형태보존암호 FF1에 대한 딥러닝 기반의 신경망 구별자

김덕영*, 김현지*, 장경배*, 윤세영**, 서화정***

*한성대학교 (대학원생), **한성대학교 (학부생), ***한성대학교 (교수)

Deep Learning-Based Neural Distinguisher for Format-Preserving Encryption Scheme FF1

Duk-Young Kim*, Hyun-Ji Kim*, Kyung-Bae Jang*, Se-Young Yoon*, Hwa-Jeong Seo*

*Hansung University (Graduate student)
*Hansung University (Professor)

요 약

차분 분석은 암호 분석기법 중 하나이며, 차분 공격을 위해 랜덤 데이터들로부터 차분 특성 (입/출력 차분)을 만족하는 데이터를 구별해 내는 것을 구별자 공격이라 한다. Neural distinguisher는 구별자에 딥러닝을 적용한 것이다. 본 논문에서는 형태보존암호인 FF1을 위한 단일 및 다중 차분을 사용한 최초의 신경 구별자를 제안하였다. 단일 차분으로 $0F$를 사용할 때, 숫자 및 소문자 도메인에서 차분 데이터 구별에 성공하였다 (정확도는 각각 0.85 및 0.52). 다중 차분에서는 $01{\sim}0F$ 까지의 입력 차분들로 구성된 숫자 데이터 셋에 대해 숫자 및 소문자 도메인에서 모두 유효한 정확도를 달성하였다.

I. 서론

차분 분석이란 암호 분석기법 중 하나로 입력 차분에 따른 출력 차분을 분석하여 키를 유추할 수 있다면, 암호 알고리즘이 안전하지 않게 설계되었음을 의미한다. 이때, 차분 공격을 위해 무작위 데이터들로부터 차분 특성 (입/출력 차분)을 만족하는 데이터를 구별해낼 수 있다면, 차분 분석의 복잡도가 줄어든다. 이러한 기법을 구별자 공격이라 한다. 최근 딥러닝 기술이 발달하면서 딥러닝 기반의 구별자에 대한 연구들이 활발히 수행되고 있다. 하지만 아직까지 형태보존암호인 FF1에 대한 딥러닝 기반의 구별자에 관한 연구는 수행되지 않았다. 본 논문에서는 FF1에 대한 딥러닝 기반의 신경망 구별자를 최초로 제안하였다.

II. 관련 연구

1.1 형태 보존 암호 FF1 32/128 [1]

형태 보존 암호는 기존 블록암호와 달리 평문의 형태와 길이를 암호문에서 온전하게 유지하도록 하는 암호화 기술이다. 또한, FF1은 형태 보존 암호 기법중 NIST의 표준으로 지정된 암호이며 10라운드와 32-bit 및 128-bit의 블록 및 키 크기를 가지며 Feistel 구조로 설계되어있다. 또한, 내부 라운드 함수로서 암호화 또는 의사 난수 함수가 사용된다. 또 다른 형태 보존 암호인 FF3-1에 비하여 더 높은 라운드 수와 넓은 범위의 길이를 지원하고, 길이 조정에서의 유연성이 존재한다. 이러한 특성으로 인해 형태가 일정한 민감 정보 (주민등록번호, 신용카드 번호)를 암호화하는 경우 사용된다. 이는 기존 블록 암호와 달리 평문의 형태 및 길이를 그대로 유지하며 암호화 할 수 있으므로 데이터베이스의 메모리 사용량 관점에서 효율적이며, 활발히 사용되고 있다.

1.2 딥러닝 기반의 신경망 구별자

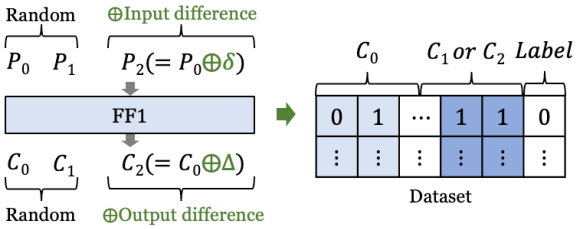
딥러닝 기술은 데이터에 대한 확률적 예측을 수행하기 적합하다. 딥러닝 기반의 신경망 구별자는 이러한 특성을 활용하여 기존의 구별자 공격에 적용한 것이며, 여러 연구들이 진행되고 있다[2,3].

III. 형태보존암호 FF1에 대한 신경망 구별자

3.1 단일 차분 모델 (*ModelOne*)

3.1.1. 데이터 셋

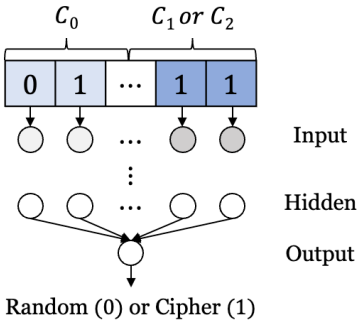
(그림 1)은 신경망 구별자의 데이터셋을 생성하는 과정이다. 먼저, 임의의 랜덤 평문 P_0, P_1을 생성 후 δ (단일 입력 차분)을 만족하는 평문 쌍을 생성하기 위해 P_0에 δ을 XOR 하여 평문 P_2을 구한다. 그 후, 각 평문 P_0, P_1, P_2을 암호화하여 암호문 C_0, C_1, C_2을 구한다. 이때 C_0와 C_1은 차분 관계가 없는 랜덤 평문을 암호화한 결과이므로, 두 값을 연접한 결과를 0 (랜덤 데이터 쌍)으로 라벨링한다. 반면 C_0와 C_2는 입력차분을 만족하는 평문의 암호문으로 특정 확률로 출력 차분을 만족하는 암호 데이터이다. 따라서, C_0와 C_2를 연접한 값은 1 (암호 데이터 쌍)로 라벨링한다. 암호화 과정에서 사용되는 평문 및 암호문은 숫자 (0~9) 또는 소문자 (a~z) 도메인에서 선택되고, 실제 데이터 셋에는 C_0, C_1, C_2의 비트 값이 저장된다.



(그림 1) 단일 차분 데이터 셋 생성 과정

3.1.2 *ModelOne* 구성

(그림 2)는 3.1절의 데이터 셋을 활용한 *ModelOne*의 시스템 구성도를 보여준다. 랜덤 또는 차분 쌍의 각 비트는 입력 레이어의 각 뉴런에 할당된다. 그 후, 히든 레이어를 거치고 출력 레이어에서 sigmoid 활성화 함수를 거쳐 0~1 사이의 값을 도출한 뒤, 해당 값과 실제 정답 (0 또는 1)의 손실을 계산한다. 이러한 과정을 통하여 입력 데이터에 올바른 구별이 가능하도록 학습이 진행되면 FF1에 대한 신경망 구별자로 동작할 수 있게 된다. <표 1>은 실제 모델 구현에 사용된 하이퍼파라미터이며, 여러 번의 실험을 통해 최적화하였다.



(그림 2) *ModelOne* 구조

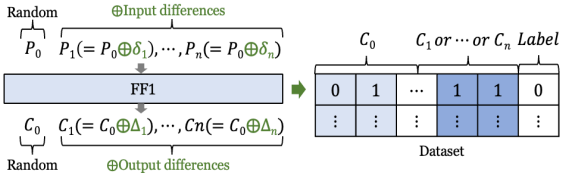
<표 1> Hyperparameters of *ModelOne*.

Epoch	20
Architecture	5 hidden layers with 64 units
The number of parameters	173,956
Batch size	32
Activation	ReLu (Hidden), Sigmoid (Output)
Optimizer (learning rate)	Adam (lr = 0.0001~0.001)
Loss function	binary_cross-entropy

3.2 다중 차분 모델 (*ModelMul*)

3.2.1 데이터 셋

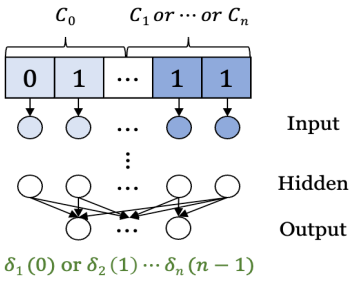
(그림 3)은 다중 차분 신경망 구별자의 데이터 셋을 생성하는 과정으로 먼저, 평문 P_0를 생성한다. 입력 차분들을 만족하는 평문 쌍들을 만들어야 하므로 P_0에 입력차분들을 XOR하여 평문P_n을 구한다. 그 후, 각 평문 P_n을 암호화하여 암호문 C_n을 구한다. 여기서 C_n들은 각기 다른 입력 차분을 갖는 랜덤 평문을 암호화한 암호문이다. 따라서 $C_0 || C_n$은 $n-1$개의 클래스로 라벨링한다. 다중 입력 차분을 사용하는 구별자에서도 암호화 과정에서는 숫자 도메인과 소문자 도메인을 사용하였으며 입력 차분으로 0x0$\|\|K$ (K는 0에서 F 사이의 16진수)를 사용하였다[4].



(그림 3) 다중 차분 데이터 셋 생성 과정

3.2.2 *ModelMul* 구성

(그림 4)는 다중 차분 모델의 시스템 구성도를 보여준다. *ModelMul*은 차분 특성을 만족하는 암호문 쌍을 입력 받아 사용된 입력 차분으로 분류한다. 즉, *ModelMul*은 입력 받은 암호 데이터에 사용된 입력 차분을 구별해낼 수 있는 것이다. 학습 과정은 단일 차분 모델과 같이 데이터 셋을 구성하는 암호문 쌍의 각 비트는 입력 레이어의 각 뉴런에 할당되고, 히든 레이어를 통과한다. 출력 레이어에서는 softmax 활성화 함수를 적용함으로써 여러 개의 클래스 (입력 차분들)에 대한 확률 값이 도출된다. 이후 손실을 계산하여 올바르게 분류할 수 있도록 모델의 파라미터들이 갱신된다. 이러한 과정을 통해 *ModelMul*은 암호문 데이터들로부터 다중 입력 차분들을 구별하여 어떠한 차분 특성이 만족되는지 알아낼 수 있게 된다. 앞서 *ModelOne*이 랜덤과 입력 차분 하나만을 분류할 수 있었다면, 다중 차분 모델은 여러 개의 차분 특성을 만족하는 데이터들에 대한 구별자로서 작동한다. <표 2>는 *ModelMul*에 사용된 최적의 하이퍼파라미터이다.



(그림 4) *ModelMul* 구성

<표 2> Hyperparameters of *ModelMul*.

Epoch	20
Architecture	5 hidden layers with 64 units
The number of parameters	173,956
Batch size	32
Activation	ReLu(Hidden), Softmax(Output)
Optimizer (Learning rate)	Adam (lr = 0.0001~0.001)
Loss function	Categorical cross-entropy

IV. 실험 결과

4.1 *ModelOne*

<표 3>은 입력 차분에 따른 *ModelOne*의 정확도를 보여준다. *ModelOne*은 숫자 도메인에서 최대 10라운드까지 데이터를 구분할 수 있으며 $0F$ 차분에서 0.85의 가장 높은 정확도 및 신뢰도 (Reliability=Ts - 0.5 (이진 분류의 랜덤 확률))를 달성하였다. 소문자 도메인에서는 평문과 암호문의 경우의 수가 증가하므로 최대 2라운드까지 데이터를 구분할 수 있으며 $03,09,0E,0F$에 대해 0.52의 정확도로 숫자 도메인에 비해 낮지만 구별자로써의 유효한 정확도를 달성하였다. 그러나 다른 차분을 사용할 경우 $0F$에 비해 두 도메인에서 상대적으로 낮은 정확도 및 신뢰도가 도출된다. 본 실험을 통해 FF1의 단일 차분 구별자에서 $0F$가 숫자와 소문자에서 공통적으로 가장 좋은 차분임을 알 수 있다.

<표 3> Result of *ModelOne* according to input difference (Tr, Val, Ts : Accuracy, Rel : Reliability).

	Number (10-round)				Lowercase (2-round)			
	Tr	*Val*	*Ts*	*Rel*	*Tr*	*Val*	*Ts*	*Rel*
01	0.73	0.74	0.73	0.23	0.50	0.50	0.50	0.00
02	0.74	0.75	0.74	0.24	0.51	0.51	0.51	0.01
03	0.71	0.71	0.71	0.21	0.52	0.51	0.52	0.02
04	0.75	0.75	0.75	0.25	0.51	0.51	0.51	0.01
05	0.75	0.75	0.75	0.25	0.51	0.51	0.51	0.01
06	0.75	0.75	0.75	0.25	0.51	0.51	0.51	0.01
07	0.75	0.75	0.75	0.25	0.51	0.51	0.51	0.01
08	0.80	0.80	0.80	0.30	0.51	0.50	0.51	0.01
09	0.84	0.83	0.84	0.34	0.52	0.52	0.52	0.02
0A	0.84	0.84	0.84	0.34	0.50	0.50	0.50	0.00
0B	0.82	0.82	0.82	0.32	0.51	0.51	0.51	0.01
0C	0.85	0.84	0.85	0.35	0.5	0.5	0.5	0.00
0D	0.78	0.78	0.78	0.28	0.51	0.51	0.51	0.01
0E	0.81	0.81	0.81	0.31	0.52	0.52	0.52	0.02
0F	0.85	0.85	0.85	0.35	0.52	0.52	0.52	0.02

<표 4>는 *ModelMul*에 대한 데이터 셋의 세부사항이다. 각 데이터 셋은 사용된 차분 쌍에 따라 설정되며 각 클래스에 대해 10만 개의 데이터를 사용하였다. 위의 실험에서 가장 좋은 차분인 $0F$를 고정 차분으로 설정하고, 01~$0E$ 까지의 차분들을 추가하면서 데이터 셋을 생성한다. 또한, 유효 정확도 (Valid accuracy)는 사용된 차분의 수에 따라 결정된다. n개의 차분을 사용하는 경우 $\frac{1}{n}$보다 높은 정확도를 달성해야 한다.

<표 4> Details of the input difference dataset.

Dataset	Data size	Input difference pair	Valid accuracy
I1	$2^{16.6341}$ per class	01, 0F	> 0.500
I2		01, 02, 0F	> 0.333
I3		01~03, 0F	> 0.250
I4		01~04, 0F	> 0.200
I5		01~05, 0F	> 0.166
I6		01~06, 0F	> 0.142
I7		01~07, 0F	> 0.125
I8		01~08, 0F	> 0.111
I9		01~09, 0F	> 0.100
I10		01~0A, 0F	> 0.090
I11		01~0B, 0F	> 0.083
I12		01~0C, 0F	> 0.076
I13		01~0D, 0F	> 0.071
I14		01~0F	> 0.066

4.2 *ModelMul*

<표 5>는 *ModelMul*의 결과를 보여준다. I1 ~ I14 중 숫자 영역에서는 I1과 I9, 소문자 영역에서는 I2가 가장 높은 정확도를 보였다. 또한, 차분의 수가 증가함에 따라 정확도가 감소하였다. 이는 구별해야 할 차분이 많을수록 작업이 복잡해지기 때문에 발생하는 당연한 현상이다. 그러나 결과를 살펴보면, 더 많은 차분을 사용했음에도 불구하고 더 높은 신뢰도를 갖는 경우가 있다. 이는 사용되는 차분 쌍에 따라 구별자의 성능이 달라질 수 있음을 의미한다. 신뢰도에 따른 차분 조합에 관한 실험은 추후 연구로 남겨둔다. 결론적으로 *ModelMul*은 숫자와 소문자 도메인 모두에서 유효한 정확도를 달성하였으므로, FF1에 대해 유효한 구별자로 사용할 수 있다.

<표 5> Result of *ModelMul* according to input difference.

	Number (10-round)				Lowercase (2-round)			
	Tr	Val	Ts	Rel	Tr	Val	Ts	Rel
I1	0.52	0.52	0.52	0.02	0.52	0.52	0.52	0.02
I2	0.34	0.33	0.34	0.007	0.36	0.36	0.36	0.027
I3	0.26	0.26	0.26	0.01	0.27	0.27	0.27	0.02
I4	0.21	0.21	0.21	0.01	0.20	0.20	0.20	0.01
I5	0.17	0.17	0.17	0.004	0.18	0.18	0.18	0.004
I6	0.15	0.15	0.15	0.008	0.15	0.15	0.15	0.008
I7	0.13	0.13	0.13	0.005	0.13	0.13	0.13	0.005
I8	0.12	0.12	0.12	0.009	0.12	0.12	0.12	0.009
I9	0.12	0.11	0.12	0.02	0.10	0.10	0.11	0.010
I10	0.10	0.10	0.10	0.01	0.10	0.10	0.10	0.010
I11	0.09	0.09	0.09	0.007	0.09	0.09	0.09	0.007
I12	0.08	0.08	0.08	0.004	0.08	0.08	0.08	0.004
I13	0.08	0.08	0.08	0.009	0.08	0.08	0.08	0.009
I14	0.07	0.07	0.07	0.004	0.07	0.07	0.07	0.004

V. 결론

본 논문에서 FF1에 대한 최초의 딥러닝 신경망 구별자를 제안하였다. 우리는 단일 차분 모델인 *ModelOne*과 다중 차분 모델인 *ModelMul*에 대한 실험을 진행하였다. *ModelOne*에서는 $0F$ 차분을 사용하였을 때 숫자 및 소문자 도메인 모두에서 가장 높은 정확도를 보였다. *ModelMul*에서는 모든 경우의 정확도가 유효 정확도를 초과하며 I2에서 가장 높은 정확도를 달성하였다. 본 실험을 통해 *ModelOne*에서는 공통적으로 0F 차분의 정확도 및 신뢰도가 높고, *ModelMul*에서는 사용되는 차분 쌍에 따라 구별자의 성능이 달라질 수 있음을 확인하였다. 향후에는 차분 조합의 신뢰도에 따른 구별자 성능을 통해 좋은 차분 얻기 위한 연구를 진행할 예정이다.

VI. Acknowledgement

This work was 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%) and this work was 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, 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] O. Dunkelman, A. Kumar, E. Lambooij, and S. K. Sanadhya, “Cryptanalysis of feistel-based format-preserving encryption,” Cryptology ePrint Archive, 2020. 5, 8, 9, 10