Deep Learning-Based Neural Distinguisher for Format-Preserving Encryption Schemes FF1 and FF3

https://youtu.be/64pXYE_g8f0

한성대학교
HANSUNG UNIVERSITY

CryptoCraft LAB

- Introduce
- Background
- ModelOne / ModelMul
- Hyperparameter of Model One and ModelMul
- Result

Introduce

Article

Deep Learning-Based Neural Distinguisher for Format-Preserving Encryption Schemes FF1 and FF3

Dukyoung Kim [1], HyunJi Kim [1], Kyungbae Jang [1], Seyoung Yoon [1], and Hwajeong Seo [1,*]

[1] Division of IT Convergence Engineering, Hansung University, Seoul 02876, Korea
[*] Correspondence: hwajeong@hansung.ac.kr; Tel.: +82-760-8033

Abstract: Distinguishing data that satisfies the differential characteristic from random data is called a distinguisher attack. At CRYPTO'19, Gohr presents the first deep learning-based distinguisher for round-reduced SPECK. Building upon Gohr's work, various works have been conducted. Among many other works, inspired by Baksi et al.'s work presented at DATE'21, we propose the first neural distinguisher using single and multiple differences on Format-Preserving Encryption (FPE) schemes FF1 and FF3. We harnessed the differential characteristics used in FF1 and FF3 classical distinguishers. They used SKINNY as the inner encryption algorithm for FF3. On the other hand, we employ the standard FF1 and FF3 implementations with AES encryption (which may be more robust). This work utilizes the differentials employed in FF1 and FF3 classical distinguishers, as presented in Dunkelman et al.'s paper. In short, when using a single 0x0F (resp. 0x08) differential, we achieve the highest accuracy of **0.85** (resp. **0.98**) for FF1 (resp. FF3) in the 10-round (resp. 8-round) number domain. In the lowercase domain, due to an increased number of plaintext and ciphertext combinations, we can distinguish with the highest accuracy of **0.52** (resp. **0.55**) for FF1 (resp. FF3) in a maximum of 2 rounds. Furthermore, we present an advanced neural distinguisher designed with multiple differentials for FF1 and FF3. With this sophisticated model, we still demonstrate valid accuracy in guessing the input difference used for encryption.

Keywords: Deep learning; Distinguisher; Differential characteristic; Format preserving encryption.

Background(Format Preserving Encryption)

2. Prerequisites

2.1. Format Preserving Encryption

When applying block ciphers to database encryption, it often leads to changes in the data type or length, necessitating database structure engineering. This issue becomes particularly critical when encrypting sensitive data such as credit card numbers.

However, Format-Preserving Encryption (FPE) [13] is a method that preserves the plaintext structure even after encryption, unlike block ciphers. As a result, there is no need for additional storage capacity to store ciphertext compared to plaintext. In this context, FPE is a cost-effective and efficient solution for integration into database systems without requiring extensive engineering efforts.

In this work, our focus is on the FPE schemes FF1 and FF3, both standardized by NIST[3]. FF1 consists of 10 rounds with the same block size and a key size of 128 bits, while FF3 comprises 8 rounds with a block size of 32 bits and a key size of 128 bits. Both FPE ciphers are designed using a Feistel architecture and incorporate encryption functions similar to AES into the inner round function[4].

Although FF1 and FF3 share some similarities, FF1 offers higher security due to its increased number of rounds and its ability to support a wider range of protected data formats compared to FF3. On the contrary, FF3 has a higher data throughput compared to FF1.

Background(Differential Characteristic)

2.2. *Differential Characteristic*

Differential cryptanalysis [1] is a representative cryptanalysis method of block ciphers. The input difference (δ) is the XOR between the plaintext pairs (P_0, P_1), and the output difference (Δ) is the XOR between the ciphertext pairs. As in Equation 1, C_0 and C_1 are the results of encrypting (E) P_0 and P_1 , respectively. The output difference (Δ) can be obtained by XORing C_0 and C_1. Here, a differential characteristic means a pair of input and output differences (δ, Δ).

In the case of an ideal cipher, when plaintext with any input difference is encrypted, the output difference should be uniform (like random). A weak cryptographic algorithm has a certain output difference corresponding to an input difference. If the probability of satisfying an output difference for an input difference is greater than the random probability, the ciphertext can be distinguished from the random. These characteristics have remained even when encryption is performed and can be inferred probabilistically.

$$
\begin{aligned}
&P_1 = P_0 \oplus \delta, \\
&C_0 = E(P_0), C_1 = E(P_1), \\
&\Delta = C_0 \oplus C_1
\end{aligned}
\tag{1}
$$

Background(Neural distinguisher for FF1 and FF3)

3. Neural distinguisher for FF1 and FF3

This section describes our neural distinguisher specifically designed for the FPE schemes (FF1 and FF3). Our neural distinguisher is based on the Baksi et al. scheme [3]. Also, our neural distinguisher for FPE schemes are based on Dunkelman et al.'s ePrint'20 paper [16]. They determined that the differential characteristic of FPE shemes. Further more, our implementation is categorized into two types based on the utilized input differences, namely, *ModelOne* and *ModelMul*.

The *ModelOne* is a binary model capable of distinguishing cipher data with a single input difference from random data, while the *ModelMul* is designed to distinguish multiple input differences. Details about both models are described in Sections 3.1 and 3.2. In addition, we perform the hyper-parameter optimization for both models.

ModelOne(Single Input Difference)

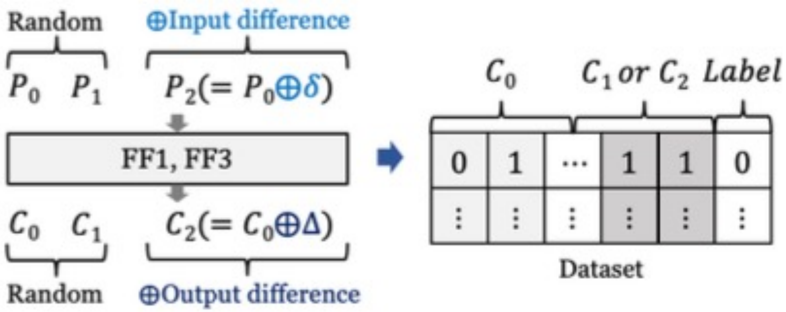


Figure 1. Dataset with one input difference.

Algorithm 1 *ModelOne*: Training procedure

Training Data $TD \leftarrow [\ ]$ ▷ Empty state
for i from 0 to $n-1$ **do**
  Choose random plaintext P_0 and P_1
  $P_2 \leftarrow P_0 \oplus \delta$
  Ciphertexts C_0, C_1, and $C_2 \leftarrow FPE_{enc}$ (P_0, P_1, and P_2) ▷ Generate ciphertexts
  $TD_i \leftarrow$ Assign labels 0 to $(C_0||C_1)$ and 1 to $(C_0||C_2)$
end for

Train model DL with TD
$a \leftarrow$ Output of DL ▷ a is training accuracy
if $a > \frac{1}{2}$ **then**
  Continue the training procedure
else ▷ $a = \frac{1}{2}$
  Abort DL
end if

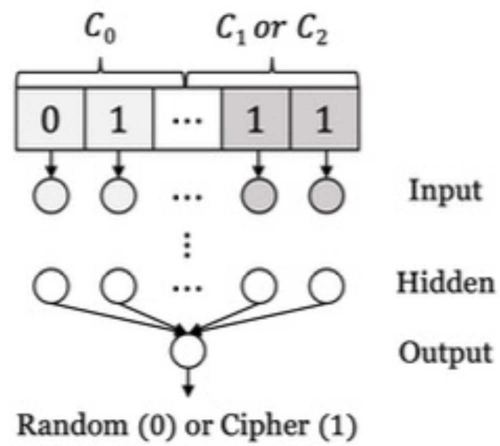
싱글 모델의 데이터 셋 생성과정으로 위의 그림1은 데이터 셋 생성과정이다.
평문 P0과 P1을 생성하여 P0에 입력 차분을 XOR하여 P2를 생성한다. 다음으로 P0, P1, P2를 암호화하여 C0, C1, C2를 생성한다. 여기서 C0와 C1은 차분 특성을 만족하지 않지만, C0와 C2는 차분 특성을 만족한다.

ModelOne(Training procedure)

3.1.2. Architecture and Training

ModelOne receives a concatenated random data ($C_0||C_1$) or cipher data ($C_0||C_2$) and classifies it into random (label 0) or cipher (label 1). Each bit of the ciphertext pair in the dataset is assigned to each neuron of the input layer. Then, the output of the input layer passes through the hidden layer. In the output layer, a final value between 0 and 1 is calculated by applying a sigmoid activation function. Then, Thenoss of the final value and the actual value (0 or 1) is calculated. Figure 2 shows the process of *ModelOne* using single input difference.

If training to distinguish input data is performed correctly, our model can work as a neural distinguisher for FF1 and FF3. To work as a valid distinguisher, it must achieve an accuracy greater than $\frac{1}{2}$, which is a random probability.



훈련 방법으로 연접한 C0와 C1는 레이블 0으로 C0와 C2는 1레이블로 분류한다.

ModelMul(Multiple Input Differences)

3.2. ModelMul: Multiple Input Differences

3.2.1. Dataset

Similar to *ModelOne*, a random plaintext P_0 is generated. Then, plaintext pairs that satisfy multiple input differences are generated. That is, P_0 is XORed with δ_n (different input difference) to obtain plaintext P_n. Lastly, each plaintext P_n (with different input differences) is encrypted to generate the ciphertext C_n. In short, *ModelMul* takes multiple ciphertexts with different input differences as a training data set.

$C_0||C_n$ is labeled as class $n-1$ since C_n is the ciphertext obtained by encrypting the plaintext with n different input differences, respectively (e.g., C_3 corresponds to Δ_3). In the distinguisher that uses multiple input differences, the number domain (0, to 9) and the lowercase domain (a to z) are also used in the FF1 and FF3 encryption process. As in *ModelOne*, we adopt the input difference 0x0$||K$ (K is a hexadecimal number ranging from 0x0 to 0xF). Figure 3 shows the generation process of the dataset using multiple input differences.

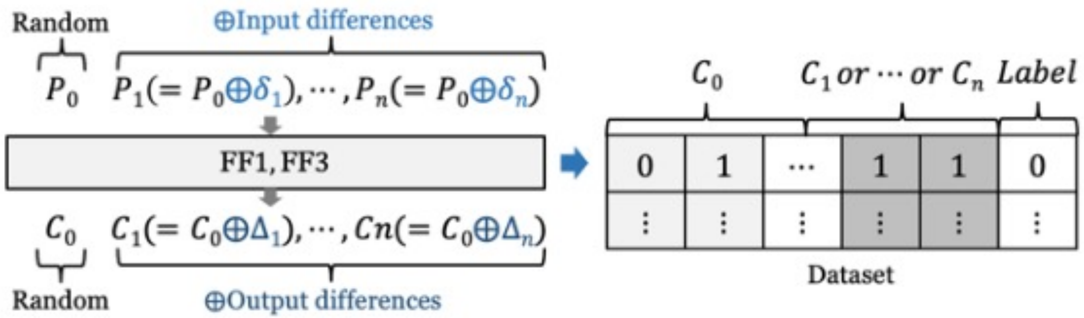


Figure 3. Dataset with multiple input differences.

ModelMul(Training procedure)

Algorithm 2 *ModelMul*: Training procedure

```
1:  Training Data TD ← [ ]                                  ▷ Empty state
2:  Choose random plaintext P                               ▷ Step 2
3:  Ciphertext C ← FPE_enc(P)          ▷ FPE_enc means FF1 or FF3 encryption
4:  for i from 0 to n − 1 do
5:      P_i ← P ⊕ δ_i
6:      C_i ← FPE_enc(P_i)
7:      Append TD with (C_i ⊕ C, i)                ▷ C_i ⊕ C is from class i
8:  end for
9:  Repeat from Step 2
10: Train DL model with TD
11: a ← Output of trained DL model                   ▷ a is training accuracy
12: if a > 1/n then
13:     Continue the training procedure
14: else                                                     ▷ a = 1/n
15:     Abort DL model
16: end if
```

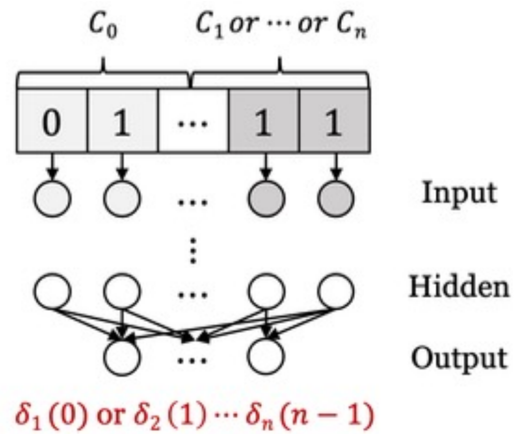


Figure 4. System diagram of *ModelMul*.

멀티모델의 데이터셋 생성과정으로 싱글 모델과 유사하게 평문 P0가 생성되며 이를 입력차분과 XOR하여 평문 Pn을 얻는다. 마지막으로 평문 Pn을 암호화하여 암호문 Cn을 생성한다.

Hyperparameter of Model One and ModelMul

Table 1. Hyperparameters of *ModelOne* and *ModelMul*.

Model	*ModelOne*	*ModelMul*
Schemes	FF1/ FF3	FF1/ FF3
Epochs	20 / 15	20 / 15
Loss function	Binary cross-entropy	Categorical cross-entropy
Optimizer	Adam (0.001 to 0.0001, learning rate decay)	
Activation function	ReLu (hidden)	
	Softmax (output)	Sigomid (output)
Batch size	32	
Hidden layers	5 / 4 hidden layers (with 64 / 128 units)	
Parameters	173,956 / 74,497	173,956 / 75,787

FF1에서는 20 FF3에서는 15 에포크 설정 싱글 모델에서는 랜덤과 차분 2가지만을 구별하기 때문에 이진분류를 사용하였으나 멀티 모델은 출력차분을 만족하는 여러 쌍의 암호문을 분류하기 때문에 다중 클래스 분류를 사용

Result

Table 2. Result of FF1 *ModelOne* according to input difference.

0x	Number (10-round)				Lowercase (2-round)			
	Training	Validation	Test	Reliability	Training	Validation	Test	Reliability
01	0.732	0.741	0.733	0.233	0.500	0.500	0.500	0.000
02	0.741	0.752	0.743	0.243	0.510	0.512	0.510	0.010
03	0.711	0.712	0.711	0.211	0.522	0.520	0.522	0.022
04	0.751	0.752	0.752	0.252	0.511	0.512	0.510	0.010
05	0.752	0.751	0.752	0.252	0.511	0.512	0.511	0.011
06	0.751	0.752	0.752	0.252	0.511	0.512	0.511	0.011
07	0.751	0.751	0.752	0.252	0.511	0.511	0.511	0.011
08	0.801	0.802	0.802	0.302	0.511	0.511	0.511	0.011
09	0.841	0.842	0.841	0.341	0.522	0.521	0.522	0.022
0A	0.842	0.841	0.841	0.341	0.500	0.510	0.510	0.010
0B	0.822	0.821	0.822	0.322	0.511	0.511	0.511	0.011
0C	0.855	0.854	0.855	0.355	0.500	0.500	0.500	0.000
0D	0.788	0.788	0.788	0.288	0.511	0.511	0.511	0.011
0E	0.811	0.812	0.811	0.311	0.522	0.521	0.522	0.022
0F	0.855	0.854	0.855	0.355	0.522	0.522	0.522	0.022

Table 3. Result of FF3 *ModelOne* according to input difference.

0x	Number (8-round)				Lowercase (2-round)			
	Training	Validation	Test	Reliability	Training	Validation	Test	Reliability
01	0.629	0.624	0.623	0.123	0.545	0.544	0.543	0.043
02	0.829	0.825	0.825	0.325	0.552	0.548	0.545	0.045
03	0.783	0.769	0.771	0.271	0.52	0.514	0.513	0.013
04	0.761	0.756	0.757	0.257	0.523	0.52	0.517	0.017
05	0.773	0.752	0.747	0.247	0.539	0.538	0.537	0.037
06	0.758	0.748	0.75	0.25	0.523	0.519	0.523	0.023
07	0.756	0.739	0.74	0.24	0.532	0.529	0.529	0.029
08	0.987	0.976	0.977	0.477	0.556	0.554	0.554	0.054
09	0.962	0.942	0.941	0.441	0.547	0.543	0.549	0.049
0A	0.969	0.953	0.951	0.451	0.538	0.534	0.532	0.032
0B	0.97	0.965	0.966	0.466	0.53	0.526	0.522	0.022
0C	0.97	0.959	0.959	0.459	0.538	0.536	0.539	0.039
0D	0.968	0.965	0.966	0.466	0.532	0.524	0.518	0.018
0E	0.964	0.963	0.963	0.463	0.549	0.549	0.551	0.051
0F	0.965	0.939	0.941	0.441	0.528	0.524	0.524	0.024

FF1 숫자 도메인에서 0C차분과 0F차분에서 0.855로 가장 높은 정확도
소문자 도메인에서는 03,08,0E,0F에서 0.522로 가장 높은 정확도를 보임

FF3는 숫자, 소문자 도메인에서 모두 08차분일때 0.977과 0.554로 가장 높은 정확도를 보임

Result

Table 4. Details of the input difference dataset.

Dataset	Data size	Input difference pair	Valid accuracy
I1	$2^{18.6097}$ per class	01, 08	>0.500
I2		01, 02, 08	>0.333
I3		01~03, 08	>0.250
I4		01~04, 08	>0.200
I5		01~05, 08	>0.166
I6		01~06, 08	>0.142
I7		01~08	>0.125
I8		01~09	>0.111
I9		01~0A	>0.100
I10		01~0B	>0.090
I11		01~0C	>0.083
I12		01~0D	>0.076
I13		01~0E	>0.071
I14		01~0F	>0.066

멀티모델의 경우 정확도가 차분의 개수에 따라 정해진다.
예) I2는 차분이 3개일때 0.333 이상의 정확도가 달성되어야 함

Result

Table 5. Result of FF1 *ModelMul* according to input differences.

Dataset	Number (8-round)				Lowercase (2-round)			
	Training	Validation	Test	Reliability	Training	Validation	Test	Reliability
I1	0.520	0.520	0.520	0.020	0.520	0.520	0.520	0.020
I2	0.340	0.339	0.340	0.007	0.360	0.360	0.360	0.207
I3	0.260	0.260	0.260	0.010	0.270	0.270	0.270	0.020
I4	0.210	0.210	0.210	0.010	0.200	0.200	0.200	0.010
I5	0.170	0.170	0.170	0.004	0.180	0.180	0.180	0.004
I6	0.150	0.150	0.150	0.008	0.150	0.150	0.150	0.008
I7	0.130	0.130	0.130	0.005	0.130	0.130	0.130	0.005
I8	0.120	0.120	0.120	0.009	0.120	0.120	0.120	0.009
I9	0.120	0.110	0.120	0.020	0.100	0.100	0.110	0.010
I10	0.100	0.100	0.100	0.010	0.100	0.100	0.100	0.010
I11	0.090	0.090	0.090	0.007	0.090	0.090	0.090	0.007
I12	0.080	0.080	0.080	0.004	0.080	0.080	0.080	0.004
I13	0.080	0.080	0.080	0.009	0.080	0.080	0.080	0.009
I14	0.070	0.070	0.070	0.004	0.070	0.070	0.070	0.004

Table 6. Result of FF3 *ModelMul* according to input differences.

Dataset	Number (8-round)				Lowercase (2-round)			
	Training	Validation	Test	Reliability	Training	Validation	Test	Reliability
I1	1.00	1.00	1.00	0.500	0.55	0.55	0.55	0.050
I2	0.99	1.00	0.99	0.657	0.54	0.54	0.54	0.207
I3	0.72	0.72	0.72	0.470	0.38	0.37	0.37	0.120
I4	0.46	0.45	0.45	0.250	0.29	0.29	0.29	0.090
I5	0.33	0.33	0.33	0.164	0.24	0.23	0.23	0.064
I6	0.25	0.25	0.25	0.108	0.20	0.20	0.20	0.058
I7	0.22	0.22	0.22	0.095	0.17	0.17	0.17	0.045
I8	0.19	0.19	0.19	0.079	0.15	0.15	0.15	0.039
I9	0.17	0.17	0.17	0.070	0.13	0.13	0.13	0.030
I10	0.16	0.15	0.15	0.06	0.12	0.12	0.12	0.030
I11	0.14	0.14	0.14	0.057	0.11	0.11	0.11	0.027
I12	0.13	0.12	0.12	0.044	0.10	0.10	0.10	0.024
I13	0.12	0.11	0.12	0.049	0.09	0.09	0.09	0.019
I14	0.11	0.11	0.11	0.044	0.08	0.08	0.08	0.014

멀티모델에서는
FF1 숫자 도메인에서 I1차분과 I9차분에서 0.520, 0.120으로 가장 높은 정확도
소문자 도메인에서는 I2에서 0.360로 가장 높은 정확도를 보임

FF3는 숫자, 소문자 도메인에서 모두 I2차분일때 0.99과 0.55로 가장 높은 정확도를 보임

Q & A