The Use of Deep Learning in Capturing Cryptographic Keys through Side Channel Analysis

Yusron Taufiq Anfasa - 18217002
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): yusrontfiq@gmail.com

***Abstract*—Side Channel Analysis is a method to catch informations using the weaknesses of the hardware implementation instead of software algorithms or source code. One of the information gained from this method is cryptographic keys. While generating the key, the process running on the hardware could release some heat, currents, and electromagnetic emission. The emissions could be processed and recovered to gain all the informations, including cryptographic keys from both symmetric and asymmetric encryption. One of the way to recover informations gained is using deep learning It could be considered as serious and dangerous threat to an encryption algorithm. This paper would propose the use of deep learning in recovering cryptographic keys from side channel attack. The deep learning makes raw data processing possible without using any human-made engineering techniques. The machine could learn the pattern of power consumption and electromagnetic signal.**

Keywords—side channel attack, recovery, cryptographic keys, deep learning

I. INTRODUCTION

In the digital world, Side Channel Analysis (abbreviated as SCA in the rest of the document) is not something new. The method that exploiting the emission gained from implementation flaws has been around since 90s. It was mentioned as "*side-channel cryptanalysis"* by the Researcher from Counterpane Systems and the University of California. [1]

Back then, the SCA methods relied only on the conventional methods such as timing attack, various power consumption pattern, and electromagnetic signal analysis. Timing attack was the simplest among other methods where it computes the time taken for certain operation because the different input received, the different time a machine consumes to gain the result. Time-based observation could be done within all computation except modullo computation which could take a plenty of time. For example, an RSA key generation would iterate the multiplication process if $e_i = 1$ as shown in Figure 1 [2]. Time-based attack could predict how many multiplications were done to get a certain length of key. However, the method is considered easier that an operation to make zero output will always take less time[1].

ALGORITHM 1
LEFT-TO-RIGHT BINARY METHOD

Input:	X, N, $E = (e_{k-1}, ..., e_1, e_0)_2$
Output:	$X^E \bmod N$

```
1 : R := 1;
2 : for i = k − 1 downto 0 do
3 :     R := R · R mod N;          — squaring
4 :     if e_i = 1 then
5 :         R := R · X mod N;      — multiplication
6 :     end if
7 : end for
8 : return R
```

Figure 1. Modular Exponentiation (Homma, N. et. al, 2010)

Meanwhile, the power consumption and electromagnetic signal analysis are considered to be tricky because they require the attacker to have some knowledge on both aspects, e.g signal processing to examine the statistics obtained from power consumption graph.

As the cryptography algorithm become more advanced, these techniques could require much time and effort to do. Seeing how fast the growing of the algorithms is, attacker started to implement machine learning theories to capture raw data from SCA. A supervised learning, part of the machine learning could help the SCA techniques to adapt themselves into different algorithm implementation.

Deep Learning is a subset of machine learning that uses an artificial neural network to learn and improve what it can do. Deep Learning is used mainly to classify problems into small clusters. In more advanced aspect, Deep Learning could be used to detect noises in numerous object such as plaintext, voice, videos, and images. A neural network uses layers to determine a classification of objects. These layers is consisted of nodes that connect each node to certain node in the hidden layer where filtering is done. A hidden layer(s) could contains many layers such as normalisation, pooling, activation, etc. depends on the need.

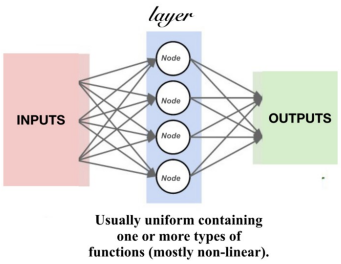

Figure 2. Layers in Neural Network (OpenGenius IQ)

In SCA, the raw data captured through hardware emissions could contain noises and unwanted informations. To recover the important information such as cryptographic keys, the attackers have to implement noise reduction. It is possible through Differential Power Analysis or DPA with some configurations [3]. But sometimes, the more complex raw data received, the more advanced a configuration should be implemented.

This paper is aimed to propose an idea about using deep learning techniques to recover cryptographic keys from side channel attacks. The contributions of this paper are written as follow:

1. The Convolutional Neural Network method is used to do signal analysis. Convolutional Neural Network has the ability to process multidimensional data such as image and speech processing. This method would be suitable for detecting misalignment given in emission signal visualisation.
2. The analysed SCA is profiled attacks. A profiled attack doesn't need to be a target device, instead the adversary will configure its attacks parameters using a clone device under the assumption that the sample device behaves exactly the same with the target [5].
3. The cryptographic algorithm analysed and proposed is AES-128 or simply AES. The algorithm is used because it does a repeating processing of transformation during encryption. Each process would emit different size of signal emission, depending on how big the data is.

II. Theories and Literatures

A. Side Channel Analysis

In capturing the raw data from hardware emissions, SCA is categorised into two [4]:

a. Unprofiled Attacks

Unprofiled Attacks purely use the leakage of hardware emissions as the main source of raw data. It processes the raw data that were usually a waveform or power consumption pattern graph using power analysis methods, such as simple power analysis, differential power analysis, and high-order differential power analysis. This attack requires the attacker to have enough knowledge on how to process signal received from power analysis and the actual target where the attack will be implemented [3]. The attackers also need to have insights or get the actual device to do attacks.

b. Profiled Attacks

In profiled attacks, the attackers has a clone device that represents the actual target. They only need to have knowledge about the clone device so they are able to do everything on the clone device, including obtaining secret keys and profiling traces [6]. Profiled attacks is consisted of 2 phases: The profiling phase and attack phase. A profiling phase is the period when the adversary calculates a probability distribution function to construct the model as shown in Figure 3. The model represents a dependency relation between secret keys and side-channel information such as electromagnetic waves or power consumption pattern [3]. After constructing a model, the attackers use it to attacks the actual target. This is the attacking phase of profiled attacks.

$$g_k : (\bar{l}, p) \rightarrow \mathrm{P}[\bar{L}=\bar{l}|(P,K)=(p,k)] \quad (1)$$

One of the standard method to do profiled attacks is *Template Attacks* or TA. It was introduced by Chari *et al.* as a method to do key extraction from side channel information. Instead of modelling the power consumption pattern from actual device leakage, TA uses clone device to accurately model the power consumption [7]. Because of its characteristics, TA is mentioned to be one of the strongest SCA attacks theoritically.

The advantage of TA that could be its biggest gain is the ability to recover cryptographic keys without many power consumption traces. This makes the template attacks should be performed in nearly identical clone devices to get the accurate power consumption traces. As a subset of profiled attacks, TA also has 2 phases, the profiling and the attacks. The model made in profiling phases is reusable, that means the attackers are able to perform using the model in any similar devices[7].

B. AES

AES or Advanced Standard Encryption is part of symmetric key encryption algorithm. The algorithm is based on block cipher. AES supports variable key length, from 128 bits to 256 bits with 32 bits steps increment [8]. AES was made through competitions which the winner's algorithm, Rijndael, is chosen.

Rijndael Algorithm supports variable key length and block size that could be chosen independently. The most common AESs key length is 128, 192, and 256 hence AESs family is widely known as AES-128, AES-192, and AES-256. With the certain key length, there are 2^{128} keys possible, which needs $5.4*10^{18}$ years to solve by the fastest computer under the assumption that it tries as many as 1 million keys every milliseconds [8].

An AES algorithms use block of bytes to encrypt data. It would run repeatedly for 10,12, or 14 times depending on what key length the user is using. For example, an AES-128 will need 10 times round. The longer the key, the more it repeats [9].

Before doing a round, the data would be devided into blocks. An AES-128 would have a block consisted by 4x4 columns. In each round except the final one, an AES would perform 4 main transformations:

a. SubBytes

In the beginning, the input would be split into bytes. SubBytes would substitute every bytes from array state to Rijndael's S-Box

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
a	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
b	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
c	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
d	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
e	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
f	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure 4. Rijndael's S-Box (Arrañaga, J. et al., 2017)

b. ShiftRows

ShiftRows would shift the last 3 rows of array state by 1 bit cyclically to the left. The initial row would be shifted and appear in previously the last row space.

c. MixColumns

MixColumns would multiply the state array in matrix state with certain matrix to produce a new column. All the column would be seen as 4 degrees polynomial equation in GF(2^8). It will produce a completely new matrix.

d. AddroundKey

AddRoundKey performs an XOR operations between the key and MixColumns' produced state. It would be stored as a state to use for another round. AddroundKey is also performed at the beginning before AES start the rounds by doing XOR operation between plaintext and cipher key. This step is so called initial Round [8].

The final round of AES would perform only SubBytes, ShiftRows, and AddroundKey to get the state. The state from this step would be counted as the final output

C. Convolutional Neural Network

Convolutional Neural Network is a deep learning approach that is used mainly to solve complex problem. The fully connected layers (as shown in Figure 2) it gets make the data generalisation more efficient. Compared to the other neural network approach, CNN has main advantages in terms of processing complex and big datasets:

1. CNN has the concept of weight sharing which could reduce the number of parameters used in model training. This way would make generalization better [10].
2. The classifications stage and feature extraction could be run together as both of them using learning rates in their process.

Due to its performance versatility towards big and complex data, CNN is widely used in many domains such as face detection, image classifications, object detection, and any complex recognition process.

A general model of CNN consists of various layer which 3 of them are considered main layers: Convolution, Pooling, and Fully Connected (Activation) layer as shown in Figure 5.

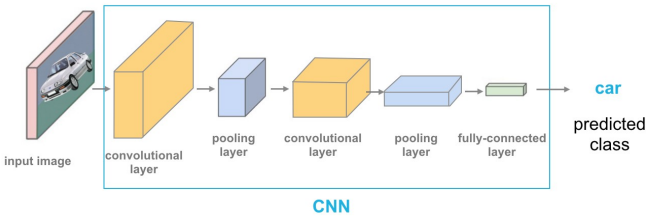

Figure 3. General Model of CNN (Cezanne, C. 2019)

A Convolution layer is the main building block of CNN. It extracts numerous features from the input. To generate the feature map, there will be a filter that slides over the matrix of input vector. A filter is 1-dimensional and usually a stride is chosen according to their length. This is also known as weight vector. The output is computed after a convolution operation is applied using formula [10]:

$$a_{ij} = \sigma((W*X)_{ij} + b) \quad (2)$$

where a_{ij} as output, X as input provided to the layer, b as the bias, also there is a convolution operation between W and X. The sigma (σ) symbol is representing nonlinearity.

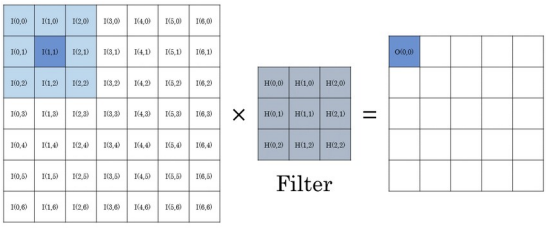

Figure 6. How a filter is used in convolution layer (Baskin, C. et al. 2017)

Pooling layer on the other hand is used to decrease the number of parameters so the computational power required would be less. Pooling layer would reduce the matrix size (or neuron size) provided from convolution layer and introduce translation invariance[10]. Just like convolution layers, pooling layer also makes some filters or kernels to slide over the input. There are two types of pooling layers: Average and Max-Pooling layers. The average pooling would take the average of

all values in the pooling matrix region meanwhile the max-pooling would see the maximum numbers (Figure 7).

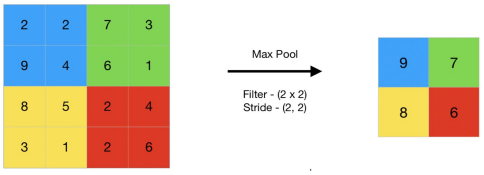

Figure 7. Max Pooling (Geeksforgeeks, 2019)

The fully connected layer acts as the activation that does the real classification process. The user usually uses gradient descent to optimise the function used in activation. It reduces the function cost by choose data from dataset randomly through *stochastic gradient descent* or doing a batch training to the entire dataset.

Inside the fully connected layer, there is an activation function that transforms all the input to correct output. It decide whether a neuron should be fired or calculated. It introduces the nonlinearity into the output. One of the activation function used in CNN is Rectified Linear Unit or ReLU [10]. ReLu wouldn't allow vanishing gradients that is usually happens in neural network so it could be used in the weight calculation method using gradient descent.

III. Research about Deep Learning in Side Channel Attack

The Use of Deep Learning in profiled based SCA to get the cryptographic keys has been conducted in some researches nowadays. For example, the research conducted by Hanley and Neil John in 2014 which proposed a general neural network use to break the AES algorithm implementation[7].

In 2016, H, Maghrebi et al[11] used MLP and CNN deep learning method to perform a key cracking on the DPA contest. It empowered the classic template attack. They compared the efficiency of their proposed attack and the most commonly used template and previous machine learning based attack. The result shows that the deep learning method they proposed outperformed classical AE learning classification.

In the same year, Martinasek et al[12] also used an MLP method to break AES-128 encryption algorithm and compared their method to classical method such as template and random attacks. The experiments and research they conducted show that MLP could improve the efficacy of profling based side channel attack compared to the previous techniques. However, their experiments turned into failure with certain datasets. Martinasek assumes that the distortion happened in MLP in one of the step affects the datasets that gave the implementation a failure.

In 2017, Cagli, E. et al[13] used CNN together with data augmentation to propose an end-to-end profiling attack strategy. The experiment they conducted proved that such strategy would improve the trace of misalignment and CNN - based attacks succesfully recover the misaligned data. Their experiment showed how CNN managed to work with high-dimensional data. Although they met an overfitting phenomenon which was quite classical in general machine learning implementation, they also implemented data enhancement techniques that adapted to misaligned traces.

In 2018, Samiotis, I.P [14] optimised the CNN architecture to do SCA and compare his to variety of CNN architectures in term of performances. He analysed the CNN structure used in DPA contests based on Maghrebi and Cagli's paper. He found that in the cases where there are relatively low noises, CNN outperformed other machine learning approaches. However, he found that it didn't do so in DPA contest v2 datasets as simple machine learning approach outperformed CNN. He insisted a future deeper research on the use of CNN model in SCA.

Based on the research, it is possible for SCA to use neural network in order to improve its performance. There are some architecture designs of CNN such as ResNet, AlexNet, VGG-Net, and so on [5]. Especially the deconstruction design of neural network after ResNet into blocks, CNN has proven their capabilities to give a good classification and generalization compared to another DL and classical side channel attack methods.

The Inception network, on the other hand has improved CNN in terms of using convolutional layers. The conventional CNN is getting stuck in using more and more layers as it would make the performance less efficient. Inception network makes the focus of CNN spread the concentration of improvement. It doesn't use a fully connected network as much as previous CNN architecture, instead it uses inception modules that could make a good local network topology [5]. Inception modules perform convolution and pooling operations towards the input in parallel and splicing the results into a deep feature map [5]. The different operations would result a different output that could be combined and give a better image characterization result.

The disadvantage of inception module is the complexity. In order to improve the performance parameter such as accuracy and speed, it needs a lot of tricks and engineering works. The evolution of inception module, however, is very promising to optimise the classification conducted in side channel attacks.

IV. CNN As a Deep Learning Approach in Side Channel Attack

The first thing we have to do is determining what kind of input from the leakage we will use to do a side channel attack based on CNN. The most common leakage emission from a computer hardware is **timing, current, heat,** and **electromagnetic waves.** Each leakage has different way to compute, such as DPA to compute power consumption or classic timing attacks.

The example of proposals discussed here is power-based attack. Power consumption during different computation may vary. The various power consumption could lead into a pattern which we can analyse. All the pattern would be processed as a dataset for CNN to process. This phase is called **data collection phase** [15]**.** This phase is important as the dataset of traces we collected would be made into a model during profiling based-attack main phase; training and attacking. The example of data that could be collected and made into a dataset

is power traces of the device when it does the cryptographic computation.

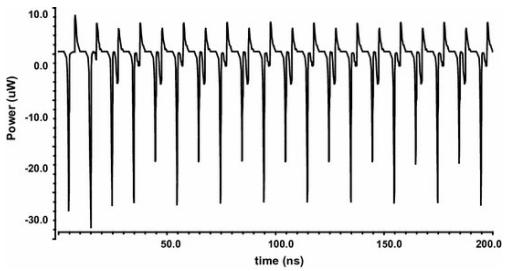

Figure 8. Example of Power Traces collected through data collection phase (Prathiba, A. et al 2018)

If we see the graph slightly, it made some identical pattern within certain time measure. In the device implementing AES without any protection, this could be viewed as the round AES does in one process. Using oscilloscope, a numerical precision is used depending on how accurate an oscilloscope measured power consumption [15].

A. Data Collection

During the data collection, the raw inputs (traces) would be made into two datasets: **training** and **evaluating.** The training data would be used for the machine to learn the power consumption and build the algorithm to attack. The power consumption is learnt to know its relation to key-generating process. In the training, the attacker train the algorithm they made with any keys or inputs with assumption that the device they use is under the control. In this step as much as possible traces are collected, depends on how protected the AES operation is. The traces collection could take several hours to day, depending on how many traces collected.

Meanwhile, the evaluation would be represent the real attack target so the keys generated and used in this case would be random to represent how the user choose their own key.

After doing the trace collection, the next step is doing the clock capturing. Clock capturing could be synchronous or asynchronous, depend on the resource we use and what kind of traces are collected.

Synchronous capture would capture clock output of target CPU to extract the power consumption in each target clock frequency. This means, there would be no information missing as it captures the internal CPU clock output. However, it would be nearly impossible for a CPU manufacturer to expose an access to all the attackers. Usually this techniques is used by chip designers to find the leakage and not an open access to public to see.

In the other hand, asynchronous capture is what is most likely done by attackers. CPU clock is not used at all to synchronise the oscilloscope [15]. It makes sense that attackers or public would have no access to the internal CPU clock where secure information is located and processed. This method also needs to have more traces to analyse as what they got would have so many misalignment. But the problem is, traces are usually quite a lot. It could be more than a million and it increases as the implementation getting more secure. The more traces are collected and made into datasets, the more challenging DL will. A lot of traces could led to a very big data and it could be a heavy load for the CPU or GPU memory. Without the right hardware and neural network model, it is nearly impossible to process asynchronous capture.

The oversampling an asynchronous capture made is usually used in electromagnetic-based SCA. That is what electromagnetic-based SCA differ from the power-based.

B. Training Phase

In the training phase, the attackers build a leakage model based on the traces collected. In the classic SCA techniques such as template attack or SCA that was assisted by machine learning, they use training data for further analysis. For example the template attack would use multivariate statistical analysis to create the model. It was also known as Hamming Weight Power Model [15][16].

The deep learning approach would skipped the techniques or using any human-engineered structure to work, instead it uses the raw data to directly build a model. The first thing to do is looking for the attack points [15]. Attack points refer to a process in AES round where the key is used. While using a key to process data, there would be memory or register value change in the hardware. A memory change could possibly emit some signal that could be a leakage. This is where the traces are collected. In the training phase, this is the most possible attack point for DL to process.

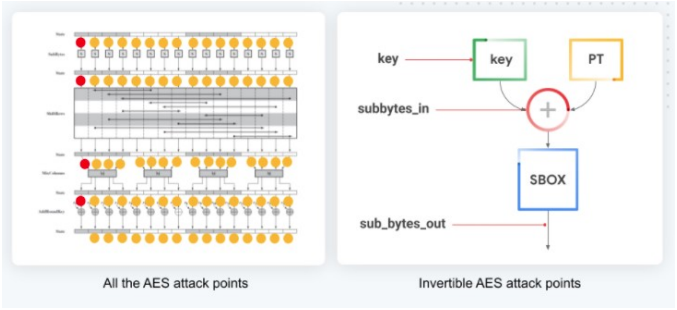

Figure 9. AES attack points (Bursztein, E. 2021)

In the Figure 9, there are several possible attack points in AES round process. Most of the possible attack points are not invertible [15]. The directly invertible process was the SubBytes, where the key is also directly used. All the invertible process were marked as red dot, where it has 3 main points:

- Key

  The value of key is stored in memory. As the memory is secured, this attack points will most likely to fail.

- SubBytes_in

  The key was stored with plaintext in this point. It is easier to attack compared to 'key' point [15].

- SubBytes_out

The byte value here has been substituted using S-Box. Also easier to attack compared to the 'key' point [15].

After deciding the attack points, the next step is building the model using CNN. Model is built using training and evaluating data collected in data collection. The problem using deep learning is usually the way to find architecture parameters. To overcome this issue, it is needed to choose a hyperparameter using hyperparameter optimisation or hypertuning.

Once the hyperparameter and the right CNN architecture is decided, the next procedure would be a standard step to train the model with some notes [15]:

- Input should be scaled to ensure the models won't converge
- The activation function uses softmax to converge the network more precisely. Softmax activation (Equation 3) would also gain a probability distribution output of a network so it could carry the attack.

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K. \tag{3}$$

- Because the output of softmax is probability distribution, the loss function used should be categorical cross-entropy.

C. Attack Phase

After training the models in the previous step, now the model would be used in actual key recovery. In the attack dataset, the key generated would be different as the attacker used in training phase. The random key would be generated to represent the real user behaviour.

To get the right prediction, we simply accumulate the model prediction to decide the most probable value. It means, the more evidence and model training we did and accumulate, the better rate we will get as long as the model converged. But it would not be a problem if we use softmax activation function as it ensures the model to converge so the prediction would be more accurate.

TABLE I. OUTPUT AND PROBABILITIES

Output	*Probabilities*
1.4	0.03
7.8	0.92
2.7	0.02
1.2	0.03

The example probabilities output is Table 1, if we sum those value up, would be exactly 1. This is how softmax would work to find the most probable value to guess what is the key using CNN spesifically.

To recover the full keys, it most likely won't happen in a single trace. Usually a few traces would be needed to get a near 100% recovered keys.

V. CONCLUSIONS

Using deep learning to perform side channel attack has proven its ability to process high-dimensional data without any human-engineered feature or structures. It also acts as bridge for any attackers without prior knowledge to signal processing or any classical profiling attack techniques. However, there are some notes to take:

1. Sometimes the deep learning, specifically CNN would work only with certain datasets. It means there are still much more research to conduct so the method would be near accurate as the classical one.
2. Deep Learning would eliminate the time used to analyse raw data as it works directly on them.
3. There are still few deep learning methods used in SCA, such as CNN, MLP, and RNN.
4. The research in this field is still developing. As the latest resource is made in May 2021, this shows that the use of deep learning in SCA hasn't reach their final yet.

ACKNOWLEDGMENT *(Heading 5)*

I would like to appreciate Mr. Rinaldi Munir as the Coding and Cryptography lecturer for the academic supports throughout the semester and providing some resources in the paper.

REFERENCES

[1] Kelsey, J., Schneier, B., Wagner, D., & Hall, C. (1998). Side Channel Cryptanalysis of Product Ciphers. In Computer Security — ESORICS 98 (pp. 97–110). Springer Berlin Heidelberg. https://doi.org/10.1007/bfb0055858

[2] Homma, N., Miyamoto, A., Aoki, T., Satoh, A., & Samir, A. (2010). Comparative Power Analysis of Modular Exponentiation Algorithms. *IEEE Transactions on Computers*, 59(6), 795–807. https://doi.org/10.1109/TC.2009.176J.

[3] Kubota, T., Yoshida, K., Shiozaki, M., & Fujino, T. (2020). Deep Learning Side-Channel Attack Against Hardware Implementations of AES. *Microprocessors and Microsystems*, 103383. https://doi.org/10.1016/j.micpro.2020.103383

[4] Standaert, F.-X., Koeune, F., & Schindler, W. (2009). How to Compare Profiled Side-Channel Attacks? In M. Abdalla, D. Pointcheval, P.-A. Fouque, & D. Vergnaud (Eds.), *Applied Cryptography and Network Security* (pp. 485–498). Springer. https://doi.org/10.1007/978-3-642-01957-9_30

[5] Song, S., Chen, K., & Zhang, Y. (2019). Overview of Side Channel Cipher Analysis Based on Deep Learning. *Journal of Physics: Conference Series*, 1213, 022013. https://doi.org/10.1088/1742-6596/1213/2/022013

[6] Picek, S. (2019). Challenges in Deep Learning-Based Profiled Side-Channel Analysis. In *Security, Privacy, and Applied Cryptography Engineering* (pp. 9–12). Springer International Publishing. https://doi.org/10.1007/978-3-030-35869-3_3

[7] Hanley, N. J. (2014). Profiling side-channel attacks on cryptographic algorithms [Doctoral thesis, University College Cork]. https://cora.ucc.ie/handle/10468/1921

[8] M, Rinaldi (2021). Advanced Encryption Standard (AES). Lecture slides of II4031 – Coding and Cryptography.

[9] Maharaj, A. (2020) A Review on Advanced Encryption Standards (AES). Fiji National University.

[10] Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. Procedia Computer Science, 132, 679–688. https://doi.org/10.1016/j.procs.2018.05.069

[11] Maghrebi, H., Portigliatti, T., & Prouff, E. (2016). Breaking Cryptographic Implementations Using Deep Learning Techniques. In Security, Privacy, and Applied Cryptography Engineering (pp. 3–26). Springer International Publishing. https://doi.org/10.1007/978-3-319-49445-6_1

[12] Martinasek, Z., Dzurenda, P., & Malina, L. (2016, June). Profiling power analysis attack based on MLP in DPA contest V4.2. 2016 39th International Conference on Telecommunications and Signal Processing (TSP). 2016 39th International Conference on Telecommunications and Signal Processing (TSP). https://doi.org/10.1109/tsp.2016.7760865

[13] Cagli, E., Dumas, C., & Prouff, E. (2017). Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures. In Lecture Notes in Computer Science (pp. 45–68). Springer International Publishing. https://doi.org/10.1007/978-3-319-66787-4_3

[14] Samiotis, I. P. (2018). Side-channel attacks using convolutional neural networks: A study on the performance of convolutional neural networks on side-channel data. https://repository.tudelft.nl/islandora/object/uuid%3A2e203eee-4c38-4c86-a92a-db94d0ffc34c

[15] Bursztein, E. (2021). Hacker's guide to deep-learning side-channel attacks: The theory. Elie Bursztein's Site. Retrieved May 25, 2021, from https://elie.net/blog/security/hacker-guide-to-deep-learning-side-channel-attacks-the-theory/

[16] Lo, O., Buchanan, W. J., & Carson, D. (2016). Power analysis attacks on the AES-128 S-box using differential power analysis (DPA) and correlation power analysis (CPA). Journal of Cyber Security Technology, 1(2), 88–107. https://doi.org/10.1080/23742917.2016.1231523

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 April 2021

Yusron Taufiq
18217002