applied sciences

MDPI

Article

A Secret Key Classification Framework of Symmetric Encryption Algorithm Based on Deep Transfer Learning

Xiaotong Cui [1], Hongxin Zhang [1,2,*], Xing Fang [1], Yuanzhen Wang [3], Danzhi Wang [1], Fan Fan [4] and Lei Shu [4]

[1] School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China; cuixiaotong@bupt.edu.cn (X.C.); fancy_t@bupt.edu.cn (X.F.); danzhiwang@bupt.edu.cn (D.W.)
[2] Beijing Key Laboratory of Work Safety Intelligent Monitoring, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China; wangyz7@bupt.edu.cn
[4] Beijing Microelectronics Technology Institute, Beijing 100076, China; 18911650922@163.com (F.F.); 18001350960@163.com (L.S.)
* Correspondence: hongxinzhang@bupt.edu.cn

Abstract: The leakage signals, including electromagnetic, energy, time, and temperature, generated during the operation of password devices contain highly correlated key information, which leads to security vulnerabilities. In traditional encryption algorithms, the length of the key greatly affects the upper limit of its security against cracking. Regarding side-channel attacks on long-key algorithms, traditional template attack methods characterize the energy traces using multivariate Gaussian distribution during the template construction phase. The exhaustive key-guessing process is expected to consume a significant amount of time and computational resources. Therefore, to analyze the effectiveness of obtaining key values from the side information of password devices, we propose an innovative attack method based on a divide-and-conquer logical structure, targeting semi-bytes. We construct a collection of key classification submodules with symmetric correlations. By integrating a differential network model for byte-block sets and an end-to-end direct attack method, we form a holistic symmetric decision framework and propose a key classification structure based on deep transfer learning. This structure consists of three main parts: side information data acquisition, analysis of key-value effectiveness, and determination of attack positions. It employs multiple parallel symmetric subnetworks, effectively improving attack efficiency and reducing the key enumeration range. Experimental results show that the optimal attack accuracy of the network model can reach 91%, with an average attack accuracy of 78%. It overcomes overfitting issues under small sample dataset conditions.

Keywords: side-channel attack; AES; symmetrical decision; deep learning; transfer learning

check for updates

Citation: Cui, X.; Zhang, H.; Fang, X.; Wang, Y.; Wang, D.; Fan, F.; Shu, L. A Secret Key Classification Framework of Symmetric Encryption Algorithm Based on Deep Transfer Learning. *Appl. Sci.* **2023**, *13*, 12025. https://doi.org/10.3390/app132112025

Academic Editor: Christos Bouras

Received: 29 September 2023
Revised: 21 October 2023
Accepted: 1 November 2023
Published: 3 November 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the rapid development of science and technology in the information age, the widespread use of various electronic devices has brought great convenience to our lives. At the same time, the security of these devices has become an important research focus. During the operation of electronic devices, inevitable unintended leaks occur, such as electromagnetic radiation, power consumption, timing, and sound, among others. Although these pieces of information may not be directly related to the mathematical foundations of cryptographic algorithms, they can provide crucial clues about the internal operation of cryptographic systems. This poses a threat to the security of devices [1]. Side-channel analysis methods play a significant role in the field of cryptography and information security. They help identify and address potential issues in cryptographic systems, enhance the security of cryptographic systems, and improve cryptographic algorithms and

implementations. This contributes to ensuring the confidentiality, integrity, and availability of data, as well as protecting personal privacy and sensitive information [2].

Since the introduction of timing attacks [3], side-channel analysis has posed new challenges to the security of cryptographic devices compared to traditional cryptographic analysis. Side-channel information is collected without attempting to break the encryption or authentication mechanisms themselves. Instead, it involves extracting information by analyzing the behavior or characteristics of the system. This makes such attacks difficult to detect and exhibits a dual nature of high flexibility and high threat. Traditional side-channel analysis techniques include correlation power analysis [4], differential power analysis [5], template attacks [6–8], and others. These methods exploit the statistical characteristics of leaked signal amplitudes in the device and rely on the data being processed in the device to make guesses about the actual key. Among them, template attacks characterize the energy traces using multivariate normal distributions, and the template is constructed based on mean vectors and covariance matrices for key pairs. Such key-scanning enumeration structures face significant probability retrieval demands in long key systems, and the computational power of practical devices cannot support the required attack actions within a reasonable time-frame.

Compared to traditional side-channel attack methods, the application of machine learning [9] and deep learning [10,11] has brought new perspectives in side-channel analysis [12]. However, existing deep learning methods still have limitations in side-channel analysis. Deep learning has been well-established for tasks such as image processing [13,14] and speech recognition [15,16], but its application in cryptographic algorithms is relatively limited. Classical handwritten digit classification involves 10 classes, while the analysis of cryptographic keys requires exponentially more classification categories, making existing classification models less suitable. Moreover, electromagnetic signals exhibit higher similarity compared to other one-dimensional signals like speech, and the leaked information contains more environmental noise and disturbances. These characteristics demand neural networks with more refined feature extraction capabilities. Additionally, training the attack network model requires a large dataset and consumes a considerable amount of time during the training phase. Therefore, in this paper, we propose a key classification framework based on deep transfer learning. The main contributions of this paper are as follows:

1. We proposed an effective key-oriented experimental design for side-channel analysis, based on the "divide and conquer" approach, and designed a key classification method based on semi-byte length.

2. We utilized an adaptive residual convolutional neural network to extract characteristics from side-channel electromagnetic signals, achieving accurate mapping between electromagnetic signals and unique keys. Developed a residual convolutional neural network model based on one-dimensional electromagnetic signals.

3. We introduced a parallel analysis structure with multiple symmetric sub-networks, combined with transfer learning methods to optimize network parameters. Improved processing efficiency at different attack positions with different keys, thus reducing the key enumeration range. Reduced the amount of training data samples, making it more applicable for small sample attack scenarios.

The remaining sections of this paper are organized as follows. In Section 2, we introduce the related work in side channel attack. In Section 3, we introduce the key classification framework based on deep transfer learning. Section 4 describes the highly effective analysis method based on residual neural networks. Section 5 presents the parallel analysis structure with multiple symmetric sub-networks. Finally, in Section 6, we showcase the corresponding experimental results.

2. Related Work

Side-channel analysis is a cryptographic attack technique that involves observing information leaks generated by cryptographic systems during their actual operation to obtain knowledge about cryptographic keys or other sensitive information. Due to the

diversity of encryption devices and algorithms, there have been numerous works in the field of side-channel analysis. T.N.Quy et al. [17] proposed a practical procedure for electromagnetic analysis attack methods and presented experimental results on smart cards. Zhang et al. [18] utilized electromagnetic signal leakage between different individuals to distinguish between different instances of the same encryption device. Experimental results demonstrated the effectiveness of deep residual network structures in electromagnetic signal recognition. S. Hong et al. [19] conducted a security analysis of cache side-channels. Ref. [20] showcased cross-device deep learning side-channel attacks (X-DeepSCA), which employ traces captured from multiple devices for enhanced training and can compromise different target devices. Authors Li and Ou [21] abstracted side-channel analysis into a regression problem based on circuit state variations and the principles of specific operations in cryptography, using regression models to improve attack accuracy. In addition to research focused on near-field electromagnetic analysis, R. Wang et al. [22] utilized far-field electromagnetic radiation signals to launch deep learning-based side-channel attacks on various Bluetooth encryption devices, achieving key recovery in an environment located 15 m away from the target.

Machine learning also has a strong foundation in the application of side-channel analysis [23,24]. In [25], the authors used a supervised classification neural network to transform the key enumeration process into a classification process. This greatly reduces the number of attempts on ineffective key datasets in traditional side-channel analysis methods, and the quantitative prior information obtained through the classification process effectively improves the attack efficiency. L. Weissbart et al. [26] explored several machine learning techniques to mount a power analysis attack on EdDSA using Curve25519 as implemented in WolfSSL. In reference [27], several commonly used machine learning methods are introduced in the identification and exploitation stages of side-channel analysis. By adjusting the appropriate parameters and conducting training, machine learning methods outperform ordinary template attacks. Sun et al. [28] employed machine learning and neural network models to successfully detect trojans using electromagnetic information leaked from devices.

The application of deep learning techniques in side-channel analysis has shown that deep networks provide a more sophisticated analysis of signals and achieve better attack results compared to template attacks and traditional machine learning [29–32]. It is crucial to train and test on traces captured from different boards and using diverse implementations of the cryptographic algorithm under attack [33]. In reference [34], a systematic understanding of the contributions of deep learning in side-channel analysis is provided, analyzing the different stages where deep learning can be employed. Lu et al. [35] proposed a neural network architecture composed of an encoder, attention mechanism, and classifier for end-to-end analysis. The constructed network can directly classify traces containing a large number of time samples, avoiding complex and tedious feature extraction steps. Zhou and Standaert [36] highlighted one advantage of deep learning in side-channel analysis, which is its tolerance for misaligned signals. The feature extraction mechanism in neural networks simplifies the data preprocessing stage, reducing workload while improving efficiency. In reference [37], a side-channel attack method based on LSTM deep learning networks is proposed. This method utilizes correlation power analysis (CPA) to extract relevant information from side-channel power consumption data. Then, based on the position of points of interest, suitable interest intervals are selected as feature vectors for creating the neural network model. L.Zhang et al. [38] proposed a novel SCA method utilizing multilabel classification from the bit-to-byte view.

Despite the extensive work in side-channel analysis, most research focuses on individual devices or specific intermediate values within encryption algorithms. Due to the high complexity designed to ensure information security, encryption algorithms rarely have side-channel analysis methods to obtain the main encryption keys. Therefore, we have designed a key classification framework based on deep transfer learning, which provides a new perspective for side-channel analysis.

3. Overview of the Framework

The purpose of this paper is to propose a key classification framework for encryption algorithms, based on deep neural network models and incorporating transfer learning methods to improve model training speed and attack efficiency. The overall content of this paper is illustrated in Figure 1. The overall design is divided into four parts: side-channel information leakage acquisition of the encryption chip, construction of the side-channel dataset, analysis of side-channel information leakage, and attack results on encryption algorithm keys. Part A showcases our acquisition platform for side-channel information leakage from the experimental cryptographic chip. The acquisition system records the relevant key information during the operation of the encryption chip and receives and stores the electromagnetic information leaked by the chip captured by an oscilloscope. Part B introduces the dataset of leakage signals generated during the operation of the encryption chip. In Parts C and D, using deep transfer learning methods, we construct side-channel attack models suitable for key classification in symmetric encryption algorithms. We elucidate the key classification method based on deep learning and the high effectiveness of transfer learning in concurrent computation mechanisms for model classification.

Our ultimate goal is to achieve the accurate classification of encryption keys. However, due to the security requirements of cryptographic algorithms, the long length of keys leads to an exponential explosion in the classification model and an insufficient quantity of data distribution. To address this issue, we have split the keys of the cryptographic algorithm. The results show that key splitting improves classification efficiency and enhances the feasibility of extracting useful information from side-channel data. Based on the characteristics of symmetric encryption algorithm keys, we adopt a "divide and conquer" approach to split the keys, analyzing them at the semi-byte level as the smallest unit. We construct a supervised dataset and feed it to a classification model based on deep neural networks. In this paper, we utilize traditional deep neural network architectures and adjust the parameters according to the characteristics of electromagnetic signals. We establish a mapping between the keys and side-channel information, and propose a new classification approach for the overall encryption algorithm by combining transfer learning. The experimental results demonstrate that this method effectively extracts information related to the encryption algorithm keys from the leakage of side-channel information, achieving high accuracy.

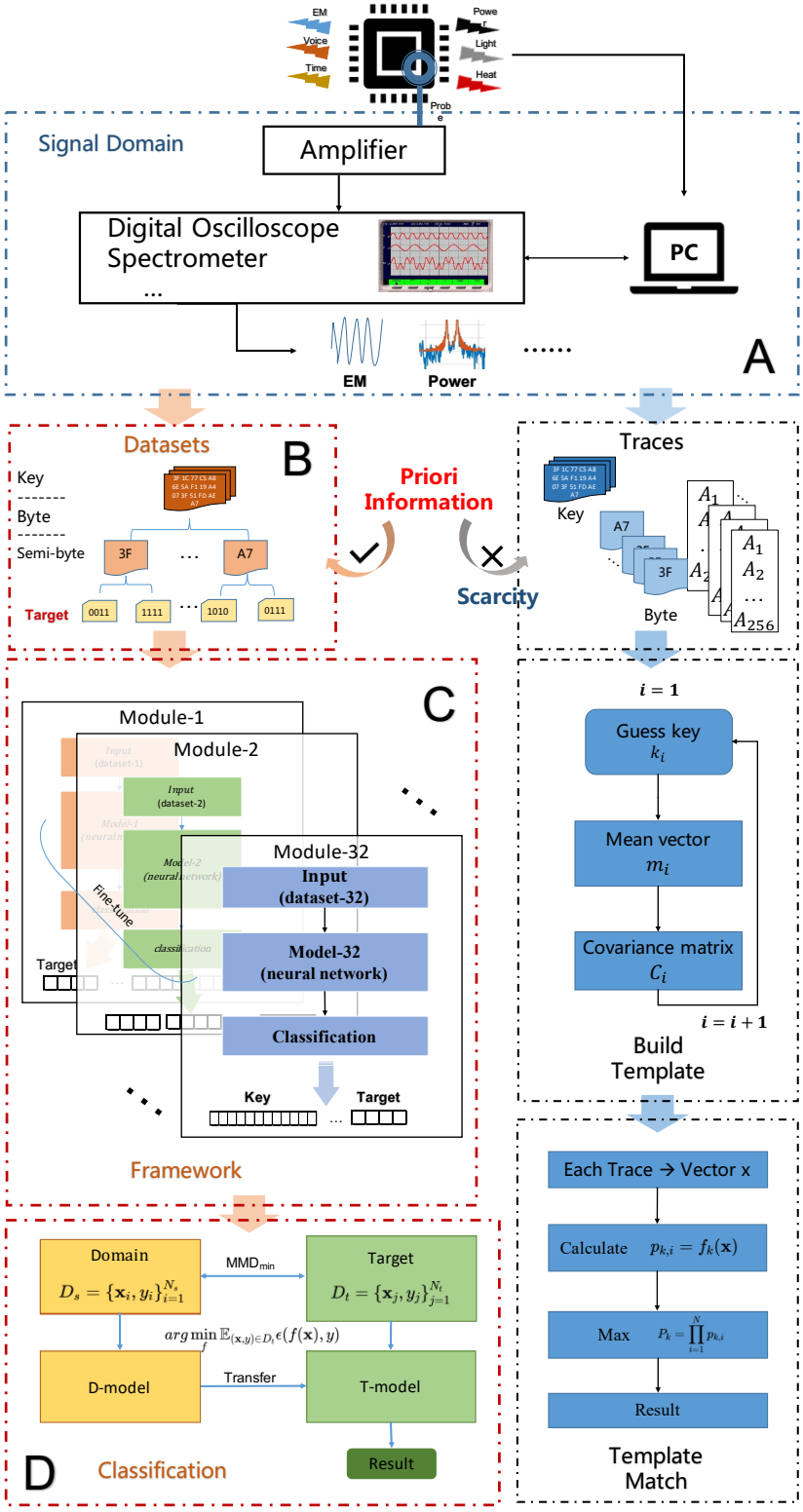


Figure 1. The Overall framework for key classification in symmetric encryption algorithm. Part (**A**) showcases our acquisition platform for side-channel information leakage from the experimental cryptographic chip. Part (**B**) introduces the datasets of leakage signals generated during the operation of the encryption chip. Parts (**C**,**D**) introduce how to obtain the side-channel attack models suitable for key classification in symmetric encryption algorithms.

4. Highly Effective Analysis Method Based on Residual Neural Networks

In this chapter, we will provide a detailed introduction to the model structures used in this paper. In our experiments, we employed 32 symmetric deep neural network models for key classification and utilized transfer learning methods for parameter sharing among different models. We also performed model parameter preprocessing. We improved upon the traditional template attack method, which serves as the foundation. The main idea behind the traditional template attack is to target a specific byte and construct a key matrix between the guessed key and the leakage points. Then, the target information is matched and guessed against the matrix. With the emergence of deep learning, template construction has transformed into key classification models. By training the models, we obtain templates related to key cracking. During the attack phase, the target information is input into the model to obtain the attack results.

4.1. Basic Structure

The convolutional layer operates by repeatedly applying a convolutional kernel to each receptive field of the input data. The result of the convolution is equivalent to the input data being filtered by different filters, and the output is a feature map obtained after data filtering. Then, the features from each local receptive field are fed as inputs to the connected neurons in the next layer. Assuming the input (net activation) of the j channel in the l convolutional layer is denoted as z_j^l, and the output is represented as o_j^l, then

$$\begin{cases} o_j^l = f(z_j^l) \\ z_j^l = \sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l \end{cases} \tag{1}$$

in which $f(\cdot)$represents the activation function, M_j represents the subset of feature maps from the $(l-1)$layer, k_{ij}^l is the convolutional kernel matrix, b_j^l is the bias term, and "$*$" denotes the convolution operation.

The pooling layer partitions the input feature map into non-overlapping regions and performs downsampling on each region (e.g., max pooling, average pooling, sum pooling, random pooling, etc.). This introduces certain translational and scaling invariance to the features, reducing computational complexity while enhancing the robustness of the model. If the input feature map has n channels, the number of feature maps remains the same after the pooling layer, but the output feature maps become smaller. Let α represent the sampling weights and denote the pooling function as $pool(\cdot)$. The formula for the output feature map of the pooling layer is

$$\begin{cases} o_j^{l+1} = f(z_j^{l+1}) \\ z_j^{l+1} = \alpha_j^{l+1} pool(x_j^l) + b_j^{l+1} \end{cases} \tag{2}$$

After multiple alternating convolutional and pooling layers, the final layer is the fully connected layer. This involves flattening all the two-dimensional feature maps extracted from the multiple convolutional layers into a one-dimensional feature sequence, which is then fed into the fully connected layer as inputs to the neurons. Through the weighted sum and activation function applied to the output of the previous layer, we can obtain the output o_j^m of the j neuron in m fully connected layer in which

$$\begin{cases} o_j^m = f(z_j^m) \\ z_j^m = w_j^m x_j^{m-1} + b_j^m \end{cases} \tag{3}$$

4.2. Network

In the design of network modules, taking into account the one-dimensional nature of side-channel signals, we selected the residual network structure as our training model. We utilized the optimal structure proposed by the authors in reference [39], which was introduced as a deep convolutional neural network architecture known for effectively training ultra-deep neural networks in reference [40]. In this paper, the residual block, as shown in Figure 2, consists of batch normalization, convolutional layers, pooling layers, and activation functions. In this network structure, BN (Batch Normalization) and ReLU are applied before the weights. This pre-activation approach makes it easier to optimize the model function and helps reduce overfitting.

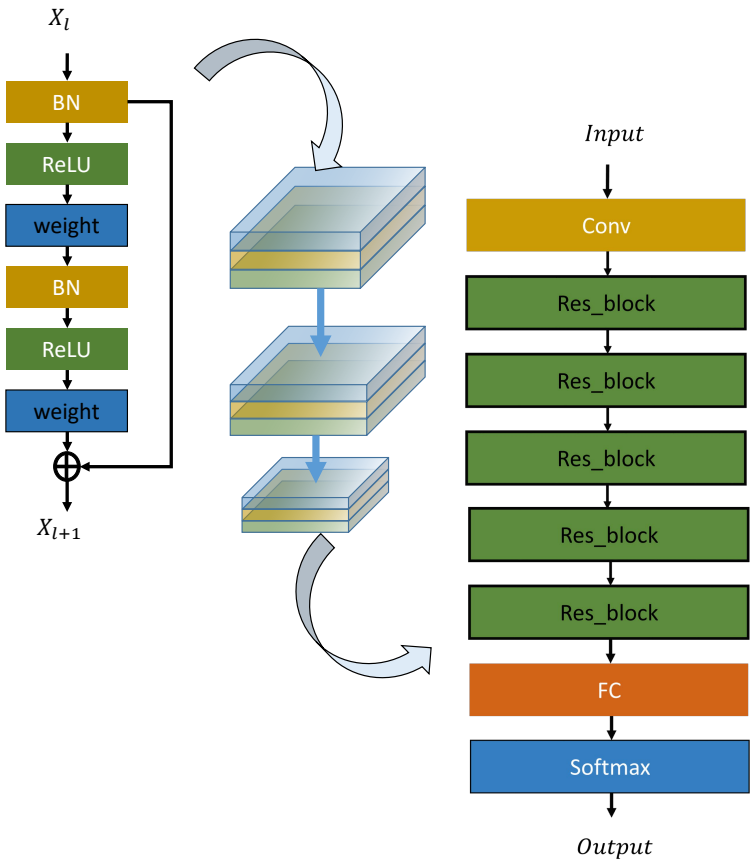


Figure 2. The detail network structure with residual blocks.

The key innovation of the ResNet structure lies in the introduction of residual structures or skip connections. These structures address the issue of gradient vanishing during the data flow process as the network becomes deeper, avoiding suboptimal training results. There are several advantages of using deep residual network models to analyze and process our electromagnetic signals:

1. The electromagnetic signals have a larger number of effective points compared to individual image pixels. While each image typically has a few hundred numerical values, electromagnetic signals usually consist of tens of thousands of effective values. Increasing the depth of the network would require dealing with a large number of parameters. However, the residual network keeps each hidden layer relatively thin, allowing for control over the parameter count and saving computational space.

2. Electromagnetic signals differ from pixel-level differences in images. The differences in electromagnetic signals are reflected in small amplitude statistical characteristics, requiring deep neural networks to extract deeper-level features. The hierarchical structure of the residual network ensures a progressive increase in the number of features at each layer, thereby maintaining the expressive power of the output features.

3. The residual structure of the identity mapping in the ResNet architecture inherently avoids the problem of gradient vanishing, which can hamper network training.

Accuracy serves as the evaluation metric in our experiments, allowing for a comprehensive quantification of the results. The specific formula for calculating accuracy is as follows:

$$A_{CC} = \frac{\sum TP}{TP + TN + FP + FN} \tag{4}$$

where A_{CC} represents the overall accuracy of the evaluation. TP, TN, FP, and TN represent the number of true positives, false negatives, false positives, and true negatives.

4.3. A Password Classification Method Targeting Semi-Byte Length

The divide-and-conquer strategy is an algorithmic design paradigm that improves the efficiency of key attacks. This algorithm recursively decomposes a problem into two or more identical or related subproblems until these subproblems become simple enough to be solved directly. Then, the solutions to the subproblems are combined to obtain the solution to the original problem. As shown in Figure 3, in the analysis of cryptographic algorithms in this paper, the divide-and-conquer strategy is used to decompose long keys into bytes, avoiding exponential explosion in template construction and alleviating the difficulty of training the model. In the classification of electromagnetic signals leaked from devices, the divide-and-conquer strategy transforms the classification of the master key into the classification of multiple similar subnetworks, reducing the complexity of the classification model structure, decreasing model parameters, and saving computational space. At the same time, due to the similar structure of the subnetworks, the application of transfer learning enables parameter sharing among different subnetworks, improving the efficiency of model training and reducing the time cost of training. As shown in the diagram, in side-channel analysis targeting AES-128 encryption, traditional template attack methods decompose the long key into 16 minimum attack units at the byte level, with each unit having a key length of 8 bits. In the template construction phase, 2^8 templates need to be built. In our proposed half-byte attack method, we use a divide-and-conquer approach to break down the key classification problem into minimum attack units with a length of half a byte. Among the 32 attack units, the key search space becomes 2^4. Therefore, the divide-and-conquer strategy based on half-byte length efficiently reduces the key search space, reducing the number of template types from $16 \times 256 = 4096$ to $32 \times 16 = 512$. This approach is particularly beneficial during the training phase of deep learning models as it significantly reduces the data requirements and the difficulty of model training.

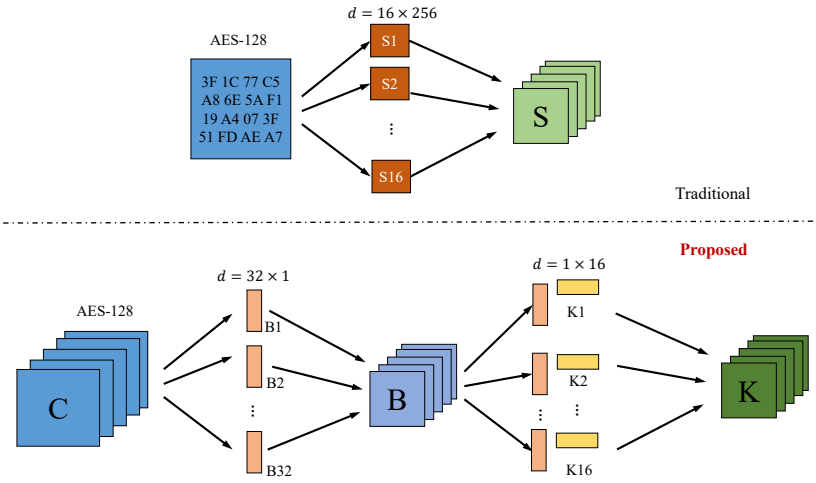


Figure 3. A key decomposition method we proposed.

5. Parallel Analysis Structure of Multiple Symmetric Subnetworks Based on Transfer Learning

The description of transfer learning [41] provides definitions for a domain and a task: A domain, denoted as $D = \{\chi, P(X)\}$, comprises two parts: the feature space χ and the marginal probability distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \chi$. A task, denoted as $T = \{y, f(x)\}$, consists of two parts: the label space y and the target prediction function $f(x)$. The function $f(x)$ can also be viewed as a conditional probability function.

Transfer learning is an important tool in deep learning for addressing the fundamental problem of insufficient training data. It attempts to transfer knowledge from a source domain to a target domain under the assumption that the training and testing data must be independent and identically distributed. This can have significant positive impacts in many domains where improvement is challenging due to limited training data. The construction of the electromagnetic attack model under transfer learning is illustrated in the figure. Since the symmetric encryption algorithms have the same byte encryption structure, the electromagnetic signals generated occupy a similar feature space. Therefore, the data from different sub-datasets exhibit high similarity. We leverage model parameter sharing among different sub-models to enhance the utilization of information.

In the process of applying transfer learning to electromagnetic attacks, we fixed the AES-128 plaintext and most of the key, and iterated through the semi-byte key. We collected the corresponding electromagnetic data as the source domain data and trained the model using ideal state data to obtain an ideal attack template with a smaller amount of data. Considering real-life scenarios where both the plaintext and ciphertext are unknown and only random data of the semi-byte key is available as the target domain, we transferred the ideal deep learning-based attack model to the actual attack model for training. This approach aims to improve the attack speed of the deep learning-based key attack model with a smaller amount of data under the situation of random key and plaintext. In model-based transfer learning, the existing model in the source domain serves as prior knowledge to learn the new model, as shown in Figure 4.

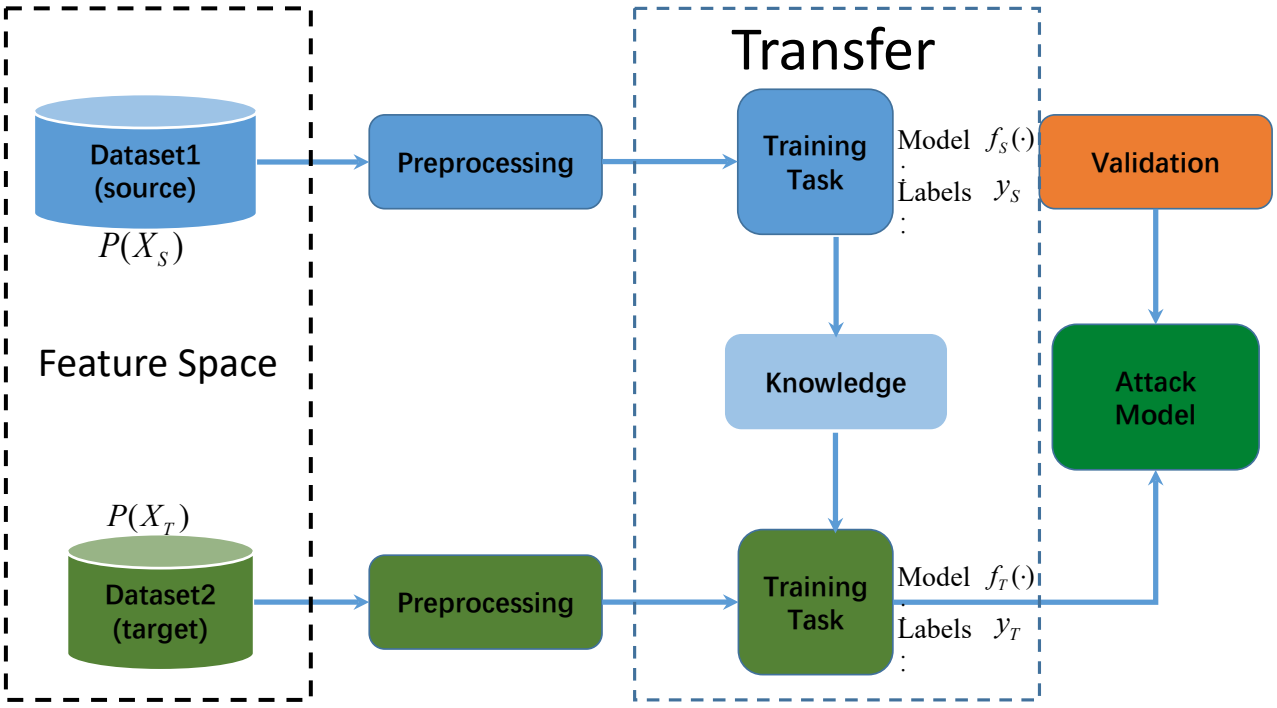


Figure 4. Transfer learning method between the source domain and the target domain.

Then, by calculating the Maximum Mean Discrepancy (MMD) distance between different data sets, we search for the optimal source domain model. In the subsequent

experimental results, we discovered that this method not only reduces the number of templates that need to be constructed and lowers the computational complexity but also allows for the utilization of more training data. As there are multiple data sets, we designed Algorithm 1 to find the optimal transfer path.

The calculation of D_{MMD} is defined by the following formula:

$$D_{MMD}(X_S, X_T) = \left\| \frac{1}{|X_S|} \sum_{x_s \in X_S} \phi(x_s) - \frac{1}{|X_T|} \sum_{x_t \in X_T} \phi(x_t) \right\| \tag{5}$$

the $\phi(\cdot)$ is representation mapping function.

Algorithm 1 Source-Task Match for Symmetrical subnetworks of electromagnetic signals analysis (EM-STM)

Input: $Datasets[\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_n]$, $sigma$
Output: $\mathbf{source - target_{pairs}}$

Batch normalization $Datasets$
for each $i \in [1, n]$ **do**
  $mmd_{min} = +\infty$
  for each $j \in [1, n]$ **do**
    calculate mmd **between** D_i **and** D_j **Equation (5),** $sigma$
    if $mmd < mmd_{min}$ **then**
      $mmd_{min} = mmd$
      $st_{match} = (D_i, D_j)$
    end if
  end for
end for
return $\mathbf{source - target_{pairs}}$

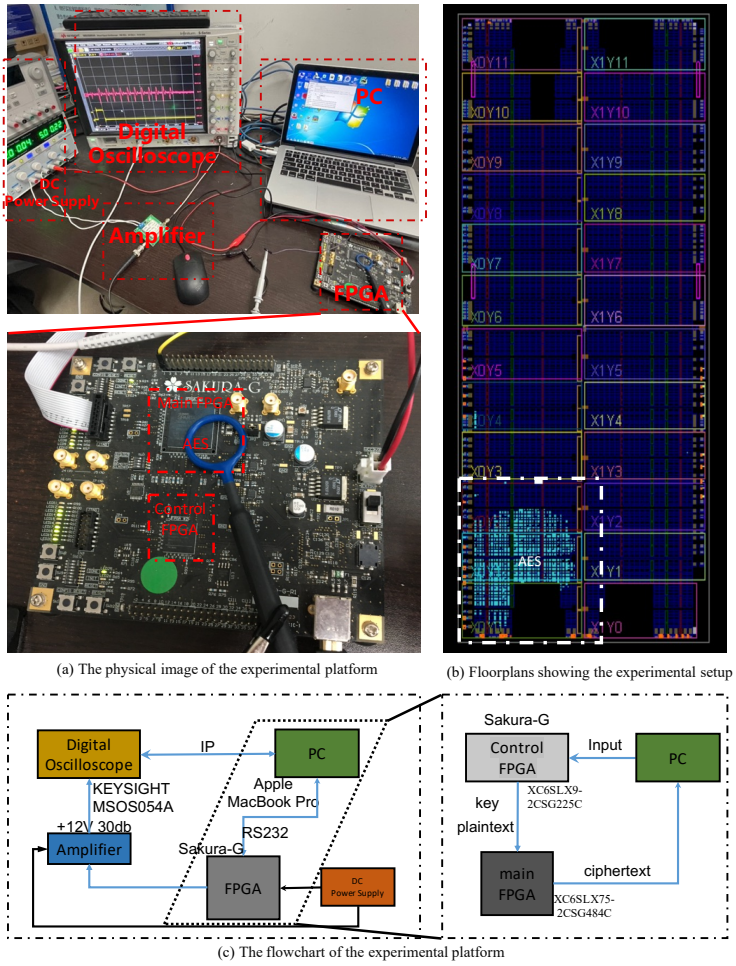
6. Experimental Results

6.1. Experimental Platform

The electromagnetic signal acquisition platform constructed in this paper is shown in Figure 5a,c. The platform consists of a digital oscilloscope, PC, FPGA chip, and electromagnetic probe. The power supply is used to power the FPGA.

The specific process is as follows:Firstly, The PC sends the plaintext and key required by the encryption algorithm to the FPGA. Secondly, the AES-128 encryption algorithm is running in the FPGA, and the electromagnetic probe collects the electromagnetic radiation signal of the near field of the chip at the same time. Thirdly, the electromagnetic probe transmits the collected signal to the oscilloscope, and the oscilloscope displays the signal and saves it. Finally, the oscilloscope sends a signal to the PC, and the PC sends the plaintext and key to the FPGA again, collecting a large number of required electromagnetic leakage signals.

In terms of the specific use of FPGA, we ran our experiments on the widely-used side channel evaluation platform SAKURA-G, which consists of two chips. The main chip is a larger XC6SLX75 for security implementations, controlled by an auxiliary Spartan6 XC6SLX9 [42]. As a proof of concept, we considered the aforementioned AES encryption module as the targeted cryptographic core, implemented in the main FPGA and ran at a frequency of 24 MHz. The AES-128 module is a relatively small implementation, occupying 1002 Flip-Flops and 3189 LUTs. Its footprint in this FPGA is 1% of the total flip flops and 6% LUTs. In the meantime, we show the entire floorplan of the experimental setup in Figure 5b, and the AES core has been labeled.



(a) The physical image of the experimental platform

(b) Floorplans showing the experimental setup

(c) The flowchart of the experimental platform

Figure 5. The physical image of the experimental platform.

6.2. Dataset

In the daily office environment, we set up the FPGA's loop encryption process on the PC and recorded every encryption operation of the AES algorithm. During the experiment, the near-field electromagnetic probe was placed on top of the encryption chip, and its position was kept fixed. Then, we combined the original dataset based on different analysis objectives to obtain the desired dataset. In the data preparation for each sub-model, we divided the traces into training and validation sets, denoted as

$$\begin{cases} \{\mathbf{x}_n^{TR}, y_n^{TR}\}_{n=1}^{N_{TR}} \\ \\ \{\mathbf{x}_n^{VA}, y_n^{VA}\}_{n=1}^{N_{VA}} \end{cases} \tag{6}$$

in which TR stands for training, and VA stands for validation. The Input is $\mathbf{x}_n \in \mathbb{R}^{1\times 10000}$ and the output is $\mathbf{y}_n \in [0, 1, 2, \cdots, 16]$. The training set and test set are mutually independent, and each data record captures the electromagnetic leakage information during one encryption operation of the chip. In multiple encryption operations of the chip, the plaintext is randomly generated, and apart from the manipulated semi-byte that is set for attack purposes, the rest of the key is also random. We randomly display an example of the collected raw data electromagnetic signal as shown in Figure 6, and the corresponding encryption information is illustrated in Table 1. The encryption mode is performed in ECB mode, with a plaintext length and key length of 128 bits, and there is no zero-padding operation involved.

Table 1. Examples of encryption details.

	DataSample		
	Key	**Plaintext**	**Ciphertext**
Dataset	0x3F1C77C5A86E5AF	0xB9D1C48E348FE77	0x42F7EA0BC6AC69F
	119A4073F51FDAEA7	1FA464A77A178FB07	DC282C7A447BE3E55
	0x489DB4B3F317296	0xDCFEAD50D1D9FD0	0xE82201EF806DFA2
	1CC2BCB4ED2E28EB7	8B386EFB08B142F74	28309ED5B271805F2
	0xB303FDCEDB10838	0x4CFDA4C3076157F	0x30F9F5D7DC56A51
	645FEB0626FACAD79	581B71E46E8CB5675	E7D8E030B45810EA2
	0x3E5A862B58B1652	0x253868B79E7C31E	0x5E96B2A5B349FC2
	7F315F97CA30B1E36	0D3BCDBAE9F37E5E5	55B65D4356CB19D1E
	0x164F8C4EFFE0CB3	0x16D62BA4D338FF7	0xA5B7EBE2E8A9FAD
	8192D6DF466C6F21F	BA6B8CF19BAB720E7	123476AB59F0A99E2

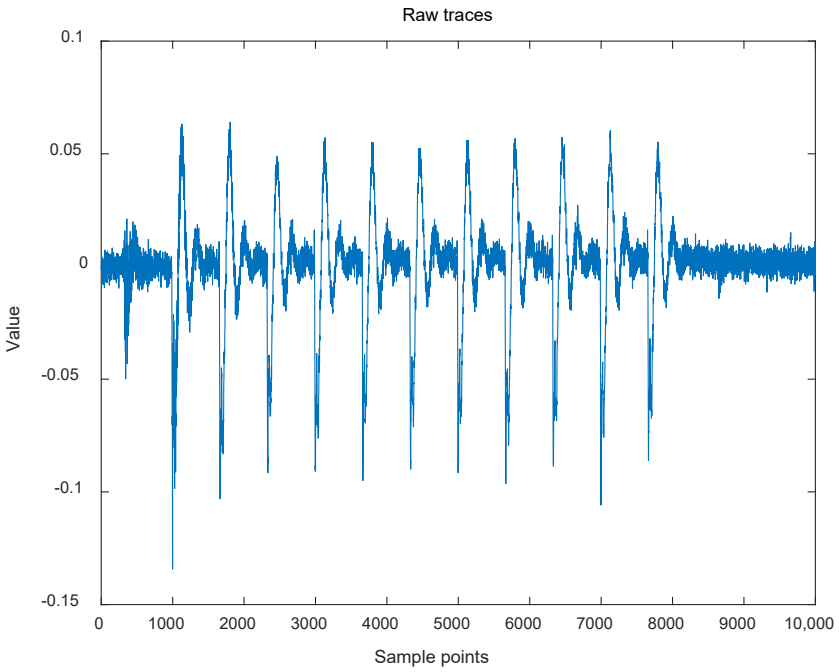


Figure 6. The signal of raw data. Schemes follow the same formatting.

6.3. Result and Discussion

In our work, the deep learning model framework utilized in this paper is based on TensorFlow 1.13. The equipment used during the training is configured as follows: GPU: NVIDIA GeForce GTX 1080 Ti and 2080 Ti. The CPU is Intel(R) Xeon(R) and CPU: E5-2678 v3 @ 2.50 GHz.

In the first part, we compared the experimental results of different network depths in our designed deep residual network model. The purpose of this work was to select the optimal network model considering the available GPU space. In the network structure, we had five different types of residuals. In the 12Layers model, their counts were [1, 1, 1, 1, 1]. In the 18Layers model, their counts were [1, 1, 2, 2, 2]. In the 22Layers model, their counts were [2, 2, 2, 2, 2]. In the 32Layers model, their counts were [3, 3, 3, 3, 3]. The training experiment results on a randomly selected dataset showed that the 22Layers model performed the best with an accuracy of 81.25% as shown in Figure 7, outperforming the other models. Therefore, we chose this network model for further experimental steps.

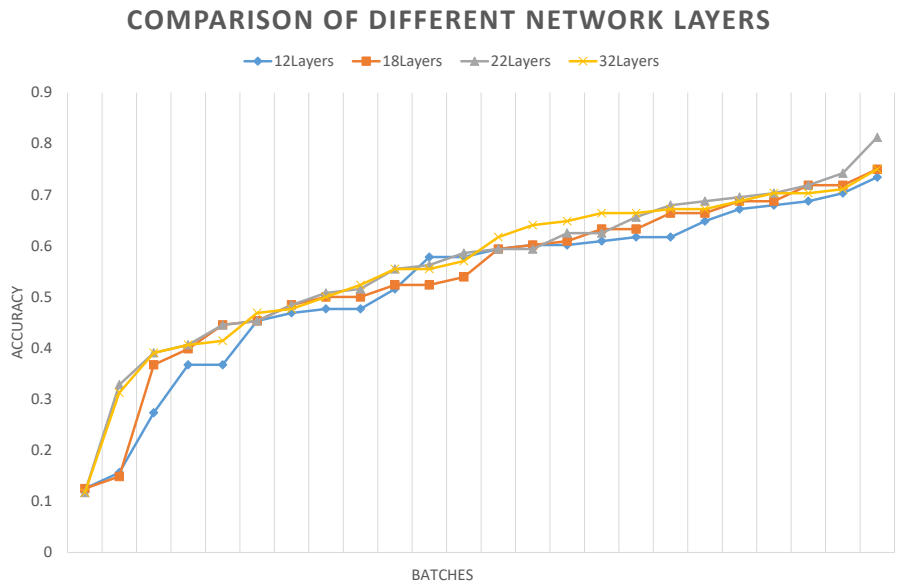


Figure 7. The comparison results of different network layers.

In the second part, we compared the impact of model parameter sharing on the accuracy of electromagnetic signal classification in transfer learning. Using the experimental platform described in Figure 5, we fixed the plaintext and key in the encryption algorithm and obtained a batch of electromagnetic signal data when all the information of the encryption algorithm was known. We considered this signal as ideal attack data. After training the model, we obtained an ideal classification template. Then, based on this model, we performed parameter-sharing training for the actual model we wanted to obtain. The experimental results showed that the network could achieve 100% classification accuracy in the ideal case.

In the comparison of transfer learning methods, for the same training batch size, the network model that underwent early transfer showed higher accuracy compared to the model with weight initialization. The network model without transfer achieved an accuracy of 80.98%, while the network model after transfer reached an accuracy of 87.5%, resulting in an improvement of nearly 7% in accuracy, as shown in Figure 8.

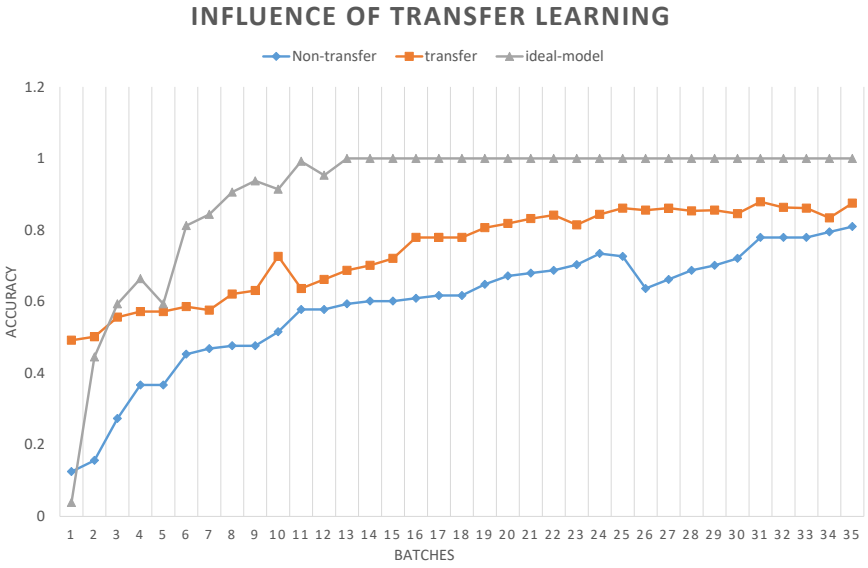


Figure 8. The influence of transfer learning.

In the third part, we validated the use of transfer learning methods for better performance in small-sample datasets. Training deep models typically requires a large amount of data, and overfitting often occurs when dealing with small-sample datasets. We compared four different-sized datasets and found that transfer learning can mitigate the overfitting phenomenon in small-sample scenarios. It partially addresses the issue caused by a small dataset and effectively extracts the value from small-sample data. The specific performance is shown in Table 2. The range from 500 to 5000 represents the number of curves for each class of key labels during model training. 'Initialization' indicates that we directly initialize the model parameters before training. 'Model-transfer' means that we share and adjust our model parameters with other models. In the table, we use the accuracy of the model's training and validation sets as evaluation metrics. The specific calculation method for accuracy is obtained from the computation method in Equation (4).

Table 2. The influence of transfer learning in small-sample datasets.

	500		1000		3000		5000	
	Training	**Verification**	**Training**	**Verification**	**Training**	**Verification**	**Training**	**Verification**
Initialization	100%	41.2% (Overfitting)	100%	43.6% (Overfitting)	77.5%	65.8%	83.2%	83.7%
Model-transfer	70.2%	69.5%	80.4%	77.2%	83.8%	82.5%	84.3%	80.5%

In the fourth part, we used the 22Layers model and combined transfer learning between different sub-models to obtain the key classification results for the complete AES-128 encryption algorithm, as shown in Figure 9, Due to the large number of models, to balance the training time cost and GPU space usage cost, we trained each model for only 20 batches. After simple training, the validation accuracies of all models were as follows: 88%, 91%, 68%, 69%, 66%, 81%, 82%, 84%, 80%, 71%, 65%, 80%, 84%, 79%, 75%, 82%, 81%, 74%, 82%, 78%, 77%, 77%, 64%, 81%, 77%, 75%, 82%, 76%, 84%, 78%, 86%, and 81%. Model-2 achieved the highest accuracy of 91%, and the overall average accuracy was 78%. This demonstrates the generalization effect of the model.

Under the same experimental conditions, we compared the performance of three classic networks: SVM [9], VGG-16 [10], and Inception-V2 [11]. The comparative results obtained are shown in the following diagram.

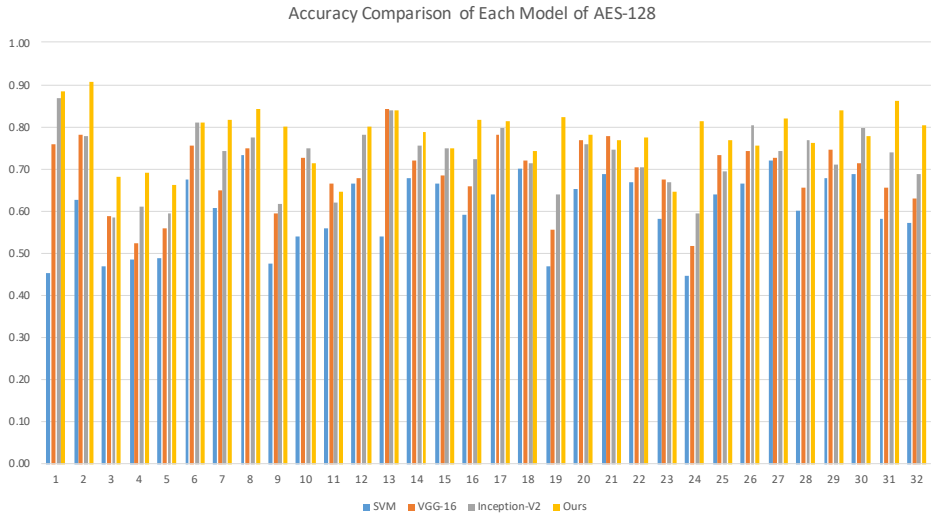


Figure 9. The attack accuracy comparison of each model of AES-128.

7. Conclusions

In this paper, we designed a key classification framework based on deep transfer learning. We proposed an effective key-oriented side-channel analysis experimental design and devised a key classification method based on semi-byte length, utilizing the "divide and conquer" approach. We employed an adaptive residual convolutional neural network to extract the side-channel electromagnetic signal characteristics, achieving accurate end-to-end mapping between the electromagnetic signal and the unique key. By leveraging a parallel analysis structure with symmetric sub-networks and employing transfer learning methods, we optimized network parameters. This approach improved processing efficiency at different attack positions with varying keys, thereby reducing the key enumeration range. Moreover, it reduced the amount of training data samples and demonstrated better application for small-sample attack scenarios.

Author Contributions: Conceptualization, X.C., H.Z. and Y.W.; methodology, X.C. and X.F.; validation, X.C., Y.W. and D.W.; formal analysis, X.C. and H.Z.; investigation, X.F. and X.C.; writing—original draft preparation, X.C.; writing—review and editing, D.W., F.F. and L.S. ; funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62071057, 62302036), Aeronautical Science Foundation of China (2019ZG073001), BUPT innovation and entrepreneurship support program (2023-YC-A165).

Institutional Review Board Statement: Not applicable,

Informed Consent Statement: Not applicable,

Data Availability Statement: Data sharing is available by emailing the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest

References

1. Ouladj, M.; Guilley, S. *Side-Channel Analysis of Embedded Systems: An Efficient Algorithmic Approach*; Springer: Cham, Switzerland, 2021. [CrossRef]
2. Jiang, M. Analysis and Mitigation of EM Side-Channel Attacks on Chip-to-Chip Interconnects. Ph.D. Thesis, University of Manchester, Manchester, UK, 2023.
3. Kocher, P.C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Proceedings of the Advances in Cryptology—CRYPTO'96: 16th Annual International Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 104–113.
4. Ahmed, A.A.; Hasan, M.K.; Islam, S.; Aman, A.H.M.; Safie, N. Design of Convolutional Neural Networks Architecture for Non-Profiled Side-Channel Attack Detection. *Elektron. Elektrotechnika* **2023**, *29*, 76–81. [CrossRef]
5. Costes, N.; Stam, M. Pincering SKINNY by Exploiting Slow Diffusion: Enhancing Differential Power Analysis with Cluster Graph Inference. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2023**,*4*, 460–492. [CrossRef]
6. Lerman, L.; Poussier, R.; Markowitch, O.; Standaert, F.X. Template attacks versus machine learning revisited and the curse of dimensionality in side-channel analysis: extended version. *J. Cryptogr. Eng.* **2018**, *8*, 301–313. [CrossRef]
7. Loiseau, A.; Lecomte, M.; Fournier, J.J. Template Attacks against ECC: Practical implementation against Curve25519. In Proceedings of the 2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), San Jose, CA, USA, 7–11 December 2020; pp. 13–22.
8. Randolph, M.; Diehl, W. Power side-channel attack analysis: A review of 20 years of study for the layman. *Cryptography* **2020**, *4*, 15. [CrossRef]
9. Suthaharan, S. Support Vector Machine. In *Machine Learning Models and Algorithms for Big Data Classification*; Integrated Series in Information Systems; Springer: Boston, MA, USA, 2016; Volume 36, pp. 207–235. . [CrossRef]
10. Tammina, S. Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *Int. J. Sci. Res. Publ.* **2019**, *9*, 143–150. [CrossRef]
11. Bose, S.R.; Kumar, V.S. Efficient inception V2 based deep convolutional neural network for real-time hand action recognition. *IET Image Process.* **2020**, *14*, 688–696. . [CrossRef]
12. Timon, B. Non-profiled deep learning-based side-channel attacks with sensitivity analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2019**, 107–131. [CrossRef]
13. Affonso, C.; Rossi, A.L.D.; Vieira, F.H.A.; de Leon Ferreira, A.C.P. Deep learning for biological image classification. *Expert Syst. Appl.* **2017**, *85*, 114–122. [CrossRef]

14. Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Deep learning for hyperspectral image classification: An overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709. [CrossRef]
15. Çayır, A.N.; Navruz, T.S. Effect of dataset size on deep learning in voice recognition. In Proceedings of the 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 11–13 June 2021; pp. 1–5.
16. Bae, H.S.; Lee, H.J.; Lee, S.G. Voice recognition based on adaptive MFCC and deep learning. In Proceedings of the 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), Hefei, China, 5–7 June 2016; pp. 1542–1546.
17. Quy, T.N.; Van Quyen, P. Experimental results of electromagnetic analysis on smartcard. *J. Sci. Technol. Inf. Secur.* **2023**, 5–14.
18. Zhang, H.X.; Liu, J.; Xu, J.; Zhang, F.; Cui, X.T.; Sun, S.F. Electromagnetic radiation-based IC device identification and verification using deep learning. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 206. [CrossRef]
19. Hong, S.; Davinroy, M.; Kaya, Y.; Locke, S.N.; Rackow, I.; Kulda, K.; Dachman-Soled, D.; Dumitraş, T. Security Analysis of Deep Neural Networks Operating in the Presence of Cache Side-Channel Attacks. *arXiv* **2020**, arXiv:1810.03487.
20. Das, D.; Golder, A.; Danial, J.; Ghosh, S.; Raychowdhury, A.; Sen, S. X-DeepSCA: Cross-Device Deep Learning Side Channel Attack. In Proceedings of the 56th Annual Design Automation Conference 2019, Las Vegas, NV, USA, 2–6 June 2019; ACM: Las Vegas, NV, USA, 2019; pp. 1–6. . [CrossRef]
21. Li, L.; Ou, Y. A deep learning-based side channel attack model for different block ciphers. *J. Comput. Sci.* **2023**, *72*, 102078. [CrossRef]
22. Wang, R.; Wang, H.; Dubrova, E. Far Field EM Side-Channel Attack on AES Using Deep Learning. In Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security, Virtual Event, 13 November 2020; pp. 35–44. [CrossRef]
23. Chabanne, H.; Danger, J.; Guiga, L.; Kühne, U. Side channel attacks for architecture extraction of neural networks. *CAAI Trans. Intell. Technol.* **2021**, *6*, 3–16. [CrossRef]
24. Hettwer, B.; Gehrer, S.; Güneysu, T. Applications of machine learning techniques in side-channel attacks: A survey. *J. Cryptogr. Eng.* **2020**, *10*, 135–162. [CrossRef]
25. Hospodar, G.; Gierlichs, B.; De Mulder, E.; Verbauwhede, I.; Vandewalle, J. Machine learning in side-channel analysis: A first study. *J. Cryptogr. Eng.* **2011**, *1*, 293. [CrossRef]
26. Weissbart, L.; Picek, S.; Batina, L. One Trace Is All It Takes: Machine Learning-Based Side-Channel Attack on EdDSA. In *Security, Privacy, and Applied Cryptography Engineering*; Series Title: Lecture Notes in Computer Science; Bhasin, S., Mendelson, A., Nandi, M., Eds.; Springer: Cham, Switzerland, 2019; Volume 11947, pp. 86–105. [CrossRef]
27. Picek, S.; Heuser, A.; Jovic, A.; Ludwig, S.A.; Guilley, S.; Jakobovic, D.; Mentens, N. Side-channel analysis and machine learning: A practical perspective. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 4095–4102.
28. Sun, S.; Zhang, H.; Cui, X.; Dong, L.; Fang, X. Electromagnetic side-channel hardware trojan detection based on transfer learning. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *69*, 1742–1746. [CrossRef]
29. Lee, Y.; Jun, S.; Cho, Y.; Han, W.; Moon, H.; Paek, Y. Precise Extraction of Deep Learning Models via Side-Channel Attacks on Edge/Endpoint Devices. In *Computer Security—ESORICS 2022*; Series Title: Lecture Notes in Computer Science; Atluri, V., Di Pietro, R., Jensen, C.D., Meng, W., Eds.; Springer: Cham, Switzerland, 2022; Volume 13556, pp. 364–383. [CrossRef]
30. Maghrebi, H.; Portigliatti, T.; Prouff, E. Breaking Cryptographic Implementations Using Deep Learning Techniques. In *Proceedings of the Security, Privacy, and Applied Cryptography Engineering, Lecture Notes in Computer Science, Hyderabad, India, 14–18 December 2016*; Carlet, C., Hasan, M.A., Saraswat, V., Eds.; Springer: Cham, Switzerland, 2016; pp. 3–26.
31. Maghrebi, H. Deep learning based side channel attacks in practice. *Cryptol. ePrint Arch.* **2019**, *preprint*.
32. Wei, L.; Luo, B.; Li, Y.; Liu, Y.; Xu, Q. I Know What You See: Power Side-Channel Attack on Convolutional Neural Network Accelerators. In Proceedings of the 34th Annual Computer Security Applications Conference, San Juan, PR, USA, 3–7 December 2018; pp. 393–406. [CrossRef]
33. Wang, H.; Brisfors, M.; Forsmark, S.; Dubrova, E. How diversity affects deep-learning side-channel attacks. In Proceedings of the 2019 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC), Helsinki, Finland, 29–30 October 2019; pp. 1–7.
34. Picek, S.; Perin, G.; Mariot, L.; Wu, L.; Batina, L. Sok: Deep learning-based physical side-channel analysis. *ACM Comput. Surv.* **2023**, *55*, 1–35. [CrossRef]
35. Lu, X.; Zhang, C.; Cao, P.; Gu, D.; Lu, H. Pay attention to raw traces: A deep learning architecture for end-to-end profiling attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2021**, *3*, 235–274. [CrossRef]
36. Zhou, Y.; Standaert, F.X. Deep learning mitigates but does not annihilate the need of aligned traces and a generalized ResNet model for side-channel attacks. *J. Cryptogr. Eng.* **2020**, *10*, 85–95. . [CrossRef]
37. Ahmed, A.A.; Hasan, M.K. Design and Implementation of Side Channel Attack Based on Deep Learning LSTM. In Proceedings of the 2023 IEEE Region 10 Symposium (TENSYMP), Canberra, Australia, 6–8 September 2023; pp. 1–6.
38. Zhang, L.; Xing, X.; Fan, J.; Wang, Z.; Wang, S. Multilabel deep learning-based side-channel attack. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2020**, *40*, 1207–1216. [CrossRef]
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. *arXiv* **2016**, arXiv:1603.05027.
40. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.

41. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; Part III 27; Springer: Berlin/Heidelberg, Germany, 2018; pp. 270–279.
42. Schellenberg, F.; Gnad, D.R.; Moradi, A.; Tahoori, M.B. An inside job: Remote power analysis attacks on FPGAs. *IEEE Des. Test* **2021**, *38*, 58–66. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.