Innovative Method of the Power Analysis

Zdenek MARTINASEK, Vaclav ZEMAN

Dept. of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology
Technicka 12, 616 00 Brno, Czech Republic

martinasek@feec.vutbr.cz, zeman@feec.vutbr.cz

Abstract. *This paper describes an innovative method of the power analysis which presents the typical example of successful attacks against trusted cryptographic devices such as RFID (Radio-Frequency IDentification) and contact smart cards. The proposed method analyzes power consumption of the AES (Advanced Encryption Standard) algorithm with a neural network, which successively classifies the first byte of the secret key. This way of the power analysis is an entirely new approach and it is designed to combine the advantages of simple and differential power analysis. In the extreme case, this feature allows to determine the whole secret key of a cryptographic module only from one measured power trace. This attribute makes the proposed method very attractive for potential attackers. Besides theoretical design of the method, we also provide the first implementation results. We assume that the method will be certainly optimized to obtain more accurate classification results in the future.*

Keywords

Power analysis, Smart cards, Neural network, SPA, DPA.

1. Current State Analysis

Power analysis presented in this paper is a completely new approach in power analysis attack. The method is designed to combine the advantages of simple and differential power analyses. The proposed method uses a neural network (NN) to determine the secret key value of the cryptographic module from the measured power consumption. Application of neural networks to analyze information leakage through the power side channel has not been completely published yet. The publications dealing with the use of NN in the side channels cryptanalysis are mostly focused on acoustic side channels where the NN are used for classification of the captured records of pressed buttons on keyboard [1], [2]. In the field of the power analysis, we found few publications dealing with classification of individual instruction in power consumption. These works are more or less oriented on possibilities of reverse engineering [3], [4]. Classification focused on identifying the specific values of a secret key has not been published yet. General use of neural networks in cryptography is focused on the application of the neural networks, for use in encryption and cryptanalysis. Neural networks are well known for their ability to selectively explore the solution space of a given problem, therefore this future is naturally used in cryptanalysis. Neural networks also offer a new approach for encryption and decryption because each function could be represented by them. Neural networks are a powerful computational tested tool that can be used to find the inverse of the encryption algorithm. As already described above, the neural networks are used mostly in public-key cryptography [5], key distribution [6], hash function [7], random number generators [8] and in the key exchange protocol [9] (equivalent to the Diffie-Hellman protocol).

Power analysis (PA) measures the power consumption of the cryptographic device depending on its activity and was characterized by Kocher in [10]. A detailed description of the side channel sources and the division of the power consumption is exhaustively summarized in the book [11]. This book describes also the basics of the measurement methods and experimental testbed. A detailed comparison of the measurement methods and the influence of the parameters affecting the results measured power consumption is discussed in [12]. After successful measurement of the power consumption, the attacker needs to process and evaluate measured data. This processing and evaluation of measured information is called side channel analysis, and subsequently application of obtained information to abuse the cryptographic device is called attack. There are two basic methods of side channel analysis, simple and differential.

Simple power analysis (SPA) was characterized by Kocher in [10]. During these attacks, the attacker tries to determine the secret key more or less directly from a power trace measured. This can make SPA attacks quite challenging in practice but the potential attackers require detailed knowledge about the implementation of the cryptographic algorithm that is executed by the device under attack. In the most extreme case, this means that the attacker attempts to reveal the secret key based on one single power trace. In most cases, it is necessary to use statistical methods to extract the useful signal. The SPA attacks are feasible on condition that the secret key must have a significant impact on the power consumption in the attacked device. A typical example of the SPA is the attack on the implementation of the asymmetric cryptographic algorithm RSA (Rivest Shamir

Adleman), where the difference in power consumption between the operations of multiplication and squaring can been observed [13] (implementation of Square and multiply algorithm). The paper [14] discussed SPA of DES (Data Encryption Standard), which was focused on determination of the Hamming weight of the encryption key to reduce the space needed for a brute force attack. Subsequent works were naturally focused on AES implementation, e.g. the paper [15] described the analysis measured power trace during execution S-box operation and cache access. Mangard discussed the SPA which was focused on operation key expansion in [16]. Another type of SPA attacks are template based attacks which were introduced in [17]. Practical aspects of these attacks have been discussed in the works [18], [19], [20], [21]. Template based attacks characterize power consumption with templates which represent mean vector and a covariance matrix of multivariate normal distribution. The attacker assumes that it is possible to determine templates for certain instructions and certain sequences of instructions. If we summarize the knowledge about SPA attacks, it is evident that there must be a strong direct correlation between power consumption and secret key in the cryptographic module (typical example implementation of the RSA algorithm). In this case, the attacker is able to establish the specific secret key value from one measured power trace. For nowadays ordinarily used standard encryption algorithm AES [22], SPA method can not determine the value of a secret key but the method can provide the attacker sensitive information such as Hamming weight of the secret keys to reduce the space needed for a brute force attack.

The goal of differential power analysis (DPA) attacks is to reveal secret keys of the cryptographic module based on a large number of the power traces that have been recorded while the device carried out encryption or decryption operation for different input data. The main advantage compared to SPA is that the attacker does not need any detailed knowledge about the device and encryption algorithm. Therefore, DPA is the most popular type of power analysis. Another important difference between SPA and DPA lies in data processing method. DPA analyzes how the power consumption at fixed time moments depends on the processed data and the shape of the traces (power patterns) is not important at all. The concept of DPA attack was first described also in [10]. The basic principle was introduced on DES algorithm using the statistical method Difference of Means. Subsequently, possible applicable statistical tests were discussed in [23]. Fundamental simulation power models, that are an essential part of DPA, were presented for the first time in [24] and analyzed in the context of smart cards in [25]. The models are progressively modified to improve the efficiency of power analysis [26], [27], [28]. Important question of the impact of preprocessing of measured data on the effectiveness of DPA was presented in [29], [30]. Brier observed that the correlation coefficient can be also used as statistical method in DPA [31]. Nowadays, this method is one of the most widely used and best known [32]. If we summarize the knowledge about DPA attacks, it is evident that the attacker needs a large number of measured power traces. On the other hand, the attacker does not need detailed knowledge about the algorithm and the device. Statistical methods enable to obtain the exact value of the secret key even if the recorded power traces are extremely noisy.

Power analysis as presented in this paper is a completely new approach in power analysis attacks. It uses a typical three-layer neural network with the back propagation learning algorithm to determine the secret key from one measured power trace. Proposed method does not use any statistical methods to determine secret key value. Our approach is based on the knowledge of template attacks, which characterize the power consumption by using the templates for certain instructions associated with the secret key value, and on the application of NN for classification in acoustic side channel. The proposed power analysis is focused on the AES algorithm because it is an encryption standard and the algorithm is resistant to conventional method of cryptanalysis and to a great extent in SPA. The method was designed with the vision to combine the advantages of simple and differential power analysis. In the extreme case, this feature allows to determine the exact secret key value of AES algorithm only from one measured power consumption. Thus, the proposed method combines the characteristic advantages of SPA and DPA methods. Experimental implementation of power analysis was focused on power trace of the operation `AddRoundKey` and operation `SubBytes` in the initialization phase of the algorithm, in which the algorithm works with the secret key. The measurement of the power consumption was performed on a test bed which is described in detail in papers [33], [34]. The authors build on their own work where they implemented and tested well known SPA and DPA alternatively SEMA and DEMA [33], [34], [35]. We also discussed in [36] the classification of acoustic side channel by neural network. The paper [4] deals with reverse engineering from power trace with neural network but this work was focused only on few instructions. In presented proposal, we combined all previously acquired knowledge and experience about side channels analysis. In this work, we first show the proof of concept of this method and we intend to continue in research.

2. Method Design

The general goal of the proposed method is to obtain the secret key value which is stored in the cryptographic module from the measured power trace. In the following text, we denote the value K_{sec} as secret key stored in the attacked cryptographic module and K_{est} represents the estimate value of secret key, which was determined with neural network. Naturally, if the method works correctly, the values K_{est} and K_{sec} will be equal at the end of classification process. The first proposal of the method excepted sequential classification. In the other words, classification is realized

byte by byte similarly as in most DPA attack. The secret key could be byte expressed as follows: $K_{sec} = \{k_1, k_2, \ldots, k_{16}\}$ for $0 \leq k_i \leq 255$ where $i = 0$ to 16 represents each step of the method. The proposed method determines the first byte value k_1 of the secret key in the first step and the second byte value k_2 in the second step and so on. The difference between each step is in power trace division into parts corresponding to time interval where the cryptographic device has worked with individual bytes. General scheme of the whole method was divided into three phases:

- preparation of patterns for the secret key k_i,
- preparation and training of neural network,
- classification of key estimates.

Realization of these phases allows the attacker to implement one step of analysis, thus determining one byte of the secret key k_i.

At the beginning, the attacker has to prepare a training set to train the neural network. The attacker must know the type of cryptographic module on which he wants to realize an attack. Typical example of suitable cryptographic module represents smart cards with implemented cryptographic algorithm [37], [38], [39]. The attacker needs to have the same type of module completely under control (provided that the attacker is going to attack the smart card containing a microcontroller PIC16F84, he must own the same type of the card). If the attacker has matching cryptographic device with implemented cryptographic primitives, it is possible to measure and record the power consumption for all variants of secret key k_i. It is not purposeful to measure whole power waveforms but it is better to locate some important operations where the cryptographic module works with intermediate result and secret key. For example, convenient place for AES algorithm represent operations `AddRoundKey` and `SubBytes`.

The attacker uses the recorded power traces to train the neural network. The train set corresponds to all possible variants of the secret keys, thus the neural network should be able to identify the correct key. After successful teaching of neural networks, the attacker can continue with the last phase of the method.

In the last phase, the attacker measures the power consumption device under attack and inserts the measured power trace on input of neural network. The neural network assigns to power consumption the probability vector that contains probabilities for all key estimates. The estimate key with the highest probability should be equal to secret key value. In this way, the value of a secret key k_i is determined. The following text describes the particular phases of the proposed analysis, including the implementation and results of classification. Experimental implementation of power analysis was focused on power trace of the operation `AddRoundKey` and operation `SubBytes` in the initialization phase of the algorithm, in which the algorithm works with the secret key.

2.1 Preparation of Patterns

The goal of this phase is to get the training patterns of power consumption for the operation `AddRoundKey` and `SubBytes` for all variants of secret key k_1 (256 possible variants). Complete AES algorithm was implemented into the cryptographic module and the synchronization was performed only for above written operations according to previously validated knowledge of the algorithm AES and cryptographic module. The program worked in the loop and before every single round, the data k_1 was loaded in the memory of microprocessor to always work with same input variables. The program allowed increment or decrement the value of the key and indicated this operation by sending the value via serial port to the computer. Synchronization signal and communication with the PC did not affect the power consumption.

Fig. 1 shows the measured power traces matching to implemented operation `AddRoundKey` and `SubBytes` for key value 1 and 255. Power waveforms are almost identical except for two places. The first lies at the beginning of the trace and corresponds to the operation `XOR` of the plaint text and secret key during the `AddRoundKey` operation. The second place is located around the time $t = 35000$ which corresponds to executing instruction during operation `SubBytes`. In Fig. 1, the sections, which are influenced by the change of the first key byte are clearly recognizable. According to the information mentioned in Sec. 2, the power traces were divided into the parts. The numbers indicate the resulting parts that correspond to the work with bytes that also correspond with steps of the method. Fig. 2 shows resulting power patterns for all values of the secret key cut for the first byte. A detail of the power peak is shown in Fig. 3 and it is clear that measured power traces are greatly synchronized and divided into several groups, according to the Hamming weight of the secret key.

Neural networks used for classification in the acoustic side channel were trained for specific courses of acoustic signals [2]. This method requires sufficient differences between corresponding waveforms and repeatability of acoustic trace measurement. These two important assumptions are not achieved in measured power traces because of two reasons. First, the power traces of each instruction are very similar [40]. Second, if we measure power consumption for specific instructions more times, the resulting power traces are not completely identical due to the changes in auxiliary registers of cryptographic module (for example the incrementing of program counter register). This property is known as the electronic noise, which seriously affects the results of PA. When the attacker is preparing patterns during the attack phase, it is necessary to reduce electronic noise to a minimum value otherwise the classification achieves bad results. It can easily happen that measured power trace determined to classification will be put to another group corresponding with Hamming weight (Fig. 3). Based on knowledge of PA, the best way to reduce electronic noise is to repeat the mea-

surement of power consumption and the subsequent calculation of average values. Therefore, the power consumption for different data values was measured more times and subsequently the average power consumption was calculated. It was experimentally verified that the optimal value of averaging is 16. In reality, this calculation was conducted using the digital oscilloscope.

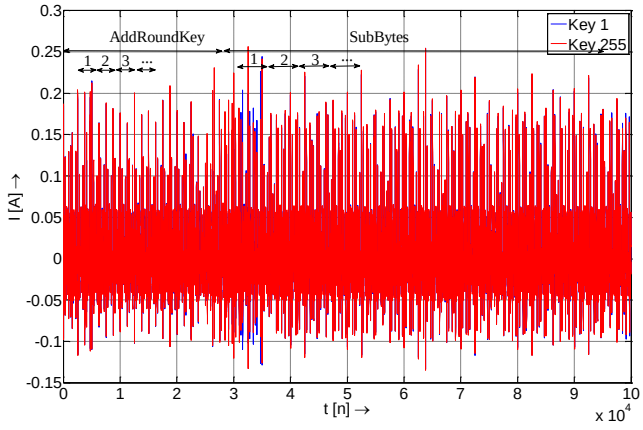

Fig. 1. Measured power traces for two key values.

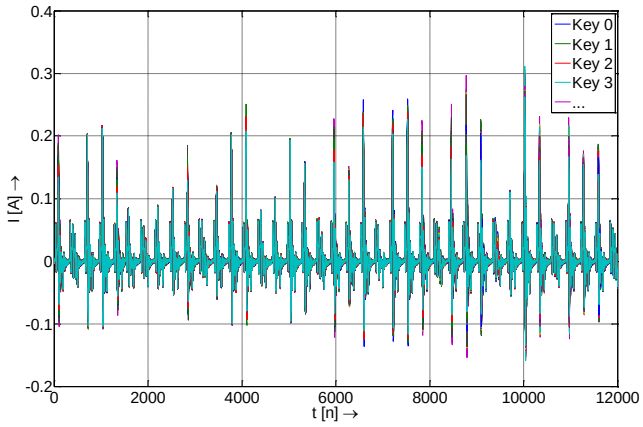

Fig. 2. The power pasterns for all key values.

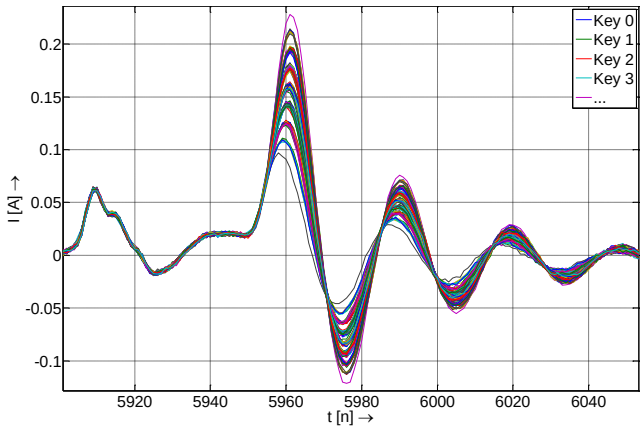

Fig. 3. Detail of power peak.

2.2 Preparation and Training of NN

The neural network was implemented in Matlab. This environment provides a wide range of the possibilities for the implementation of the mathematical methods, signal processing, simulation and testing. Another great advantage is the availability of toolboxes that solves a specific problem. For the implementation of the neural networks, neural network toolbox called Netlab was used. The authors of this toolbox are Nabney Ian and Christopher Bishop from Aston University in Birmingham. The toolbox is free to download [41]. 256 measured power patterns were imported and stored in the matrix **P** in Matlab for subsequent processing. This chapter describes the main points of implementation. The created neural network was a typical three-layer neural network with Backpropagation learning method. The input layer must have the same number of neurons as the number of samples in the measured power trace, for our purpose 12,000. Output layer classifies the input vector to key value and therefore it must contain 256 neurons for all combinations of key values 0 to 255. Hidden layer could contain various number of neurons depending on the complexity of the problem. In our implementation, we used 100 neurons in the hidden layer. Of course, we tested the influence of neurons in hidden layer from 50 to 256 to the result of classification. The chosen configuration containing 100 neurons in hidden layer was balanced in terms of the time demands for learning and of correct classification. If the hidden layer contained smaller number of neurons the results of classification were poor. On the other hand, the results were not better and training process was much more time-consuming if the number of neurons was higher. As activation function the standard sigmoid was chosen (parameter ”logistic” in source code). The following text contains the most important part of the program implementation.

```
%Creation of neural network
nn = mlp(input, hidden, output,'logistic');
%Setting
options = zeros(1,18);
options(1) = true;
options(14) = iteration;
%Training
[nn, options] = netopt(nn, options, ...
                ... P, clas, 'scg');
```

The program lines correspond to creation, setting and training of neural network. The training process needs a classification matrix which determines the correct classification of a given input to the appropriate key. The size of the classification matrix was 256×256 because the number of lines corresponds with the number of NN outputs, and number of columns corresponds with the number of power pasterns. Elements equal to 1 in the matrix assign power patterns to appropriate outputs values. In our case, the classification matrix was unit matrix. After successful training, neural networks is ready to carry out the attack phase.

2.3 Classification of Key Estimates

In the attack phase, the attacker needs to measure at least one of the power consumption traces of the device under attack. In a real attack, the attacker would measure power consumption of the cryptographic module and he would try to isolate the part of algorithm on which neural network is trained. We assumed that the attacker knows the type of the attacked cryptographic device and the algorithm. Thus, the same implementation of the algorithm and the same synchronization signal for training and attack data is used. Provided the attacker has this option, the power trace measurement and attack will be easy to perform. This section is considered as critical and it is important that the data used to train and attack are synchronized in the same way. Unless, the attacker has no such possibilities, the attack will be still feasible but the power traces measurement and processing will be harder. The attacker can measure the whole power trace of the algorithm and gradual analysis will find the necessary operations in the power trace. It is not hard to synchronize this roughly trimmed power trace, for example to the first power peak. Positive factor affecting the classification results is compliance to the same procedure of electronic noise reduction (described in Sec. 2.2). After a correct power trace measurement corresponding to the value K_{sec}, classification is performed and neural network classifies the first byte of secret key like the output with the highest probability.

For method verification, we measured once again the whole set of power traces corresponding to all the secret key values and this set was subsequently analyzed with the neural network. In this manner, we obtained the classification results for all possible key values and the first idea of how successful the method is. The following part of the source code shows the classification of measured test data.

```
for i=1:256
V_total = [V_total; mlpfwd(nn,test(i,:))];
end
```

The result of analysis for all power traces was matrix $\mathbf{V_{total}}$ of size 255×255. The lines index corresponds to the value of a secret key K_{sec} and columns index corresponds to the value of key estimation K_{est}. In other words, neural network assigned to measured power trace probability vector for individual key estimates. Whole matrix $\mathbf{V_{total}}$ of classification is shown graphically in Fig. 4 and for better understanding it is the part of the matrix written in Tab. 1.

From Tab. 1, it can be seen that the neural network classified the power trace corresponding $K_{sec} = 0$ with probability 36.77 % to key estimate $K_{est} = 0$ and the power trace corresponding to the secret key 1 classified key estimate 1 with probability 66,42 % and so on. Each line of the matrix $\mathbf{V_{total}}$ corresponds to the output probability vector which is result of power trace classification. Each column contains probability for individual estimation key value.

		Probability of key estimation K_{est}			
		0	1	2	3
K_{sec} value	⋮	…	…	…	…
	6	0,00%	0,00%	0,00%	0,00%
	5	0,00%	0,00%	0,08%	0,00%
	4	0,00%	0,00%	0,00%	0,00%
	3	0,00%	0,00%	0,00%	23,79%
	2	0,00%	0,00%	6,44%	0,00%
	1	0,00%	66,42%	0,00%	0,00%
	0	36,77%	0,00%	00,00%	0,00%

Tab. 1. Part of the resulting matrix $\mathbf{V_{total}}$.

To obtain a better understanding of the whole classification results, the graph displayed in Fig. 5 shows the classification results (output probability vectors) for the five randomly chosen secret keys. Appropriate probability vectors for chosen $K_{sec} = 5, 41, 81, 129, 248$ values are color distinguished. From Fig. 5, it can be read that the probability vector contained five possible key estimations for power trace $K_{sec} = 5$. Classified key estimations were $K_{est} = 5$ with probability 35 %, $K_{est} = 18$ with probability 5 %, $K_{est} = 74$ with probability 6 %, $K_{est} = 76$ with probability 23 % and $K_{est} = 105$ with probability 7 %. Analogously, probability for a different secret key value could be determined. Key estimation with highest probability was always equal to secret key value for this random selected values. These partial results show good functionality of the proposed method. However, it was necessary to investigate all selected key estimations with the highest probability.

Fig. 6 shows the selected highest probability value of keys estimations for classified power trace corresponding to all values of the secret key. In other words, which key value was classified with highest probability for a specific power trace. The graph is displayed with two Y-axes for better clarity. X-axis represents secret key values and blue Y-axis shows the probability of highest selected probability and the red Y-axis corresponds to the chosen key estimation.

K_{sec} [value]	2	3	18	84	114	120
K_{est} [value]	112	33	10	82	106	10
P_{max} [%]	8.32	27.77	7.31	19.15	21.23	13.60
Δ_{err} [%]	0.44	4.00	0.60	0.50	2.00	2.00
K_{taj} [value]	151	173	195	199	210	234
K_{est} [value]	253	171	197	228	202	206
P_{max}[%]	31.57	13.23	6.02	6.44	45.59	18.27
Δ_{err} [%]	6.17	0.20	1.20	0.80	5.00	2.50
K_{sec} [value]	149	150	245	251		
K_{est} [value]	249	250	207	223		
P_{max}[%]	9.20	18.59	7.82	16.60		
Δ_{err} [%]	1.30	4.50	1.90	6.00		

Tab. 2. Classification errors.

The main goal of the method is to have the estimate key value equal to secret key value after classification. In the other word, the function $K_{est} = K_{sec}$ is true. Shape of this function is clearly visible at first sight in Fig. 6 and only a few points interrupt linear progression of the function. These points are estimates that were wrongly classified, when the

selected estimate with highest probability did not correspond to the secret key value ($K_{est} \neq K_{sec}$). From these complete classification results, very hopeful functionality follows the method. Enumeration of erroneously classified secret keys and determined key estimation given in Tab. 2. The highest probability of key estimation P_{max} and difference between the highest probability and probability for correct estimate Δ_{err} are written in the table. From the obtained data, it is clear that the value Δ_{err} was really small for every wrong key estimation. The average value of Δ_{err} was only 2.46 %.

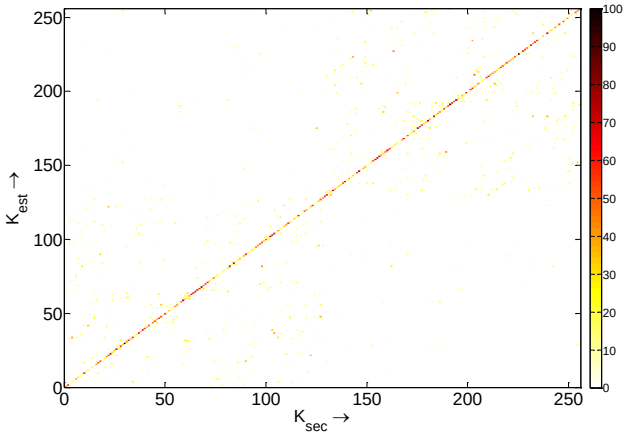

Fig. 4. Graphically depicted complete classification results.

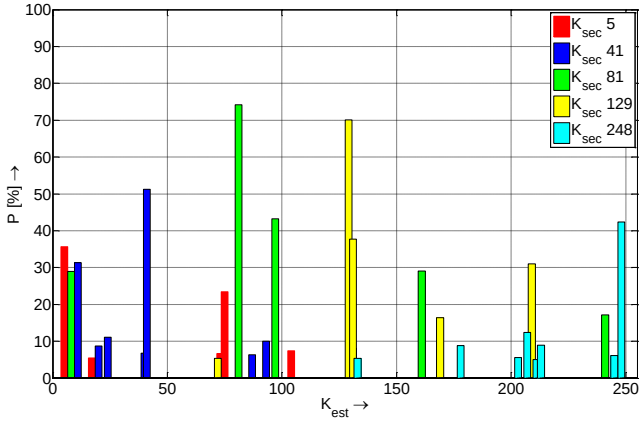

Fig. 5. Output probability vectors for the five randomly chosen secret keys.

From the whole proofing set (256 measured power traces), the neural network classified wrong key estimate sixteen times. This number of mistakes corresponds to 6.27 % of measured and tested power trace. Therefore, we can declare that the proposed method identified the correct value of the secret key in 93.72 % of cases. The method achieved similar results around 90…95 % of sucsesful clasification during the repeated tests and the classification errors occurred for key estimate with low value of the highest probability. This observation is confirmed by the data in Tab. 2, median of highest probability that led to the wrong classification is 15 % and the average value is 17 %. We can determine that for estimates keys classified with a probability lower than 20 %, probability of wrong clasification is higher.

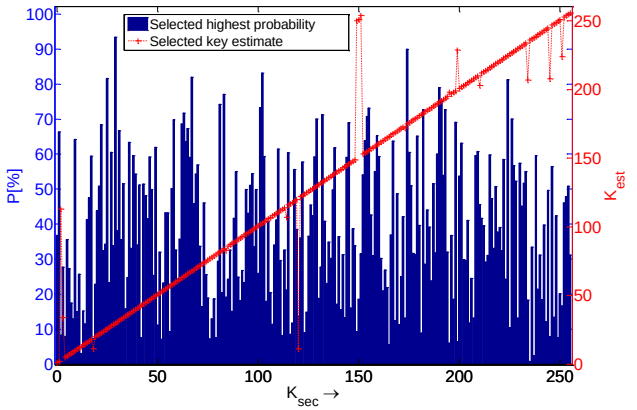

Fig. 6. The highest probabiliteis values and estimated key.

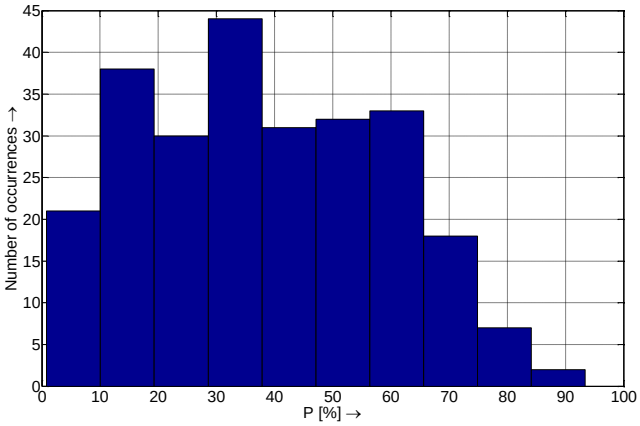

Fig. 7. Histogram of highest probabilities.

The probability of correct key estimate should be as high as possible to improve the classification results. Figure 6 shows that the occurrences of probabilities 14 %, 18 % and 20 % are not an exception. Therefore, it was decided to analyze the results in detail. Fig. 7 displays a histogram of selected highest probabilities. From the histogram, it can be seen that the occurrence of probabilities to 10 % occured twenty-one times and the probability 10 – 20 % occured thirty-eight times which was together 59 occurrences from all 256 values. The total number of potentially predisposed keys to incorrect classification is about 23 % which would mean that the proposed method would work with success about 80 %. This result is not sufficient enough, therefore the optimization of method will be subject to further research.

A very important and interesting factor, which flows from the low values Δ_{err}, was that the probability of correct key estimation was always the second highest probability for all erroneously classified keys. The example of probability vector which corresponds to value $K_{sec} = 151$ is shown in Fig. 8. This great feature, the attacker would use for reduction of the space needed for a brute force attack if it happened that the key was bad classified at the end of the classification process. Let us assume that the attacker attacks the AES algorithm working with 16 bytes (128 bits) secret key

and after classification of the whole secret key, the attacker tests decrypt captured cryptogram and from device output, the nonsensical text appears. In this situations, the attacker has two possibilities. First, he could check the classification results and he cloud try for key estimation with lower probability than 20 % the second key estimation. In the worst case, he would have to try two possibilities for every key byte. That means that this method reduces the space needed for a brute force attack from 2^{128} to 2^{16}, which corresponds to the reduction of 34 orders.

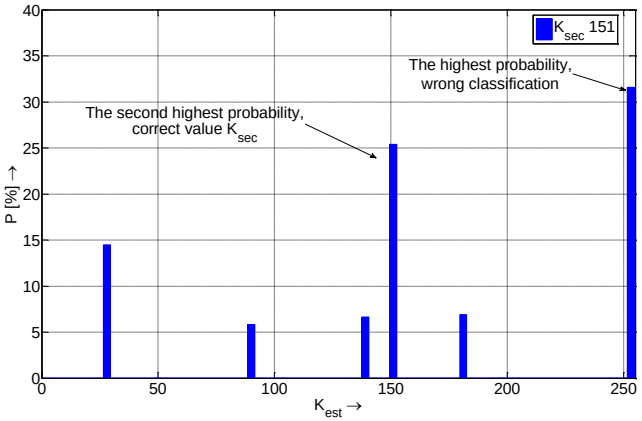

Fig. 8. Probability vector for $K_{sec} = 151$.

3. Repeatability of Realization

We measured ten times larger set of power consumption to analyze a possibility of the method repeatability and method feasibility. The main goal was to test if the results of classification are the same for more measured power traces corresponding to the same secret key value. 2560 power traces were measured corresponding to all values of the secret key. Ten power traces were independently stored for each value of the secret key. This power traces were not measured successively and were not measured in same day because we wanted to test the impact of workplace reconfiguration. Created set was classified with neural networks in the same manner as described in the previous section. In this way, we obtained the results for all possible secret key values from independent measurements. The number of wrong classified key estimation and the overall success rate are given in the following Tab. 3.

The number of wrong classified key estimation (from 2560)	Success [%]
378	85,23

Tab. 3. Classification errors for 2560 power traces.

For completion of the results, Tab. 4 gives classification results for seven selected keys. The first five selected keys (5, 41 , 81, 129 and 248) correspond to the key selected in the previous chapters and the selected highest probabilities are almost identical. As an example of incorrect classifications, the classification results for value $K_{sec} = 19$ are displayed. Selected key estimates were classified with low probability therefore the neural network wrongly classified five times from ten power traces measured. The results confirmed that repeated measurements do not influence classification results and it is possible to use the method. In this case, the proposed method reached 85 % of the successful classification of secret key.

K_{taj}	**5**	**41**	**81**	**129**	**248**	**19**
Probabilities [%]	28,74	38,20	79,92	67,46	30,07	7,67
	27,21	41,48	79,99	67,68	39,26	17,27
	27,10	39,78	80,51	67,87	36,55	11,49
	28,96	42,19	80,15	69,02	31,05	9,30
	23,03	41,37	79,93	68,02	38,97	7,73
	28,81	31,34	77,83	67,94	37,95	12,63
	23,75	36,28	80,03	67,83	34,38	8,73
	26,95	37,32	77,32	66,56	36,04	13,85
	22,02	33,39	80,30	67,91	39,25	8,27
	28,44	38,25	80,43	67,99	34,82	7,81

Tab. 4. Results of classification.

4. Future Work

This method achieved similar results around 90 % of the successful classification during the repeated tests but from the detailed analysis of results it was established that total number of potentially predisposed keys to incorrect classification is about 23 % which would mean that the proposed method would work with success about 80 %. Experiment performed with 2560 power traces reported success equal to 85% and confirmed this fact. This result is not sufficient enough, therefore optimization of the method will be proposed. We intend to use the knowledge and experiments from [4] to increase the difference between the measured power traces. We have already performed the first test and the results confirmed that the optimized method is able to work with success rate around 96 % and the correct key estimate is determined with probability more than 90 % This optimization will be subject to further research.

The proposed method assumes that the attacker must know the type of cryptographic module on which he wants to realize an attack. This requirement is considered critical and it is important that the data used to train and attack are synchronized in the same way. If we assume that the attacker knows the attacked cryptographic device and the algorithm, the same implementation of the algorithm and the same synchronization signal for training and attack data can be used. In this case, the realization of the proposed attack is not difficult. If the attacker has no such possibilities, the attack will be still feasible but measuring the power traces and processing will be harder. The attacker can measure the whole power trace of an algorithm and the gradual analysis will find the necessary operations in the power trace. It is not hard to synchronize this roughly trimmed power trace, for example to the first power peak. These two realizations of the attack suppose the identical type of the cryptographic module. Therefore, we want to focus on verification of the

method for different types of the cryptographic modules in the future research.

In our first experiment, we focus on the first byte of the secret key because all subsequent bytes are involved in the same operations as the first one. Therefore, we are confident that the proposed method is able to classify the whole AES secret key from only one measured power trace. However, we would like to verify the application of only one trained neural network to classify whole secret key using experimental testing.

5. Conclusion

This paper presented the innovative method of the power analysis attack which used the neural network to classify measured power traces. If we compare our method with well known methods used in DPA and SPA, the main advantage is that the method has the ability to determine the first byte of the secret key with probability around 90 % using only one measured power trace for an algorithm resistant to conventional analysis method or SPA method. This way of the power analysis is an entirely new approach and is designed to combine the advantages of simple and differential power analysis. We proved experimental applicability of the method on power analysis of the operation `AddRoundKey` and operation `SubBytes` in the initialization phase of the AES algorithm. The proposed method identified the correct value of the secret key in 93.72 % of cases. In the worst case, the attacker is able to reduce the space needed for a brute force attack from 2^{128} to 2^{16} which corresponds to the reduction of 34 orders.

According to the future plans, we would like to improve the correct classification of the method. The improvement will be based on increasing of the difference between the measured power traces. First tests confirmed that the method is able to work with success rate around 96 % and the correct key estimate is determined with probability more than 90 % We intend to use our own knowledge and experiments from [4]. In this way, we want to determine the whole secret key of a cryptographic module only from one measured power trace.

Acknowledgements

This research work is funded by projects SIX CZ.1.05/2.1.00/03.007; the Technology Agency of the Czech Republic projects TA02011260 and TA03010818; the Ministry of Industry and Trade of the Czech Republic project FR-TI4/647.

References

[1] HIU, A., FIONA, Y. *ERG4920CM Thesis II Keyboard Acoustic Triangulation Attack*. PhD thesis. Hong Kong (China): University of Hong Kong, 2006.

[2] ZHUANG, L., ZHOU, F., TYGAR, J. D. Keyboard acoustic emanations revisited. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*. New York (NY, USA), 2005, p. 373 - 382.

[3] KUR, J., SMOLKA, T., SVENDA, P. Improving resiliency of java card code against power analysis. In *Mikulaska kryptobesidka, Sbornik prispevku*. 2009, p. 29 - 39.

[4] MARTINASEK, Z., MACHA, T., ZEMAN, V. Classifier of power side channel. In *Proceedings of NIMT*. 2010.

[5] LIU, N., GUO, D. Security analysis of public-key encryption scheme based on neural networks and its implementing. In *International Conference on Computational Intelligence and Security.* Guangzhou (China), 2006, vol. 2, p. 1327 - 1330.

[6] KIM, H.-M., KANG, D.-J., KIM, T.-H. Flexible key distribution for scada network using multi-agent system. *ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security (BLISS)*. Edinburgh (UK), 2007, p. 29 - 34.

[7] LIAN, S., SUN, J., WANG, Z. One-way hash function based on neural network. *CoRR abs/0707.4032* (2007).

[8] WANG, Y.-H., SHEN, Z.-D., ZHANG, H.-G. Pseudo random number generator based on hopfield neural network. In *International Conference on Machine Learning and Cybernetics*. Dalian (China), 2006, p. 2810 - 2813.

[9] MISLOVATY, R., PERCHENOK, Y., KANTER, I., KINZEL, W. Secure key-exchange protocol with an absence of injective functions. *Physics Review E*, 2002, vol. 66, p. 066102.

[10] KOCHER, P. C., JAFFE, J., JUN, B. Differential power analysis. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO)*. London (UK), 1999, p. 388 - 397.

[11] MANGARD, S., OSWALD, E., POPP, T. *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Springer, 2007.

[12] MARTINASEK, Z., PETRIK, T., STANCIK, P. Conditions affecting the measurement of power analysis. In *The 13th International Conference on Research in Telecommunication Technologies (RTT).* Techov (Czech Republic), 2011, p. 1 - 5.

[13] JOYE, M., OLIVIER, F. Side channel analysis. *Encyclopedia of Cryptography and Security (2nd Ed.)*. 2011, p. 1198 - 1204.

[14] MESSERGES, T. S., DABBISH, E. A., SLOAN, R. H., MESSERGES, T. S., DABBISH, E. A., SLOAN, R. H. Investigations of power analysis attacks on smartcards. In *USENIX Workshop on Smartcard Technology*. 1999, p. 151 - 162.

[15] BERTONI, G., ZACCARIA, V., BREVEGLIERI, L., MONCHIERO, M., PALERMO, G. AES power attack based on induced cache miss and countermeasure. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*. Washington, DC (USA), 2005 vol. I, p. 586 - 591.

[16] MANGARD, S. A simple power-analysis (SPA) attack on implementations of the AES key expansion. In *5th International Conference on Information Security and Cryptology (ICISC)*. Seoul (Korea), 2002, p. 343 - 358.

[17] CHARI, S., RAO, J. R., ROHATGI, P. Template attacks. In *4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*. Redwood Shores (CA, USA), 2002, p. 13 - 28.

[18] RECHBERGER, C., OSWALD, E. Practical template attacks. In *5th International Workshop on Information Security Applications (WISA)*. Jeju Island, (Korea), 2004, p. 443 - 457.

[19] AGRAWAL, D., RAO, J., ROHATGI, P., SCHRAMM, K. Templates as master keys. In *7th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)* Edinburgh (UK), 2005, p. 15 - 29.

[20] HANLEY, N., TUNSTALL, M., MARNANE, W. P. Using templates to distinguish multiplications from squaring operations. *International Journal Information Security*, 2011, vol. 10, no. 4, p. 255 - 266.

[21] OSWALD, D., PAAR, C. Breaking Mifare DESFire MF3ICD40: Power analysis and templates in the real world. In *Proceedings of the 13th International Conference on Cryptographic Hardware and Embedded Systems (CHES)*. Nara (Japan), 2011, p. 207 - 222.

[22] Federal information processing standards publication (FIPS 197). *Advanced Encryption Standard (AES)*, 2001.

[23] CORON, J.-S., NACCACHE, D., KOCHER, P. Statistics and secret leakage. *ACM Transactions on Embedded Computing Systems (TECS)*, 2004, vol. 3, no. 3, p. 492 - 508.

[24] CHARI, S., JUTLA, C., RAO, J. R., ROHATGI, P. A cautionary note regarding evaluation of AES candidates on smart-cards. In *Second Advanced Encryption Standard (AES) Candidate Conference*. Rome (Italy), 1999, p. 133 - 147.

[25] AKKAR, M.-L., BEVAN, R., DISCHAMP, P., MOYART, D. Power analysis, what is now possible... In *Advances in Cryptology - ASIACRYPT 2000*. Kyoto (Japan), 2000, p. 489 - 502.

[26] PEETERS, E., STANDAERT, F.-X., QUISQUATER, J.-J. Power and electromagnetic analysis: Improved model, consequences and comparisons. *Integration, the VLSI Journal – Special Issue: Embedded Cryptographic Hardware*, 2007, vol. 40, no. 1, p. 52 - 60.

[27] MORADI, A., SALMASIZADEH, M., MANZURI SHALMANI, M. T., EISENBARTH, T. Vulnerability modeling of cryptographic hardware to power analysis attacks. *Integration, the VLSI Journal*, 2009, vol. 42, no. 4, p. 468 - 478.

[28] FEI, Y., LUO, Q., DING, A. A statistical model for DPA with novel algorithmic confusion analysis. In *Cryptographic Hardware and Embedded Systems (CHES)*. Leuven (Belgium), 2012, p. 233 - 250.

[29] JOYE, M., PAILLIER, P., SCHOENMAKERS, B. On second-order differential power analysis. In *Cryptographic Hardware and Embedded Systems (CHES).* Edinburgh (UK), 2005, p. 293 - 308.

[30] HERBST, C., OSWALD, E., MANGARD, S. An AES smart card implementation resistant to power analysis attacks. In *Second International Conference on Applied Cryptography and Network Security (ACNS)*. Singapore, 2006, p. 239 - 252.

[31] BRIER, E., CLAVIER, C., OLIVIER, F. Correlation power analysis with a leakage model. In *Cryptographic Hardware and Embedded Systems (CHES)*. Cambridge (USA), 2004, p. 16 - 29.

[32] CLAVIER, C., FEIX, B., GAGNEROT, G., ROUSSELLET, M., VERNEUIL, V. Improved collision-correlation power analysis on first order protected AES. In *Cryptographic Hardware and Embedded Systems (CHES).* Nara (Japan), 2011, p. 49 - 62.

[33] MARTINASEK, Z., MACHA, T., RASO, O., MARTINASEK, J., SILHAVY, P. Optimization of differential power analysis. *Przeglad Elektrotechniczny*, 2011, vol. 87, no. 12, p. 140 - 144.

[34] MARTINASEK, Z., ZEMAN, V., SYSEL, P., TRASY, K. Near electromagnetic field measurement of microprocessor. *Przeglad Elektrotechniczny*, 2013, vol. 89, no. 2, p. 203 - 207.

[35] MARTINASEK, Z., ZEMAN, V., TRASY, K. Simple electromagnetic analysis in cryptography. *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems*, 2012, vol. 1, no. 1, p. 1 - 6.

[36] MARTINASEK, Z., MACHU, P. New side channel in cryptography. In *Proceedings of the 17th Conference Student EEICT 2011.* Brno (Czech Republic), 2011.

[37] HAJNY, J., MALINA, L. Anonymous credentials with practical revocation. In *IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*. Rome (Italy), 2012, p. 1 - 6.

[38] HAJNY, J., MALINA, L. Unlinkable attribute-based credentials with practical revocation on smart-cards. In *Proceedings of the 11th International Conference on Smart Card Research and Advanced Applications (CARDIS)*. Graz (Austria), 2012, p. 62 - 76.

[39] MALINA, L., HAJNY, J. Accelerated modular arithmetic for low-performance devices. In *34th International Conference on Telecommunications and Signal Processing (TSP)*. Budapest (Hungary), 2011, p. 131 - 135.

[40] AMBROSE, J., ALDON, N., IGNJATOVIC, A., PARAMESWARAN, S. Anatomy of differential power analysis for AES. In *10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. Timisoara (Romania), 2008, p. 459 - 466.

[41] NABNEY, I. T. *NETLAB: Algorithms for Pattern Recognition*. Springer, 2002.

About Authors ...

Zdenek MARTINASEK was born in 1984. He received his BSc. from the Department of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology in 2006. He received MSc. (Ing.) from the same department in 2008. Now he is a Ph.D. student at the same faculty. He also helps to cover pedagogically a Master's program course. The area of his professional interests is cryptography, side channel analysis, sensors and modern data communication.

Vaclav ZEMAN received MSc. (Ing) from Faculty of Electrical Engineering and Communication, Brno University of Technology in 1991. He received Ph.D. from the Department of Telecommunications at the same Faculty in 2003. Now he is the Associate Professor (doc, 2005) at Faculty of Electrical Engineering and Communication, Brno University of Technology. He publishes in the cryptology area and communication systems area. He is also a lecturer and deals with cryptology and information system security.