Side-channel Attack Using Word Embedding and Long Short Term Memories

Zixin Liu[1], Zhibo Wang[2] and Mingxing Ling[1,*]

[1]*State Key Laboratory of Nuclear Resources and Environment East China University of Technology Nanchang 330013, Jiangxi, China*
[2]*Software college, East China University of Technology Nanchang 330000, China*
E-mail: paololew@ecut.edu.cn
[*]*Corresponding Author*

Received 13 August 2021; Accepted 18 August 2021;
Publication 03 January 2022

Abstract

Side-channel attack (SCA) based on machine learning has proved to be a valid technique in cybersecurity, especially subjecting to the symmetric-key crypto implementations in serial operation. At the same time, parallel-encryption computing based on Field Programmable Gate Arrays (FPGAs) grows into a new influencer, but the attack results using machine learning are exiguous. Research on the traditional SCA has been mostly restricted to pre-processing: Signal Noisy Ratio (SNR) and Principal Component Analysis (PCA), etc. In this work, firstly, we propose to replace Points of Interests (POIs) and dimensionality reduction by utilizing word embedding, which converts power traces into sensitive vectors. Secondly, we combined sensitive vectors with Long Short Term Memories (LSTM) to execute SCA based on FPGA crypto-implementations. In addition, compared with traditional Template Attack (TA), Multiple Multilayer Perceptron (MLP) and Convolutional

Journal of Web Engineering, Vol. 21_2, 285–306.
doi: 10.13052/jwe1540-9589.2127

Neural Network (CNN). The result shows that the proposed model can not only reduce the manual operation, such as parametric assumptions and dimensionality setting, which limits their range of application, but improve the effectiveness of side-channel attacks as well.

Keywords: Side-channel attack, word embedding, long short term memories.

1 Introduction

Side-channel attack (SCA) leaks sensitive cryptographic information consisting of instantaneous power consumption, executing encryption time, physical power leaking. It challenges the security of hardware devices, takes up a wide range of threats.

Simple Power Analysis (SPA) is the first tool to execute real side channel attack, which tries to find out the relationship between traces and cryptographic positions [1]. It depends on the interpretation of the power consumption. What is more, SPA could yield info about device's operation as well as key material. Thus, many cryptosystems are catching on this: they need to be revised to prevent SPA, may need to take measures to protect security protocols and algorithms. Differential Power Analysis (DPA) [2] utilizes a more advanced and confidential statistical research by establishing a theoretic model subjected to power consumption for each subkey [3]. The joint likelihood of the observed power consumption for each research model is applied for predicting the subkey. Suresh Chari present Template Attack (TA), maybe the strongest form of SCA [4]. It focuses on the precisely modelling noisy, at the same time, leads the development from traces analysis towards template classification, which provides a foundation for hodiernal machine learning. We need to know that the role of side-channel analysis in the field of hardware security is not necessarily to completely break out the key or sub-key. If it can help us know that a certain hardware or chip has been leaked, then it has fulfilled one of its missions. Werner Schindler proposed the Stochastic Model (SM) [5], which approximates the real leakage function limited an adoptable vector subspace by using parametric model. As for key extraction, research apply minimum principle that solely utilizes deterministic data dependencies and maximum likelihood principle that additionally incorporates the characterization of the noise revealed during training stages.

Those traditional attack methods mostly performed on generating templates, especially different keys subjected to multi-variate Gaussian distribution Simulink of the manually selected Points-of-Interests (POIs), Signal Noise Ratio (SNR).

Recently, the hardware security research has transferred attention to the machine learning (ML) based profiled attacks and ciphers classification. Support Vector Machine (SVM) [6], Random Forest (RF) [7], and other deep learning [8] based attacks not only perform valid attacks but reduce the concern on POIs.

2017, Eleonora Cagli proposed a point-to-point profiling attack method based on the application of Convolutional Neural Networks (CNN) [9–15]. It does not need a manually traces realignment nor accurate selection of POIs. Moreover, they firstly utilized classical data augmentation tool to improve attack performance. In 2021 year, lots of algorithms demonstrated or benchmarked for image classification which are typically working with 1-dimensional (1-D) data, Yoo-Seung Won proposed a novel criterion for attack success or failure based on statistical confidence level rather than determination of a correct key using a ranking system and Binarized Neural Network (BNN) learning method that relies on a BNN's natural properties to improve variables [16]. J Wei [17], in 2020 year, firstly combined the LSTM from Hochreiter S [18] with side channel attack, then Paguada S in 2021 improve the efficiency of LSTM on side channel attack [19]. In addition, many domestic and foreign scholars have proposed rich power curve preprocessing methods based on deep learning to assist training [20, 21].

In recent years, machine learning has enriched the tools of SCA, but it is worth to mention that most previously presented attacks were based on the serial implementation of cryptographic algorithm, and there are sparse on hardware attacks aimed at the parallel encryption implementation of cryptographic algorithm on FPGA.

In addition, the pre-processing of power trace and the selection of POIs are also the key points in side-channel attacks. DPAv2 is the data of SCA against FPGA devices which operate in parallel with cryptographic algorithms. Figure 1 shows the development of DPA Contest.

The results proposed in this paper enrich the previous traces treatment in natural language processing directions.

Our main contributions are:

On one hand, we presented to use word embedding methods instead of SNR or POIs. The power value corresponding to each sampling point in a fixed clock is vectorized, and the word vector is used to replace the power.

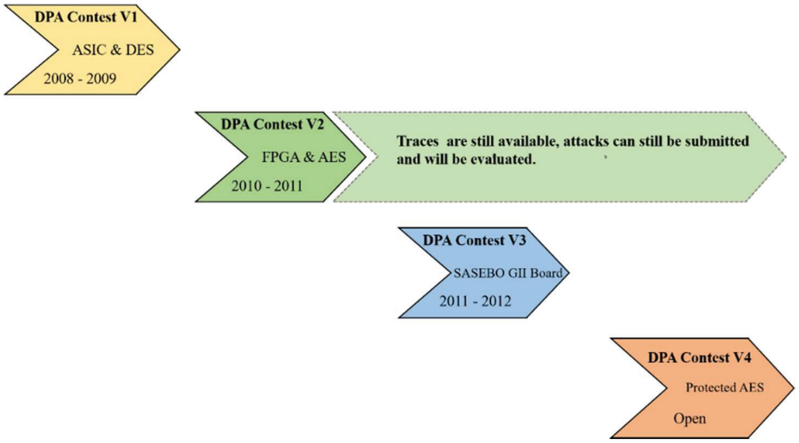

Figure 1 The development of DPA contest.

Mathematical modelling is performed using the word bag technique in the direction of natural language processing.

On the other hand, Combined word embedding with Long Short Term Memories and based on the timing series characteristics of the sampling points in the fixed clock in the DPAV2 dataset, LSTM[11] solves the gradient vanishing and gradient explosion problems in long time series training, improves the attack efficiency.

2 Background

2.1 Difficultiys on SCA Based on FPGA Parallel Implementation

As for serial implementation of AES-128 encryption algorithm, after simple filtering of the traces, it can be easily identified by determining the clock in the first round of 16 S-box operations. The realization of each S-box is based on the multiplication inverse of finite field and affine transformation, and the calculation of 16 S-boxes is independent of each other. This is also the reason why we choose the intermediate value of the input and output of the S-box as attack object in SCA. However, we found that traces consumption of a parallel encryption algorithm based on FPGA were extremely smooth, data for this study were collected using DPAv2, requiring no manual alignment and further filtering because of official operation. Therefore, in the case of parallel operation, it increases the difficulty of attack.

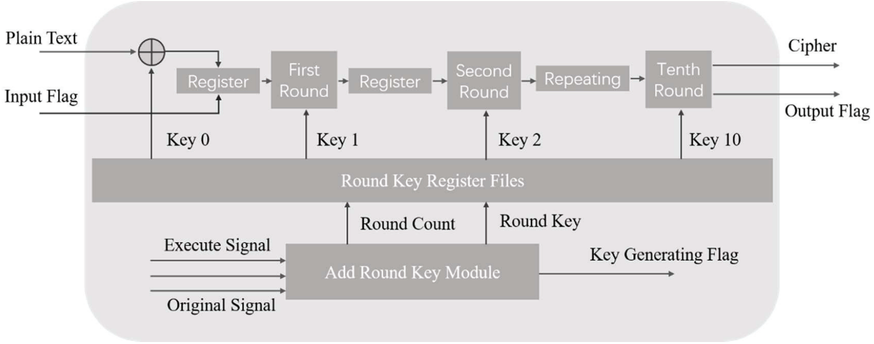

Figure 2 AES-128 implementation process.

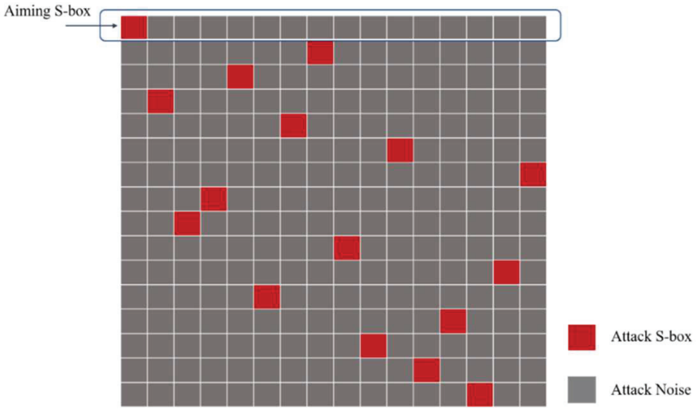

Figure 3 AES-128 S-box attacking noisy analysis.

Secondly, Figure 2 presents the pipeline design of the AES algorithm with a key length of 128 bits. Currently, there are 10 rounds of AES calculation. Byte substitution can be performed by lookup tables. Each byte needs an S-box of 8 width and 256 depths, and the grouping length of AES is 128 bits. Thus, if one clock cycle is needed to complete subbytes, a total of 16 parallel S-boxes are needed. Therefore, when we look for vulnerable points to attack, there are 16 S-boxes, only one of which has input and output values that we need, and the remaining 15 S-boxes will be noise-blocking side-channel attacks. Figure 3 depicts the AES-128 S-box attacking noisy analysis.

At last, we try to use stochastic model to attack the parallel implementation based on FPGA, it means we need set up a noisy model. However, it is different from traditional template attacks. Stochastic attacks are no longer

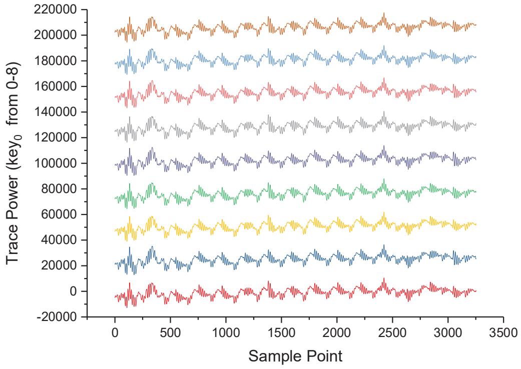

Figure 4 DPAv2 traces distribution (Axis = 1).

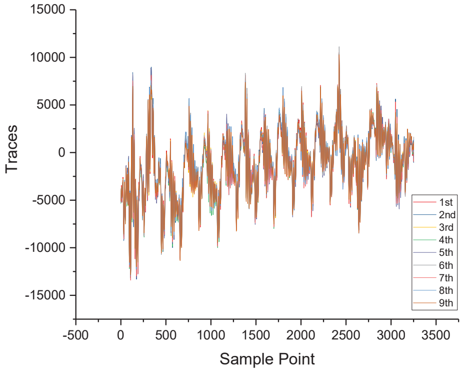

Figure 5 DPAv2 traces (Axis = 0).

the creation of multiple templates at once, but an attempt to train a probability recognizer based on a noise model to predict or guess the joint probability which one is the correct subkey. Meanwhile, when the joint probability value is low, we will try to add a small value to prevent overflowing. Figures 4 and 5 show the alignment of traces. The following training vector of the key probability recognizer contains all sorts of possible noise vectors of the selected intermediate value. $I_t(\mu, k)$ has a scheduling recognition function for the generated noise vector, where μ denotes plaintext, k depicts the subkey. The recognizer is used in the attack step to calculate the joint probability subjected to related attack data. The result of the guessing-attack is the subkey with the highest probability.

We use all the training data for training. When establishing a mathematic model for the leaking data, we should distinguish the data as the leak based on time series. Therefore, time node becomes an important reference for us to select POIs. Using the bit energy conversion coefficient vector β_i related to time is the innovate improvement.

The stochastic model assumes that the energy consumption at time t includes two parts: the useful part of the data and the white noise:

$$I_t(\mu, k) = h_t(\mu, k) + R_t \tag{1}$$

Among them, $I_t(\mu, k)$ is the energy consumption generated by the plaintext μ and the key k at the time of the power consumption curve, including data-related energy consumption $h_t(\mu, k)$ and noise R_t. If we obtain the signal data from the serial hardware chip, then different γ subkey involving S-boxes correspond to different operating intervals, and it is easy to analyse the approximate range of the existence of interest points. The stochastic model further assumes that $h_t(\mu, k)$ is a linear combination of data bit energy consumption:

$$h_t(\mu, k) = \sum_{i=1}^{\gamma} \beta_i g_i(\mu, k) \tag{2}$$

Among them, $g_i(\mu, k)$ is the selection function, which denotes the ith bit of an intermediate value generated in the selected encryption process (for example, the ith bit of the S-box output). β_i is the bit energy conversion coefficient. When the attack is evacuated on an FPGA device running in parallel with the encryption algorithm, the 16 parallel operations will bring extra 15 parameters. This equation denotes the stochastic attack model in serial implementation. Suppose we use such equation to set up suitable model for parallel execution, then it should be revised to that equation:

$$h_t(\mu, k) = \sum_{s=1}^{16} \sum_{i=1}^{\gamma} \beta_i g_i(\mu, k) \tag{3}$$

Where the s denotes the sth S-box, then it results into explosive growth.

The above difficulties have brought obstacles to the traditional side channel attack methods, which also began to drive the shift of research direction.

2.2 Preprocessing of Side-Channel Attack

The physical quantities such as the power consumption curve collected by the side channel analysis contain a large amount of redundant information.

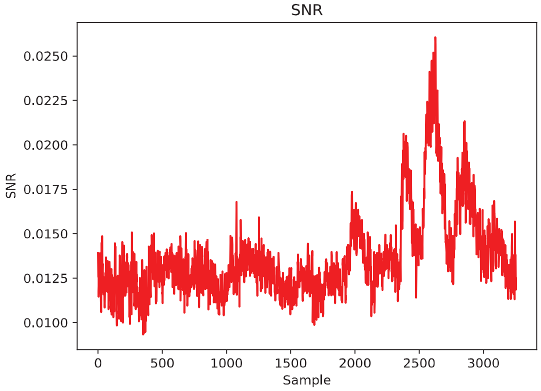

Figure 6 DPAv2 traces (Axis = 0).

How to extract feature points from it is one of the key issues for the success of the side channel attack. Since the template attack was proposed, the commonly used feature extraction techniques are mainly based on specific statistics, including mean difference, sum of square difference, sum of T difference, Pearson correlation coefficient, signal-to-noise ratio, variance and mutual information, etc. Principal component analysis [20] can be used to project electromagnetic and power consumption curves into a low-dimensional subspace to select key features. In addition, alignment is more cumbersome. The existing curve alignment technology can be divided into static alignment and dynamic alignment.

In 2017, CAGLI et al. proposed that if the convolutional neural network is regarded as a feature extractor [21], it can extract the attribute features with invariant translation, so the collected curves can be broken through the cryptographic system with random delay protection without aligning. In the above, we discussed the power consumption curve intercepted from the FPGA hardware device of parallel encryption using the probe, no further alignment is required, so I will not repeat the alignment pre-processing method in this article. Aiming at the pre-processing method of FPGA side-channel attack, this article gives the signal-to-noise ratio of training dataset and attack dataset of FPGA encryption equipment from the perspective of SNR. Figures 6 and 7 separately depict the SNR of Template dataset and Public dataset. Figure 7 uses Butterworth filtering method.

The second is the SNR of the energy trace, $\mathrm{SNR} = \mathrm{Var}(E(signal))/E(Var(signal))$, and the SNR is used to reflect the leakage of the power curve. In this article, when we calculate the signal-to-noise ratio of the power

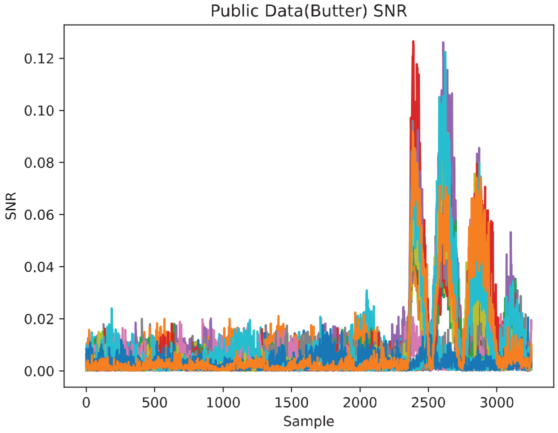

Figure 7 DPAv2 traces (Axis = 0).

consumption curve, because we need to use the leakage points of the energy trace curve (the sampling points in the original data that are strongly related to the key), we can use an example: the first round of AES-128. The output value of an S-box can be expressed as $value_{first-round} = Sbox(p_1 \bigoplus k_1)$. If there is a formula cleaning to see that it is affected by the key k_1, then if we calculate the energy trace and the process output When the SNR of a certain point is found to be higher than the value of the corresponding SNR of other sampling points, the corresponding sampling point is likely to be encrypted according to the key k_1.

2.3 Word Embedding Model

In this paper, word vector has a better semantic characteristic and is always used for expressing the characteristics of words. The value of each dimension of the word vector represents a feature with a certain semantic and grammatical interpretation. Therefore, each dimension of the word vector can be called a word feature [22], Word vectors have varieties of shapes, and distributed representation is also one of them. As we know, a distributed representation is a dense, low-dimensional real-valued vector. Each dimension of the distributed representation denotes a potential feature of the word, which captures useful syntactic and semantic characteristics. The word distributed in the distributed representation embodies the characteristic of the word vector: the different syntactic and semantic features of the word are distributed to each of its dimensions to represent it.

In 2003, Bengio et al. proposed a natural language estimation model Neural Network Language Model which is based on a three-layer neural network [23], It can help to predict the probability that the next word is w_t, in a certain context, that is $p(w_t = i|context)$, and the word vector is byproduct of its training. NNLM generates a corresponding vocabulary V according to the corpus C. Each word in V corresponds to a number i. To determine the parameters of the neural network, training samples need to be constructed through the corpus and used as the input of the neural network. The sample construction process is, for any word w_i in C, get its context $(context(w_i)$, selected words from 1st to $(n-1)th$ to get a tuple $(context(w_i), w_i)$, then use this tuple as the input of the neural network for training.

Word2vec is an implementation of the model proposed by Mykolaiv et al. [24], which can be used to train word vectors quickly and effectively. It includes an input layer, a projection layer and an output layer. We select CBOW model to use the context to predict the current word, because of time series encryption studied in this article, we regard Hamming weight, the current output value of the S-box, as the target to predict.

Each trace has a power value, set the max value be the dimension of word bag, a set of trace will be transferred into $[3253, max(Trace\_value)]$ matrix, then vector the traces, which have been converted. Figure 8 shows traces converting process.

3 Design and Implementation

3.1 Word Embedding

In recent years, word embedding has attracted great attention in the field of natural language processing as a method of mining the deep-level related semantics of words. Word embedding is mainly based on the core idea of words with similar contexts and similar semantics. Words or words are embedded into another special vector space, so that words with similar semantics have similar directions in the vector space. With the help of word embedding technology, the object to be aligned can be transformed into a low-dimensional vector composed of real numbers. Since the vector reflects the contextual semantic feature information of the object, the vector can be used to measure the correlation more accurately between the objects. Therefore, the alignment method based on word embedding can learn the in-depth semantic information of words from the corpus, thereby effectively improving the alignment accuracy.

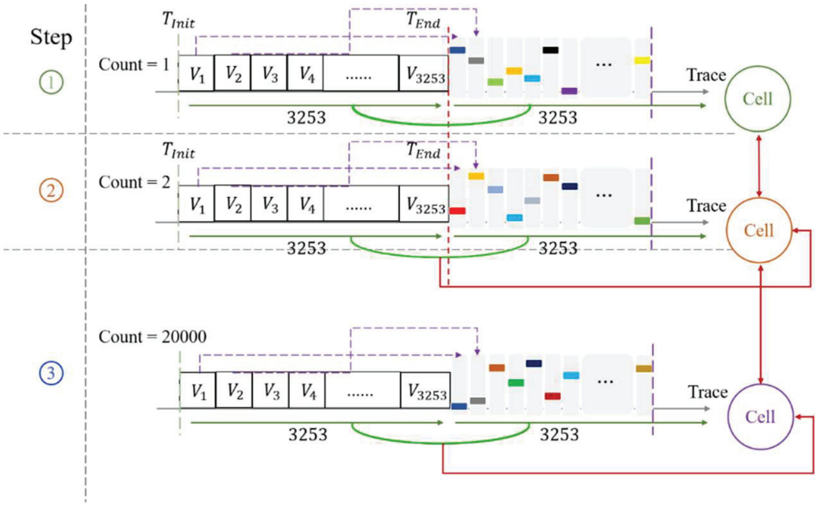

Figure 8 Traces to vectors.

In the above, we have introduced the traditional side-channel attack preprocessing method, but in the actual attack, the training data set and the attack data set are different in signal leakage and noise distribution, etc., It results into the artificial detection of the signal-to-noise ratio cannot fully detect the leakage.

3.2 Word Embedding With LSTM

In this paper, we present an improved method by using word embedding to transfer the power traces value into vector, then send the vector to LSTM model training vectors. As we known, LSTM model is based on RNN(Recurrent Neural Network). RNN use deterministic transition from previous state to current condition by using lots of cells. The deterministic state transition is a function h_t^l, Equations (4) and (5) [25] substitute for the old backpropagation process.

$$RNN{:}h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l \tag{4}$$

$$h_t^l = f(T_{n,n}h_t^{l-1} + T_{n,n}h_{t-1}^l), \quad where\ f \in \{sigm, tanh\} \tag{5}$$

The LSTM has complicated dynamics that allow it to easily memorize information for those expanded timesteps.

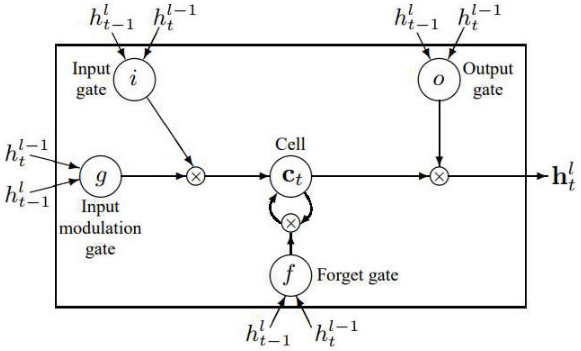

Figure 9 RNN.

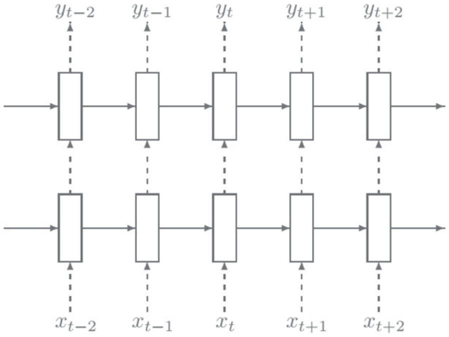

Figure 10 RNN original link.

The LSTM is stored in a vector of memory cells $c_t^l \in R^n$. Although more and more LSTM structures that differ in connectivity structure and activation functions, all architectures have explicit memory cells for storing useful data for long periods of time. The LSTM can decide to overwrite the memory cell, retrieve it, or keep it for the next time step. Figure 9 depicts the LSTM.

At the same time, we apply dropout to LSTMs in a way preventing overfitting. The other reason why we use dropout is: suppose we attack the intermediate value, which denotes Hamming Weight of the first round S-box output. The sampling points we need to select should be within first round S-box operation. From the sampling points where we finished first round S-box operation to 3253th sampling points, all these points prove pointless even hinder our attack goal. Similarly with the last round input or output of S-box operation. Figure 10 shows the memory stage without dropout, Figure 11 depicts the new process with dropout.

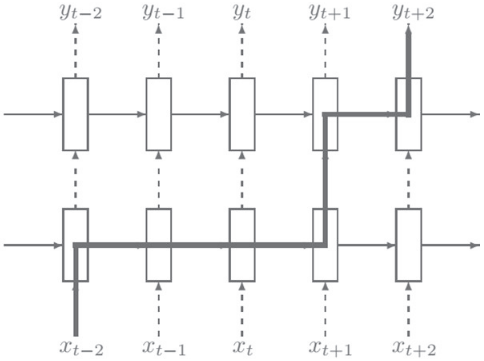

Figure 11 RNN with dropout.

3.3 Evaluation Model

The number of traces measurement is limited by the attacker's ability on monitoring the cryptographic device. Considering reconstruct our attack experiment, we will apply DPAv2 dataset to execute implementation. We transfer the DPAv2 dataset into h5py formation for conveniently deploy. The dataset has been divided into two datasets: DPAV2 Template Dataset.h5 and DPAV2 Public Dataset.h5. The first dataset consists of:

plaintext(1000000,16), *first round key*(16,1), *last round key*(16,1), *ciphers* (1000000,16), and *traces*(1000000,3253).

The other dataset includes 32 subkeys:

plaintext(20000,16), *first round key*(16,1), *last round key*(16,1), *ciphers* (20000,16), and *traces*(20000,3253), where the (...) contains a matrix of rows and columns.

Now the problems are "what about my adversary?", "how to evaluate the efficiency?"

In this experiment, we employ GE(Guessing Entropy) [14] to evaluate the attack efficiency. During the process of training data, we get the possible curve data corresponding to each hamming weight. Then when we use the attack data, we determine the possible values of the top three maximum probabilities of the sub-key in turn according to the way of finding the maximum likelihood function and joint probability. Then we do the violent combination, and since we already know the plaintext, we encrypt the violent combination and compare it with the Hamming weight of the ciphertext and the median. If the ciphertext and hamming weight match the actual data, the attack was successful. In fact, we determine an average location for each attack (starting with the subkey dataset 1000–2000). Add 1000–2000 pieces of corresponding sub-key attack data to trace data successively, and then cycle successively to get the final average position. We use a flexible metric as a goal to the attack efficiency.

Let S be the target hamming weight class discrete variable of SCA and s be a realization of this variable. Let $X_q = [X_1, X_2, X_3 \ldots X_q]$ denote a vector of variables containing a sequence of inputs to the target, $x_q = [x_1, x, x_3 \ldots x_q]$ denotes a realization of this vector. Let O_q represent a random vector denoting the side-channel observations generated with q traces. $O_q = [O_1, O_2, O_3 \ldots O_q]$ be a realization of this random vector. Each output of the leakage function O_q corresponding to the input vector x_q. Let $\Pr[s|O_q]$ denote the conditional probability of a key class s given a leakage O_q. We define the conditional entropy matrix as Equation (6):

$$Z^q_{s,s^*} = -\sum_{O_q} \Pr[O_q|s] \cdot log_2 \Pr[s^*|O_q] \tag{6}$$

where s and s^* respectively denote the correct target classification and a candidate out of the S possible ones. We can derive Shannon's conditional entropy using Equation (7):

$$Z[S|O_q] = -\sum_{s} \Pr[s] \sum_{O_q} \Pr[O_q|s] \cdot log_2 \Pr[s^*|O_q] = E_s(Z^q_{s,s^*}) \tag{7}$$

It directly yields the mutual information: $I(S, O_q) = Z[S] - Z[S|O_q]$. Note that the inputs and outputs of an abstract computer are generally given to the side-channel adversary. Therefore, it is implicitly a computational type of entropy that is proposed to evaluate the physical leakages.

4 Evaluation

Here we use DPAv2 dataset to execute SCA, combined word embedding with LSTM. Normalization, Butterworth filtering and Fourier transform are respectively used to cope with the DPAv2 dataset, and the Hamming Weight of the S-box output in the last round is attacked as the classification. We use MLP (Figures 12 and 13), CNN (Figures 14 and 15) and LSTM (Figures 16 and 17) model to carry out side-channel attack dividedly and compare the guess entropy.

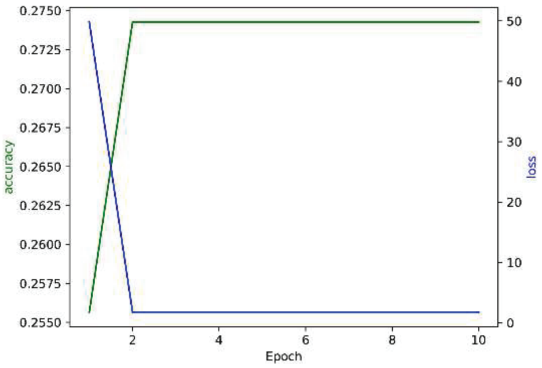

Figure 12 Acc and Loss of MLP (Epoch: 10).

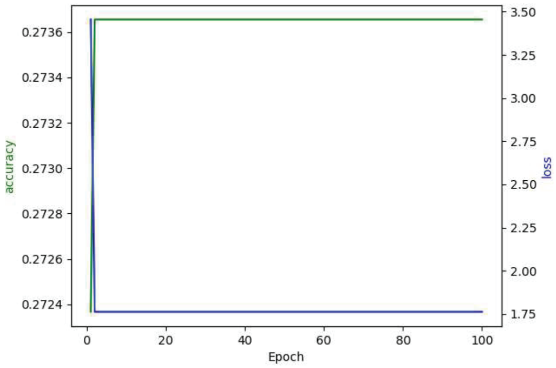

Figure 13 Acc and Loss of MLP (Epoch: 100).

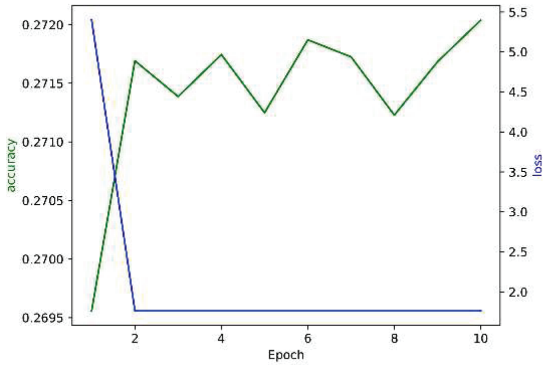

Figure 14 Acc and Loss of CNN (Epoch: 10).

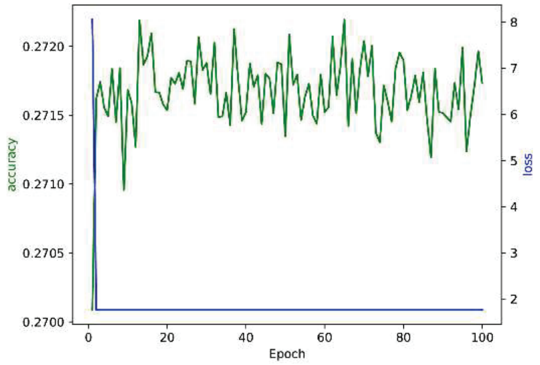

Figure 15 Acc and Loss of CNN (Epoch: 100).

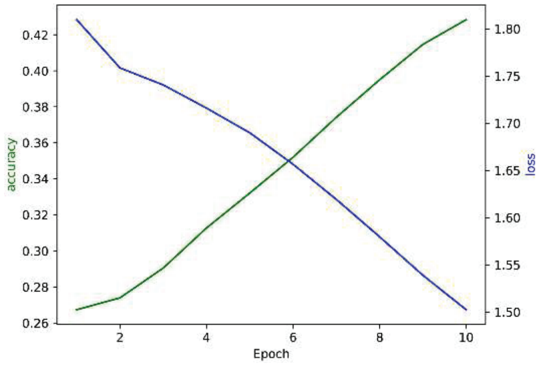

Figure 16 Acc and Loss of LSTM (Epoch: 10).

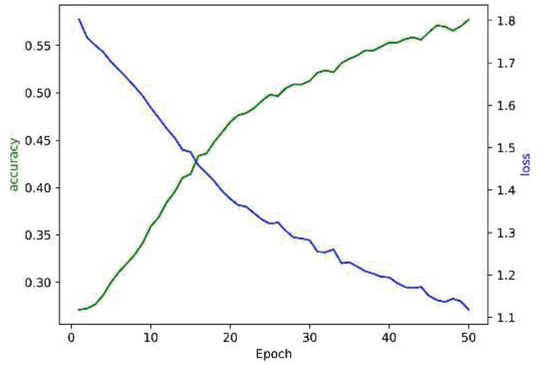

Figure 17 Acc and Loss of LSTM (Epoch: 50).

Table 1 Guessing entropy of using MLP attack (from round 1 to round 10)

Preprocessing	1rd	2rd	3rd	4rd	5rd	6rd	7rd	8rd	9rd	10rd
MLP	129	94.5	82.7	77.7	72.3	62.9	65.1	54.9	44.5	41.2
RE&MLP	96.6	82.9	81.5	79.6	67.4	63.5	52.3	54.9	44.5	38.4
Butter&MLP	83.2	85.4	79.2	68.6	63.8	58.4	57.5	43	39.7	33.1
FFT&MLP	82.9	63.5	74.6	69.2	66	63.7	59.3	54.9	44.5	40.7

Table 2 Guessing entropy of using CNN attack (from round 1 to round 10)

Preprocessing	1rd	2rd	3rd	4rd	5rd	6rd	7rd	8rd	9rd	10rd
CNN	89	73.4	62	69.1	58.2	53.8	47.3	36.4	34.7	29.3
RE&CNN	85.6	72.9	71	62.4	58.3	53.9	42.5	40.9	39.5	28.2
Butter&CNN	83.2	75.4	69.9	68.6	63.8	58.4	57.5	47.6	38.2	33.7
FFT&CNN	89.2	63.5	71.4	59.7	68.3	53.9	52.5	44.9	34.5	30.1

Table 3 Guessing entropy of using LSTM attack (from round 1 to round 10)

Preprocessing	1rd	2rd	3rd	4rd	5rd	6rd	7rd	8rd	9rd	10rd
LSTM	109	73.5	69.5	59.7	58	53.9	47.1	46.9	34.3	27.4
DP0.1&LSTM	93.2	85.4	72.1	68.6	53.8	48.4	37.5	37.6	28.3	13.7
DP0.2&LSTM	89	63.5	61.5	59.7	58.3	53.9	42.5	34.9	25.5	10.1
WE&LSTM	86.7	82.6	71.9	69.3	57.2	33.9	32	26.9	14.5	12

5 Discussion

Normalized pre-processing method combined with MLP model attack is effective. With the increase of the number of energy trace curves, the effects of training and attack are improved when we put them into the model together.

Compared with the normalized processing, Butterworth filter pre-processing is not a perfect method to improve the effectiveness of attacks. It has an obvious effect in improving the accuracy of training, the effect is not good yet when the trained model is used for attacks.

The use of Fourier transform sacrifices the time domain information, but we found that the training effect of using MLP and CNN model is better than that of using MLP and CNN only by relying on the information of frequency domain.

Compared with the model combined word embedding with LSTM, the above three pre-processing methods are superior to the above models in both training accuracy, attack effect, and save time.

Dropout must be used in this experiment. The learning rate is set artificially at $10e-3$ is not better than the experimental result of combining the learning rate at $10e-2$ with Dropout. Therefore, the use of Dropout should be considered in addition to the case of overfitting.

In addition, in the experiment with Tensorflow2.x version, we found that the use of CUDA was directly related to the content of the code. If the activation function were changed (in the side channel attack, we mostly used the SELU activation function), CUDA could not be used, and the experimental speed was significantly reduced.

Limitation: Although the training accuracy and attack effects are better than the previous models, there is still a problem that the total training time exceeds the training time of the previous models due to the increase of parameters.

6 Conclusion

In this article, we put forward a kind of based on embedded and LSTM combination of side channel analysis method, not only can replace the previous POIs selection and dimension reduction pre-processing operations. In addition, this method helps us improve the training accuracy and attack effect, the training dataset and validation dataset have effective attack in terms of different random key datasets.

At the same time, we also analyse the shortcoming of this model, its training time and attack time are much higher than other models. The mapping expression of power consumption curve can be regarded as a process of encoding and decoding. Most of the time, experimental data will be lost, or the noise distribution will be affected under such operation. Therefore, we tried several dimensions of the word vector, as did the activation function and the middle layer. In the future work, we hope to find ways to reduce the running time and optimize the algorithm. At last, we are also considering whether self-attention model can be applied to the side-channel analysis to replace the previous idea of noise modelling.

Acknowledgments

This work is supported by Side Channel Analysis based on Transfer Learning, NO. TICPSH202103015-ZC. Side Channel Analysis of Geo-big Data based on Neural Network, NO. JELRGBDT202005. Research on Nuclear Geoscience Data Sharing based on Blockchain Technology, NO. GJJ200721.

References

[1] Giraud C. An RSA Implementation Resistant to Fault Attacks and to Simple Power Analysis[J]. IEEE Transactions on Computers, 2006, 55(9):1116–1120.

[2] Kocher P C. Timing attacks on implementations of diffehellman, RSA, DSS, and another systems[J]. Adances in Cryptography-CRYPTO'96, 1996.

[3] Kocher P C, Jaffe J, Jun B. Differential Power Analysis[M]. Springer US, 2007.

[4] Chari S, Rao J R, Rohatgi P. Template Attacks[J]. International Workshop on Cryptographic Hardware & Embedded Systems, 2002.

[5] Schindler W, Lemke K, Paar C. A Stochastic Model for Differential Side Channel Cryptanalysis[C]// Cryptographic Hardware and Embedded Systems – CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 – September 1, 2005, Proceedings. Springer-Verlag, 2005.

[6] T Bartkewitz, K Lemkerust. Efficient Template Attacks Based on Probabilistic Multi-class Support Vector Machines[M]. Springer Berlin Heidelberg, 2013.

[7] Olivier, Markowitch, Liran, et al. Power analysis attack: an approach based on machine learning[J]. International journal of applied cryptography: IJACT, 2014, 3(2):97–115.

[8] Ramezanpour K, Ampadu P, Diehl W. SCAUL: Power Side-Channel Analysis with Unsupervised Learning[J]. IEEE Transactions on Computers, 2020, PP(99):1–1.

[9] Jin M, Zheng M, Hu H, et al. An Enhanced Convolutional Neural Network in Side-Channel Attacks and Its Visualization[J]. 2020.

[10] Zhang H, Zhou Y. Template Attack vs. Stochastic Model: An Empirical Study on the Performances of Profiling Attacks in Real Scenarios[J]. Microprocessors and Microsystems, 2019, 66(APR.):43–54.

[11] Benadjila R, Prouff E, Strullu R, et al. Deep learning for side-channel analysis and introduction to ASCAD database[J]. Journal of Cryptographic Engineering, 2019, 10(Feb).

[12] Golder A, Das D, Danial J, et al. Practical Approaches Towards Deep-Learning Based Cross-Device Power Side Channel Attack[J]. 2019.

[13] Cagli E, Dumas C, Prouff E. Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures[C]// International Conference on Cryptographic Hardware and Embedded Systems. Springer, Cham, 2017.

[14] Moini S, Tian S, Szefer J, et al. Remote Power Side-Channel Attacks on CNN Accelerators in FPGAs[J]. 2020.

[15] Ramezanpour K, Ampadu P, Diehl W. SCAUL: Power Side-Channel Analysis with Unsupervised Learning[J]. IEEE Transactions on Computers, 2020, PP(99):1–1.

[16] Won Y S, Han D G, Jap D, et al. Non-Profiled Side-Channel Attack based on Deep Learning using Picture Trace[J]. IEEE Access, 2021, PP(99):1–1.

[17] J Wei, Y Zhang, Zhou Z, et al. Leaky DNN: Stealing Deep-learning Model Secret with GPU Context-switching Side-channel[C]// IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, 2020.

[18] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735–1780.

[19] Paguada S, Batina L, Armendariz I. Toward practical autoencoder-based side-channel analysis evaluations[J]. Computer Networks, 2021(4):108230.

[20] Wold S, Esbensen K, Geladi P. Principal component analysis[J]. Chemometrics & Intelligent Laboratory Systems, 1987, 2(1–3):37–52.

[21] Kwon D, Kim H, Hong S. Non-Profiled Deep Learning-Based Side-Channel Preprocessing With Autoencoders[J]. IEEE Access, 2021, PP(99):1–1.

[22] Bojanowski P, Grave E, Joulin A, et al. Enriching Word Vectors with Subword Information[J]. Transactions of the Association for Computational Linguistics, 2017, 5:135–146.

[23] Bengio Y, Réjean Ducharme, Vincent P, et al. A Neural Probabilistic Language Model.[J]. Journal of Machine Learning Research, 2003.

[24] CHURCH, Ward K. Word2Vec[J]. Natural Language Engineering, 2017, 23(01):155–162.

[25] Graves A, Mohamed A R, Hinton G. Speech recognition with deep recurrent neural networks[C]// Acoustics, Speech, and Signal Processing, 1988. ICASSP-88. 1988 International Conference on. IEEE, 2013.

Biographies

Zixin Liu, male, born in 1985, a member of the Communist Party of China, lecturer, graduate degree. 2009.6 graduated from School of Software, East China Normal University, majoring in software engineering. Since September 2009, he has been a teacher at software academy, East China University of Technology. He has been engaged in the research of hardware Side-channel attack, artificial intelligence and digital geological crossover application.

Zhibo Wang, male, born in February 1984, PhD, supervisor of postgraduate, Dr 2017.6 graduated from Wuhan university of software engineering, is mainly engaged in large data analysis, block chain technology in areas such as research, including such as SCI, EI retrieval, papers published more than 20 articles, apply for a patent for invention 6, 13 utility model patents (including authorized 10), 3 software Copyrights (3 have been authorized). Presided over and participated in many provincial and ministerial research projects.

Mingxing Ling, male, Ph.D., distinguished Professor and researcher of East China Institute of Technology. He has been selected as the innovation Leader of “Double Thousand Plan” of Jiangxi Province, Young JingGang Scholar, Outstanding Youth of Guangdong Natural Science Foundation, Top Young Talents of Science and Technology Innovation of Guangdong Special Support Program, member of Youth Innovation Promotion Association of Chinese Academy of Sciences and other talent programs. His research interests include supernormal enrichment and mineralization mechanism of key metals, plate subduction and magmatic activity and mineralization in eastern China, metal isotope technology and geological application. He has presided over more than 10 projects of national Natural Science Foundation of China and National Key Research and development Program.