IEICE TRANS. INF. & SYST., VOL.E107–D, NO.10 OCTOBER 2024

LETTER

Differential-Neural Cryptanalysis on AES*

Liu ZHANG[†,††], ***Student Member***, **Zilong WANG**[†,††], ***and*** **Jinyu LU**[†††a)], ***Nonmembers***

SUMMARY Based on the framework of a multi-stage key recovery attack for a large block cipher, 2 and 3-round differential-neural distinguishers were trained for AES using partial ciphertext bits. The study introduces the differential characteristics employed for the 2-round ciphertext pairs and explores the reasons behind the near 100% accuracy of the 2-round differential neural distinguisher. Utilizing the trained 2-round distinguisher, the 3-round subkey of AES is successfully recovered through a multi-stage key guessing. Additionally, a complexity analysis of the attack is provided, validating the effectiveness of the proposed method.

key words: *deep learning, differential-neural distinguisher, AES, key recovery attack*

1. Introduction

In CRYPTO 2019, Gohr introduced the concept of differential-neural cryptanalysis [1]. This technique enables a distinguisher to differentiate between ciphertexts encrypted from plaintexts with a specific input difference and those encrypted from random numbers. Gohr effectively integrated this distinguisher with classical differentials, facilitating a 12-round key recovery attack on `SPECK32/64` (from a total of 22 rounds). At EUROCRYPT 2021, Benamira [2] noted that Gohr's differential-neural distinguisher provides a robust approximation of the cipher's differential distribution table (DDT) and learns additional information beyond the DDT. In ASIACRYPT 2022, Bao et al. [3] refined the concept of neutral bits, extending key recovery attacks to 13 rounds for `SPECK32/64` and to 16 rounds (from a total of 32) for `SIMON32/64`. Further developments were presented at ASIACRYPT 2023, where Bao et al. [4] proposed specific rules that, when used in conjunction with the DDT, enhance the accuracy of DDT-based distinguishers for `SPECK32/64`. Additionally, they demonstrated that combining these rules does not improve the performance of the differential-neural distinguisher, suggesting that the distinguisher may already be leveraging these rules or their equivalent forms—underscoring the effectiveness of neural networks in cryptanalysis.

Most previous research has focused on ciphers with smaller block sizes. However, Yi Chen *et al.* [5] have proposed a multi-stage differential-neural cryptanalysis framework applicable to larger block ciphers, facilitating key recovery attacks on all ciphers in the `SPECK` family. In this study, we apply differential-neural cryptanalysis to AES, utilizing the multi-stage key recovery framework designed for large block ciphers.

Our Contribution. We have developed a differential-neural distinguisher for AES, which was trained using various subsets of ciphertext data, including full ciphertext, a single row or column, two bytes, and one byte. Currently, our capabilities are limited to training a 3-round differential-neural distinguisher. Our findings indicate that there is a proportional decrease in the true negative rate of the distinguisher as the amount of ciphertext used for training diminishes. Remarkably, using just two bytes of ciphertext, we have achieved a distinguisher accuracy close to 100%. We attribute this high level of accuracy to the effective capture of differential propagation characteristics inherent in AES. Furthermore, we utilize the trained 2-round distinguisher to conduct a 3-round key recovery attack on AES. The source codes for our experiments are publicly available at https://github.com/CryptAnalystDesigner/differential-neural-cryptanalysis-on-aes.git.

The remainder of this letter is organized as follows: Sect. 2 introduces the fundamental concepts of AES, describes the neural network architecture, outlines the training process, and presents the results of the distinguisher trained using the full ciphertext. Section 3 details the training and performance of the differential-neural distinguisher using partial ciphertexts. The methodology and results of the key recovery attack, utilizing the distinguisher trained with partial ciphertexts, are discussed in Sect. 4. Finally, Sect. 5 summarises our work.

2. Differential-Neural Distinguisher on AES-128

2.1 Description of AES-128

The AES [6] is a Substitution-Permutation network that supports key sizes of 128, 192 and 256 bits. The 128-bit plaintext initializes the internal state represented by a 4×4 matrix of bytes seen as values. Depending on the version of AES, N_r

Manuscript received May 8, 2024.
Manuscript publicized June 20, 2024.

[†]School of Cyber Engineering, Xidian University, Xi'an 710126, China.
[††]State Key Laboratory of Cryptology, P. O. Box 5159, Beijing, 100878, China.
[†††]College of Sciences, National University of Defense Technology, Hunan, Changsha 410073, China.

*The work is supported by the National Natural Science Foundation of China (No. 62172319, U19B2021) and Postgraduate Scientific Research Innovation Project of Hunan Province (No. CX20220016).

a) E-mail: jinyu_smile@foxmail.com

DOI: 10.1587/transinf.2024EDL8044

rounds are applied to the state: $N_r = 10$ for AES-128. A round function applies four operations to the state matrix:

- SubBytes (SB): applying the same 8-bit to 8-bit invertible S-Box 16 times in parallel on each byte of the state;
- ShiftRows (SR): cyclic shift of each row (i-th row is shifted by i bytes to the left);
- MixColumns (MC): multiplication of each column by a constant 4×4 invertible matrix over the field $GF(2^8)$;
- AddRoundKey (AK): XORing the state with a 128-bit subkey.

One round of AES can be described as $R(x) = AK \oplus MC \circ SR \circ SB(x)$. In the first round an additional AddRoundKey operation (using a whitening key) is applied, and in the last round the MixColumns operation is omitted.

2.2 Network Architecture

We employ the neural network architecture originally designed by Gohr for Speck32/64 [1], with adaptations made solely to accommodate the data format of AES ciphertext.

- **Input Representation.** The neural network takes a ciphertext pair (C, C') or parts thereof as input, reformatted into a $[2, N_{pb}]$ matrix which is then transposed.
- **Initial Convolution.** The input is processed through an initial convolution layer with width-1 and $N_f = 16$ channels, followed by batch normalization and a ReLU activation, producing a $[N_{pb}, N_f]$ matrix.
- **Convolutional Blocks.** Each block contains two convolution layers with N_f filters and a kernel size of $k_s = 3$, followed by batch normalization and a ReLU layer. A skip connection from the output of the block's final ReLU layer to its input enhances continuity and flow to the next block. The model includes five such blocks.
- **Prediction Head.** The prediction head features a fully connected layer with 64 neurons each in two segments ($d_1 = d_2 = 64$), culminating in a Sigmoid activation function for the output.

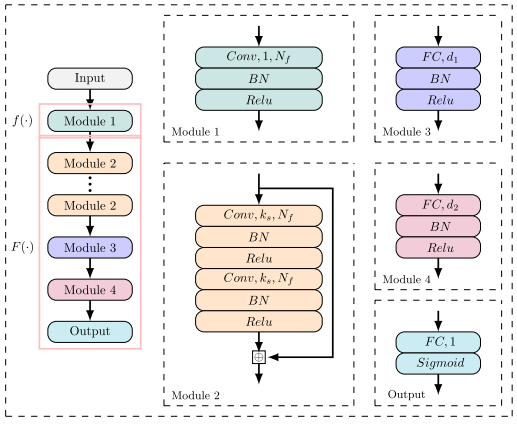

Fig. 1 Network architecture [1].

2.3 Model Training Process and Results

Training and test sets were generated using the Linux random number generator to produce uniformly distributed keys K_i and plaintext pairs (P_i, P_i') with an input difference of $\Delta = 0x80$, along with a vector of binary-valued labels Y_i. For each pair, if $Y_i = 1$, both plaintexts were encrypted for r rounds. If $Y_i = 0$, the second plaintext was replaced by a newly generated random plaintext and then encrypted for r rounds.

We trained over 20 epochs on a dataset consisting of $N = 10^7$ instances and $M = 10^6$ instances for testing. The batch size was set to $B_s = 1000$. Optimization was carried out against a cost function comprising mean square error loss augmented by an L2 regularization term with a parameter $\lambda = 10^{-5}$, using the Adam optimization algorithm. A cyclic learning rate schedule was employed, where the learning rate l_i for the i-th epoch was defined as: $l_i = \alpha + \left(\frac{(n-i) \mod (n+1)}{n}\right)(\beta - \alpha)$, with $\beta = 0.002$, $\alpha = 0.0001$, and $n = 9$. Networks obtained at the end of each epoch were archived, and the network exhibiting the lowest validation loss was subsequently evaluated against the test set. We report the accuracy (Acc), true positive rate (TPR), and true negative rate (TNR) of the distinguishers, as evaluated on newly generated datasets, in Table 1.

In Table 1, the 2 and 3-round differential-neural distinguishers are trained using a complete ciphertext pair. Notably, the ciphertext pair used for training the 3-round distinguisher does not undergo the MC operation in the final round of the encryption process, classifying the approach as a pure distinguish attack. However, the values for TPR and TNR are somewhat peculiar. All positive instances are predicted correctly, while negative instances are only predicted correctly with a probability of about $\frac{1}{16}$. This suggests that the differential-neural distinguisher might have learned a characteristic that all positive instances satisfy, whereas the negative instances, being random numbers, have a $\frac{15}{16}$ probability of satisfying it.

3. Distinguisher Using Partial Ciphertext on AES-128

According to the framework proposed by Gohr for key recovery attacks using differential-neural distinguishers [1], the entire subkey space must be guessed. However, the subkey space for AES amounts to 2^{128}, impractical for exhaustive search. Therefore, we employ the key recovery attack framework for large block ciphers proposed by Yi Chen et al. [5], which suggests dividing the subkey space into manageable

Table 1 Acc, TPR, TNR of differential-neural distinguisher on AES.

r	Acc	TPR	TNR
2	1.0	1.0	1.0
3	0.4987	1.0	0.0
3*	0.5309	1.0	0.0603

*: without MC in the last round

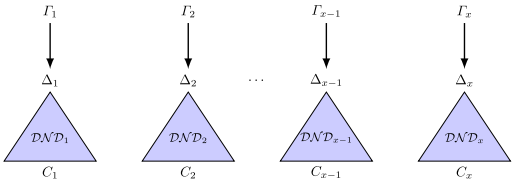

Fig. 2 The multi-stage key recovery framework for large state block ciphers.

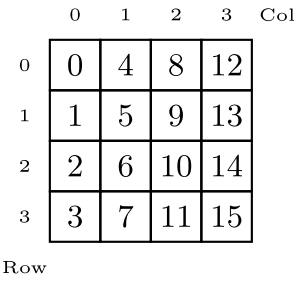

Fig. 3 The numbering of the AES state.

Table 2 Acc, TPR, TNR of 3-round distinguisher with one row and col.

Row	Acc	TPR	TNR	Col	Acc	TPR	TNR
0	0.5077	1.0	0.0156	0	0.5081	1.0	0.0154
1	0.5076	1.0	0.0155	1	0.5073	1.0	0.0155
2	0.5080	0.9953	0.0202	2	0.5066	0.9981	0.0172
3	0.5083	1.0	0.0157	3	0.5079	0.9999	0.0153

parts for sequential recovery. This approach is illustrated in Fig. 2 (where Γ_i or Δ_i, $i \in [1, x]$, can be the same).

In differential-neural cryptanalysis, it is essential to conduct experiments to verify the success of key recovery attacks. Notably, partial decryption techniques can be utilised since the last round of AES encryption does not involve the MC operation. Consequently, we train the differential-neural distinguisher using partial ciphertexts. It is important to highlight that during the subsequent training of the 3-round distinguisher, the final round of the encryption process also omits the MC operation.

3.1 3-Round Distinguisher Using Partial Ciphertext

We first present the state of AES with byte numbering in Fig. 3, to facilitate an understanding of the data used in the subsequent training process of the distinguisher. The parameter N_{pb} in the neural network is adjusted according to the size of the partial ciphertext used. This adjustment ensures the neural network architecture is optimally configured to handle the specific data inputs derived from the AES encryption process.

From Tables 1, 2, and 3, it is evident that the TNRs for the differential-neural distinguisher trained with different data subsets—namely, all ciphertexts, one column or one row of ciphertext, and two bytes—are approximately $\frac{1}{16}$, $\frac{1}{64}$, and $\frac{1}{128}$, respectively. The reduction in data volume corresponds proportionally with these TNR values. Despite the low overall accuracy of the distinguishers, these TNR values suggest that the distinguisher has learned a specific characteristic from the data.

Table 3 Acc, TPR, TNR of 3-round distinguisher with two bytes.

Index	Acc	TPR	TNR	Index	Acc	TPR	TNR
{0,1}	0.5040	1.0	0.0079	{8,9}	0.5045	0.9980	0.0097
{2,3}	0.5042	0.9987	0.0089	{10,11}	0.5033	0.9959	0.0116
{4,5}	0.5044	1.0	0.0078	{12,13}	0.5041	1.0	0.0079
{6,7}	0.5041	1.0	0.0080	{14,15}	0.5041	0.9891	0.0188

Table 4 Acc, TPR, TNR of 2-round distinguisher with one byte.

Index	Acc	TPR	TNR	Index	Acc	TPR	TNR
0	0.5136	0.5644	0.4628	8	0.5140	0.5517	0.4764
1	0.5120	0.5959	0.4280	9	0.5132	0.5613	0.4651
2	0.5144	0.5607	0.4679	10	0.5138	0.5141	0.5034
3	0.5131	0.5795	0.4465	11	0.5133	0.5621	0.4644
4	0.5137	0.5591	0.4683	12	0.5135	0.5664	0.4605
5	0.5146	0.5545	0.4747	13	0.5133	0.5554	0.4713
6	0.5134	0.6024	0.4244	14	0.5146	0.5225	0.5069
7	0.5128	0.5656	0.4603	15	0.5129	0.5418	0.4841

Table 5 Acc, TPR, TNR of 2-round distinguisher with two bytes.

Index	Acc	TPR	TNR	Index	Acc	TPR	TNR
{0,1}	0.9980	1.0	0.9961	{8,9}	0.9980	1.0	0.9960
{2,3}	0.9980	1.0	0.9959	{10,11}	0.9980	1.0	0.9961
{4,5}	0.9981	1.0	0.9962	{12,13}	0.9981	1.0	0.9961
{6,7}	0.9979	1.0	0.9960	{14,15}	0.9981	1.0	0.9962

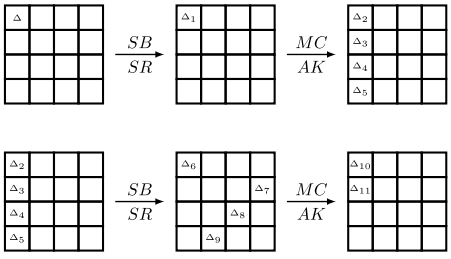

Fig. 4 The differential propagation of 2-round AES.

3.2 2-Round Distinguisher Using Partial Ciphertext

Given the challenges associated with training an effective 3-round distinguisher, our efforts have shifted towards developing a 2-round differential-neural distinguisher. As illustrated in Table 4, utilizing only one byte of ciphertext proves insufficient for training an effective distinguisher. However, as demonstrated in Table 5, employing two bytes of ciphertext enables the training of a 2-round differential-neural distinguisher, achieving an accuracy close to 100%. Notably, this 2-round distinguisher is tailored for key recovery attacks, and thus the last round of encryption incorporates the MC operation.

Figure 4 shows the propagation diagram for two rounds of AES, from which it can be inferred that a strong relationship exists between Δ_{10} and Δ_{11}.

$$\Delta_{10} = 02_{MC} \circ \Delta_6;\ \Delta_{11} = 01_{MC} \circ \Delta_6$$

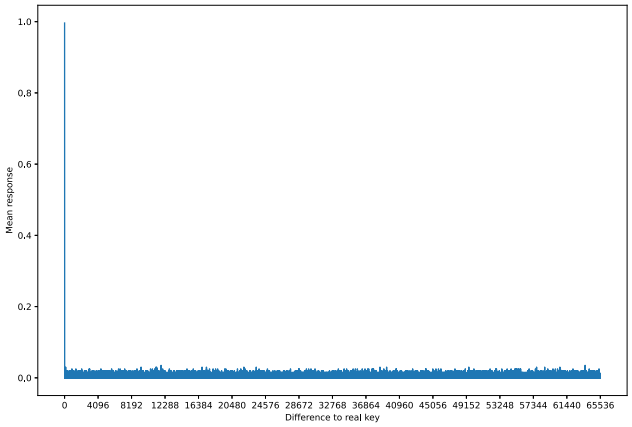

Fig. 5 Wrong key response profile of distinguisher using 0 and 1 bytes.

The distinctive characteristics encapsulated within these two bytes likely contribute to the outcomes observed in Table 5, differentiating them significantly from random numbers. In contrast, when training the differential-neural distinguisher using only one byte of ciphertext (meeting Δ_{10} or Δ_{11}), the ciphertext is subject to two SB (SubBytes) operations. This produces a very low probability of producing distinguishable differentials, rendering them indistinguishable from random numbers, as seen in Table 4.

4. Key Recovery Attack on AES-128

In light of Gohr's work on differential-neural distinguishers [1], we also examine the wrong key response profile for AES.

Wrong Key Response Profile. We generated 200 random key and plaintext pairs (P_0, P_1), encrypted them through $r+1$ rounds to produce ciphertexts (C_0, C_1). Assuming the last round partial subkey is k, we traversed all possible δ, performed single-round partial decryption on $E^{-1}_{k\oplus\delta}(C_0)$ and $E^{-1}_{k\oplus\delta}(C_1)$, and evaluated them using an r-round differential-neural distinguisher. It is essential that the subkey positions align with the ciphertext positions used by the distinguisher. The results, specifically the empirical means μ_δ, are shown in Fig. 5, where the X-axis represents the difference between the real and guessed subkey, and the Y-axis shows the average response.

From Fig. 5, it is evident that the wrong key hypothesis for AES is validated. Theoretically, decrypted data will be identified as ciphertext (a positive instance) only if the subkey guess is exact. Figure 5 specifically illustrates the wrong key response profile for indices $\{0, 1\}$ from Table 5, as the trends for other distinguishers align consistently with this depiction.

Theoretical Key Recovery Attack. According to Fig. 5, data decrypted using the correct subkey is identified as positive. Consequently, one can traverse two bytes of the subkey, i.e., 2^{16} subkeys, with just one ciphertext pair to isolate the actual subkey. To recover the complete 3-round subkey, this experiment is repeated 8 times.

Practical Key Recovery Attack. For a more robust filter, scores s from 3 ciphertext pairs are combined using the formula $\sum_{i=0}^{2} \log \frac{s_i}{1-s_i}$, effectively isolating the real subkey. Using only 2 ciphertext pairs retains 2 guessed subkeys, including the real one. Hence, the data complexity D_c for a 3-round key recovery attack is $3 \times 2 = 2^{2.585}$; the time complexity T_c is $2^{16} \times D_c \times 8 = 2^{21.585}$.

It is feasible to prepend a high-probability classical differential to the differential-neural distinguisher to extend the number of rounds in the key recovery attack. However, this adjustment correspondingly increases the complexity.

5. Conclusions

This paper has demonstrated the utility of differential-neural distinguishers in enhancing key recovery attacks of AES. Our investigations confirmed that minimal ciphertext data, such as two bytes, enables distinguishers to achieve near-perfect accuracy. The validation of the wrong key hypothesis through the wrong key response profile has been pivotal in optimizing theoretical and practical key recovery strategies. These findings illustrate the potential of differential-neural distinguishers to refine cryptographic security assessments and guide future research directions.

References

[1] A. Gohr, "Improving attacks on round-reduced speck32/64 using deep learning," CRYPTO 2019, Lecture Notes in Computer Science, vol.11693, pp.150–179, Springer, 2019.

[2] A. Benamira, D. Gérault, T. Peyrin, and Q.Q. Tan, "A deeper look at machine learning-based cryptanalysis," EUROCRYPT 2021, Lecture Notes in Computer Science, vol.12696, pp.805–835, Springer, 2021.

[3] Z. Bao, J. Guo, M. Liu, L. Ma, and Y. Tu, "Conditional differential-neural cryptanalysis," IACR Cryptol. ePrint Arch., p.719, 2021.

[4] Z. Bao, J. Lu, Y. Yao, and L. Zhang, "More insight on deep learning-aided cryptanalysis," International Conference on the Theory and Application of Cryptology and Information Security, pp.436–467, Springer, 2023.

[5] Y. Chen, Z. Bao, Y. Shen, and H. Yu, "A deep learning aided key recovery framework for large-state block ciphers," IACR Cryptol. ePrint Arch., p.1659, 2022.

[6] J. Daemen and V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard, Information Security and Cryptography, Springer, 2002.