# FF3 블록 암호에 대한 딥러닝 기반의 신경망 구별자

김덕영[1], 김현지[2], 장경배[2], 임세진[1], 오유진[1], 서화정[3]
[1]한성대학교 IT융합공학과 석사과정
[2]한성대학교 정보컴퓨터공학과 박사과정
[3]한성대학교 융합보안학과 교수
dudejrdl123@gmail.com, khj1594012@gmail.com, starj1023@gmail.com,
dlatpwls834@gamil.com, oyj0922@gmail.com, hwajeong84@gmail.com

# Deep learning-based neural distinguisher for FF3 block cipher

Duk-Young Kim[1], Hyun-Ji Kim[2], Kyung-Bae Jang[2], Se-Jin Lim[1], Yu-Jin Oh[1], Hwa-Jeong Seo[3]
[1]Dept. of Convergence Security, Hansung University
[2]Dept. of Computer Information Engineering, Hansung University
[3]Dept. of Convergence Security, Hansung University

요 약

구별자 공격은 암호 알고리즘이 특정 확률로 특정 차분 특성을 만족한다는 사실을 활용하여 랜덤 데이터들로부터 암호 데이터를 구별해내는 작업이며, 데이터에 대한 확률적인 예측을 수행하는 딥러닝 기술은 이에 대한 좋은 솔루션이 될 수 있다. 최근 딥러닝 기술이 발달함에 따라 실제로 신경망 구별자에 대한 많은 연구들이 진행되고 있지만, 형태 보존 암호인 FF3에 대한 딥러닝 기반의 구별자 공격에 대한 연구는 아직 수행되지 않았다. 본 논문에서는 형태 보존암호인 FF3에 대한 딥러닝 기반의 신경망 구별자를 최초로 제안하였다. 실험 결과, 0x08 (입력 차분)에 대해서는 숫자 도메인에서 8 라운드까지 0.98 이상의 정확도를 달성하였으며, 소문자 도메인에서는 2라운드까지 구별이 가능하였다. 향후에는 또 다른 형태 보존 암호에 대한 신경망 구별자와 더 큰 도메인 및 높은 라운드에서도 동작 가능한 FF3 신경망 구별자를 구현할 예정이다.

## 1. 서론

차분 분석이란 암호 분석기법 중 하나이며, 입력 차분에 따른 출력 차분을 분석하여 키를 유추할 수 있다면, 암호 알고리즘이 안전하지 않게 설계되었다고 볼 수 있다. 이때, 차분 공격을 위해 랜덤 데이터들로부터 차분 특성 (입/출력 차분)을 만족하는 데이터를 구별해내는 것을 구별자 공격이라고 하며, 차분 공격 시 데이터 복잡도를 감소시킬 수 있다. 최근들어 딥러닝 기술이 발전함에 따라 딥러닝 기반의 구별자에 대한 연구들이 수행되고 있다. 딥러닝 기술은 데이터에 대한 확률적 예측을 수행하기 때문에 확률적으로 존재하는 차분 특성을 갖는 데이터를 분류해내는 작업에 적합하다. 이로 인해 신경망 구별자에 관한 많은 연구들이 진행되고 있지만 형태 보존 암호인 FF3에 대한 딥러닝 기반의 구별자에 관한 연구는 아직 수행되지 않았다. 본 논문에서는 FF3에 대한 딥러닝 기반의 신경망 구별자를 최초로 제안하였다.

## 2. 관련 연구

### 2.1 형태 보존 암호 FF3 32/128 [1]

형태 보존 암호는 기존 블록암호와 달리 평문의 형태를 암호문에서 온전하게 유지하도록 하는 암호화 기술이다. 즉, 임의의 도메인에 있는 평문을 암호화할 경우, 암호문도 동일한 도메인에 속하며 같은 길이를 가진다. 이때, 도메인은 숫자, 문자 등이 될 수 있다. FF3는 형태 보존 암호 기법 중 NIST의 표준으로 지정된 암호이고 8라운드와 32-bit 및 128-bit의 블록 및 키 크기를 가진다. Feistel 구조로 설계 되어 있으며, 내부 라운드 함수로써 암호화 또는 의사 난수 함수가 사용된다.

### 2.2 딥러닝 기반의 신경망 구별자

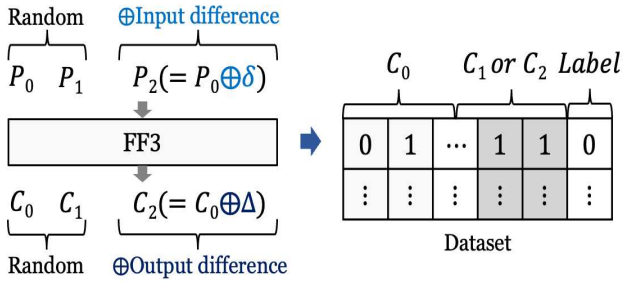
앞서 설명하였듯이 신경망 구별자는 데이터에 대한 확률적 예측을 수행하는 딥러닝 기술을 차분 특성을 활용하는 구별자 공격에 적용한 것이다. 현재 진행되고 있는 신경망 구별자에 관한 연구는 [2]로부터 파생되었고, 여러 암호와 입력 차분들을 중점으로 연구되고 있다. CRYPTO 2019에서 제안된 [2]에서는 라

운드 축소된 Speck32/64에 대해 최초의 신경망 구별자가 제안되었다. 이들의 신경망 구별자는 암호 데이터와 랜덤 데이터를 7라운드까지 성공적으로 구별하였고, 전이학습을 통해 8라운드까지 확장하였다. [3]에서는 Gohr와 다르게 다중 입력 차분과 단일 차분을 고려한 두 가지의 구별자 모델을 제시하였으며, 대상 암호는 GIMLI, ASCON, KNOT, Chaskey이다. 제안된 MLP 기반의 신경망 구별자는 8라운드 GIMLI, 3라운드 ASCON, 10/12라운드 KNOT (256/512-bit), 4라운드 Chaskey를 성공적으로 구별해냈다. 이외에도 Speck을 중점으로, 다양한 암호 및 차분 특성을 대상으로 하는 연구들이 진행되고 있다.

## 3. FF3에 대한 신경망 구별자

### 3.1 데이터 셋

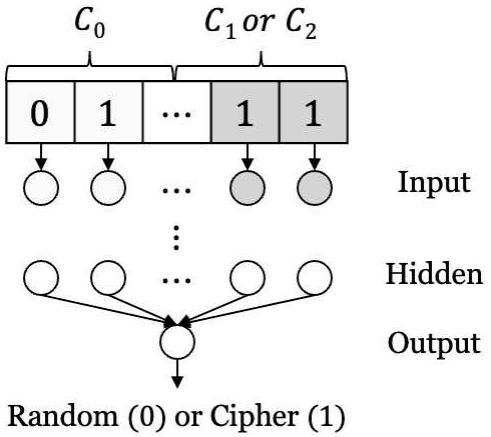
(그림 1)은 신경망 구별자의 데이터셋을 생성하는 과정을 보여준다. 우선 랜덤 평문 $P_0$, $P_1$을 생성한다. 입력 차분을 만족하는 평문 쌍을 만들어야 하므로 $P_0$에 $\delta$ (입력 차분)을 XOR 하여 평문 $P_2$를 구한다. 그 후, 각 평문 $P_0$, $P_1$, $P_2$를 암호화하여 암호문 $C_0$, $C_1$, $C_2$를 구한다. 여기서 $C_0$와 $C_1$은 차분 관계가 아닌 랜덤 평문을 암호화한 암호문이므로, 두 값을 연접한 결과는 0 (랜덤 데이터)으로 라벨링 한다. $C_0$와 $C_2$는 $\delta$ (입력 차분)을 만족하는 평문의 암호문이므로 특정 확률로 $\Delta$ (출력 차분)을 만족하는 암호 데이터이다. 따라서 $C_0$와 $C_2$를 연접한 값은 1 (Cipher)로 라벨링 한다. 암호화 과정에서 사용되는 평문 및 암호문은 숫자 (0~9) 또는 소문자 (a~z) 도메인에서 선택되고, 실제 데이터 셋에는 $C_0$, $C_1$, $C_2$의 비트 값이 저장된다.




(그림 1) 데이터 셋 생성 과정

### 3.2 모델 구성

FF3에 대한 신경망 구별자의 전체적인 구조는 (그림 2)와 같다. 랜덤 또는 차분 암호문 쌍의 각 비트는 입력 레이어의 각 뉴런에 할당된다. 이후, 히든 레이어를 거치고 출력 레이어에서 sigmoid 활성화 함수를 거쳐 0~1 사이의 값을 얻어낸 뒤 해당 값과 실제 정답 (0 또는 1)의 손실을 계산한다. 이러한 과정을 통해 입력 데이터에 대해 올바른 구별이 가능하도록 학습을 진행하면 FF3에 대한 신경망 구별자로써 동작할 수 있게 된다.




(그림 2) FF3 신경망 구별자의 구조

<표 1>은 FF3에 대한 신경망 구별자의 하이퍼파라미터를 나타낸 것이다. Epoch은 50으로 설정하였고, Dense 레이어를 사용하였다. 또한, 신경망 구별자는 차분을 갖는 데이터와 랜덤 데이터를 구별해야하므로 이진 분류를 수행한다. 따라서 손실함수로 'binary_crossentropy'를 사용하였다. 최적화 함수는 일반적으로 성능이 우수한 Adam을 사용하였다. 이때, 최적화 함수의 학습률 (Learning rate)를 조절하여 더욱 정교한 학습을 하기 위해 학습률을 0.001에서 시작하여 0.0001까지 감소하도록 하였다.

<표 1> Hyperparameters of the proposed neural distinguisher for FF3

| | |
|---|---|
| Epoch | 50 |
| Architecture | 10 hidden layers with 64 units |
| The number of parameters | 45,825 |
| Batch size | 32 |
| Activation | ReLu |
| Optimizer (Learning rate) | Adam(lr = 0.0001~0.001) |
| Loss function | binary_crossentropy |

## 4. 실험 및 성능 평가

본 실험은 Ubuntu 20.04.5 LTS와 Tesla T4 (GPU) 12GB RAM를 지원하는 클라우드 컴퓨팅 플랫폼 Google colab에서 수행되었다. 프로그래밍 환경으로는 tensorflow 2.12.0 및 Python 3.9.16를 사용하였다.

### 4.1 데이터 셋

본 실험에서는 <표 2>와 같이 데이터의 도메인을 숫자 (0~9)와 소문자 (a~z)로 나누어 사용하였다. [4]의 수식 (3)에 따르면 '0x0|$K$' ($K$는 0~F사이의 16진수)를 사용할 경우 더 높은 차분 확률을 가지므로

[4]에서 제시된 입력 차분들에 대한 실험을 진행하였다. 또한, 해당 입력 차분들은 라운드 함수의 종류와 독립적이므로 FF3의 모든 구현에 활용할 수 있다.

<표 2> Details of dataset (Tr Val, Ts : Train, Validation, Test accuracy)

| Domain | | Digits | Lowercase |
|---|---|---|---|
| Block size | | 32-bit | |
| Input difference | | 0x08, 0x01 etc.[4] | |
| The number of data | Tr | $2^{19.1206}$ | $2^{22.4425}$ |
| | Val | $2^{18.6096}$ | $2^{21.9315}$ |
| | Ts | $2^{14.8726}$ | $2^{18.1946}$ |

### 4.2 실험 결과

#### 4.2.1 Digits

<표 3>은 도메인이 Digits인 경우에 대한 라운드 및 입력 차분 별 정확도를 보여준다. 입력 차분으로 0x08을 사용하였을 경우, 최대 8 라운드까지 구별 가능하였으며 0.98의 높은 정확도를 달성하였다. 또한, [4]에서 제시된 또 다른 입력 차분들을 사용하여 실험해본 결과 0x08에 비해 낮은 정확도를 달성하였다. 0x01와 0x02의 경우, 각각 정확도가 0.3541, 0.1516만큼 감소하였다. 이러한 결과가 나올 수 있는 이유는 [4]에서 언급한 바와 같이, 0x08을 입력 차분으로 사용할 경우 예상된 차분 특성을 가지기 때문이다. 우리는 본 실험을 통해 해당 입력 차분에 대한 출력 차분을 갖는 데이터를 높은 확률로 예측할 수 있음을 확인하였다.

<표 3> Accuracies of our neural distinguisher for FF3 (Domain : Digits)

| | 0x08 | | | 0x01 | | |
|---|---|---|---|---|---|---|
| | Tr | Val | Ts | Tr | Val | Ts |
| 2/4/6 | 0.996 | 0.997 | 0.996 | 0.992 | 0.993 | 0.992 |
| | 0.982 | 0.973 | 0.973 | 0.632 | 0.626 | 0.627 |
| | 0.996 | 0.981 | 0.981 | 0.656 | 0.664 | 0.656 |
| 8 | 0.987 | 0.976 | 0.977 | 0.629 | 0.624 | 0.623 |

#### 4.2.2 Lowercase

Lowercase 도메인에서는 평문 및 암호문의 경우의 수가 $2^{26}$으로 증가함에 따라 최대 2 라운드까지 구별 가능하였다. 입력 차분 0x08에 대해 0.554의 정확도를 달성하였으며, 숫자 도메인에 비해 낮은 정확도를 보였다. 0x01 입력 차분에 대한 실험을 진행한 결과, 0x08 입력 차분을 사용한 경우보다 0.01 낮은 정확도를 달성하였으며 Digits에 비해 적은 차이를 보였다.

<표 4> Accuracies of the proposed neural distinguisher (Domain : Lowercase)

| | 0x08 | | | 0x01 | | |
|---|---|---|---|---|---|---|
| | Tr | Val | Ts | Tr | Val | Ts |
| 2 | 0.556 | 0.554 | 0.554 | 0.545 | 0.544 | 0.543 |

## 5. 결론

본 논문에서는 형태 보존암호인 FF3에 대한 최초의 딥러닝 기반의 신경망 구별자를 제안하였다. 실험 결과, 0x08 (입력 차분)에 대해서는 숫자 도메인에서 8 라운드까지 0.98 이상의 정확도를 달성하였으며, 소문자 도메인에서는 2라운드까지 구별이 가능하였다. 향후에는 또 다른 형태 보존 암호에 대한 신경망 구별자와 더 큰 도메인 및 높은 라운드에서도 동작 가능한 FF3 신경망 구별자를 구현할 예정이다.

## 6. Acknowledgements

This work was partly supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2018-0-00264, Research on Blockchain Security Technology for IoT Services, 25%) and this work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-00540, Development of Fast Design and Implementation of Cryptographic Algorithms based on GPU/ASIC, 25%) and this work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00627, Development of Lightweight BIoT technology for Highly Constrained Devices, 50%).

## 참고문헌

[1] Dworkin, Morris. "Recommendation for block cipher modes of operation: methods for format-preserving encryption." NIST Special Publication 800 (2016): 38G.

[2] Gohr, Aron. "Improving attacks on round-reduced speck32/64 using deep learning," Annual International Cryptology Conference. Springer, Cham, 2019.

[3] Baksi, Anubhab. "Machine learning-assisted differential distinguishers for lightweight ciphers," Classical and Physical Security of Symmetric Key Cryptographic Algorithms. Springer, Singapore, 141-162, 2022.

[4] Dunkelman, Orr, et al. "Cryptanalysis of Feistel-based format-preserving encryption." Cryptology ePrint Archive, 2020.