks outperform traditional approaches, achieving a 7.7% improvement in matching accuracy.

References

[1] Kocher P C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems [C]// Proc of CRYPTO. 1996: 104-113.

[2] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. Computer Science, 2014.

[3] Kocher P C, Jaffe J, Jun B, et al. Introduction to differential power analysis [J]. Journal of Cryptographic Engineering, 2011, 1(1): 5-27.

[4] Goodfellow I, Bengio Y, Courville A. Deep Learning [M]. Translated by Zhao Shenjian, Li Yijun, Fu Tianfan, et al. Beijing: People' s Posts and Telecommunications Press, 2017.

[5] Ou Changhai, Wang Zhu, Huang Weiqing, et al. Hamming weight model based cryptographic device amplification template attack [J]. Journal of Cryptology, 2015, 2(5): 477-486.

[6] Zhang Xiaoyu, Chen Kaiyan, Zhang Yang, et al. Dynamic telescopic alignment of side-channel power consumption signal based on DTW algorithm [J]. Application Research of Computers, 2017, 34(9): 2782-2785.

[7] Zhang Yang, Chen Kaiyan, Li Xiongwei, et al. Research on password chip side-channel attack based on difference degree [J]. Journal of Communications, 2015, 36(3): 2015066.

[8] Mangard S, Oswald E, Popp T. Power Analysis Attacks [M]. Translated by Feng Dengguo, et al. Beijing: Science Press, 2010.

[9] Deng Gaoming, Zhao Qiang, Zhang Peng, et al. Electromagnetic domain template analysis attacks on cipher chips [J]. Chinese Journal of Computers, 2009, 32(4): 602-610.

[10] Yu Kai, Jia Lei, Chen Yuqiang, et al. Deep learning: yesterday, today, and tomorrow [J]. Journal of Computer Research and Development, 2013, 50(9): 1799-1804.

[11] Cai Z, Fan Q, Feris R S, et al. A unified multi-scale deep convolutional neural network for fast object detection [C]// Proc of European Conference on Computer Vision. Cham: Springer, 2016: 354-370.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.