creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

석사학위논문

형태보존암호 FF1 및 FF3를 위한 딥러닝 기반 신경망 구별자

HANSUNG UNIVERSITY

2025년

한 성 대 학 교 대 학 원

융 합 보 안 학 과

융 합 보 안 전 공

김 덕 영

HANSUNG
UNIVERSITY

석사학위논문
지도교수 서화정

형태보존암호 FF1 및 FF3를 위한 딥러닝 기반 신경망 구별자

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

HANSUNG UNIVERSITY

2024년 12월 일

한성대학교 대학원

융합보안학과

융합보안전공

김 덕 영

석 사 학 위 논 문
지도교수 서화정

형태보존암호 FF1 및 FF3를 위한 딥러닝 기반 신경망 구별자

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

위 논문을 공학 석사학위 논문으로 제출함

2024년 12월 일

한 성 대 학 교 대 학 원

융 합 보 안 학 과

융 합 보 안 전 공

김 덕 영

HANSUNG UNIVERSITY

김덕영의 공학 석사학위 논문을 인준함

2024년 12월 일

심사위원장 박 명 서 (인)

심 사 위 원 이 웅 희 (인)

심 사 위 원 서 화 정 (인)

HANSUNG UNIVERSITY

국 문 초 록

형태보존암호 FF1 및 FF3를 위한 딥러닝 기반 신경망 구별자

한 성 대 학 교 대 학 원
융 합 보 안 학 과
융 합 보 안 전 공
김 덕 영

차분 특성을 만족하는 데이터를 무작위 데이터와 구별하는 작업을 구별자 공격 (Distinguisher attack)이라고 한다. CRYPTO 2019에서 Gohr는 라운드 축소된 SPECK에 대해 최초의 딥러닝 기반 구별자를 발표했으며, 이후 이를 바탕으로 다양한 후속 연구가 이어졌다. 이 연구들을 바탕으로, 본 논문에서는 NIST (국립표준기술연구소) 표준 형태보존암호인 FF1, FF3-1에 대한 싱글 및 멀티 차분을 활용한 최초의 신경망 구별자를 제안한다. 기존 연구는 FF3-1의 내부 암호화 알고리즘으로 SKINNY를 사용한 반면, 본 연구에서는 AES 암호화를 사용하는 표준 FF1과 FF3-1 구현을 적용하고 기존 구별자에서 사용된 차분을 활용한다. 단일 0x0F (또는 0x08) 차분을 사용할 경우 FF1은 10라운드에서 0.85, FF3-1은 8라운드에서 0.98의 최고 정확도를 달성했다. 소문자 도메인에서는 평문과 암호문 조합 수가 증가함에 따라, FF1은 최대 2라운드에서 0.52, FF3-1은 최대 2라운드에서 0.55의 최고 정확도로 구별할 수 있었다. 또한, 본 논문에서 FF1과 FF3-1에 대해 다중 차분을 사용

하는 고급 신경망 구별자를 제안하였다.

【주요어】 차분 분석, AES 암호, 구별자 공격, 형태보존암호, 딥러닝

HANSUNG UNIVERSITY

목 차

HANSUNG UNIVERSITY

HANSUNG UNIVERSITY

표 목 차

그 림 목 차

SANSUNG UNIVERSITY

수 식 목 차

알 고 리 즘 목 차

HANSUNG UNIVERSITY

제 1 장 서론

제 1 절 형태보존암호에 대한 구별자 공격

차분 분석은 주요 암호 분석 기술 중 하나로 차분 특성을 분석하여 키를 예측할 수 있는 기술이며, 이를 위해 차분 특성을 만족하는 여러 암호문이 필요하다. 차분 특성을 만족하는 데이터 (입력/출력 차분)와 무작위 데이터를 구별하는 과정을 구별자 공격이라고 한다. 차분을 분류하는 방식에 따라 이진 분류 모델 (무작위 데이터와 입력 차분을 구별)과 다중 분류 모델 (여러 입력 차분을 구별)로 나눌 수 있다. 다양한 암호 알고리즘에 대한 딥러닝 기반의 구별자 공격 연구들이 진행되었다.

딥러닝은 데이터가 가지는 특징을 분석하고 그에 따른 확률적 예측을 제공 이러한 이유로 차분 특성을 활용한 딥러닝 기반 구별자에 대한 연구가 활발히 진행되고 있으나, 형태보존암호 (FPE) 방식에 대한 딥러닝 기반 구별자 연구는 아직 충분히 이루어지지 않았다.

본 논문에서는 NIST (국립표준기술연구소) 표준 형태보존암호인 FF1, FF3-1에 대한 단일 및 다중 차분을 활용한 최초의 신경망 구별자를 제안한다.

제 2 장 관련 연구

제 1 절 형태보존암호 (FPE)

형태보존암호는 NIST 표준에서 선정된 암호 알고리즘이다. 최근 개인정보보호법 시행 등으로 인해 데이터베이스 (Database, DB) 암호화의 중요성이 커졌으며, 특히 주민등록번호, 신용카드번호 등의 암호화가 주요 이슈로 대두된다. DB 암호화에 기존 암호기술을 적용할 경우 데이터의 타입이 변하거나 길이가 증가하여 DB 스키마 변경이 필요하지만 형태보존암호를 사용 할 경우 데이터의 타입과 같이 보존하는 암호화 방식이므로 DB 스키마의 변경 없이 암호화를 적용할 수 있다.

이와 같이 형태보존암호는 일반적인 블록 암호와 달리 평문의 형식과 길이를 그대로 유지한 상태로 암호화를 수행하는 기법이다. 예를 들어, 신용카드 번호 (16자리 10진수) 중 특정 6자리를 암호화해야 하는 상황에 128비트 블록 크기의 AES 블록 암호를 이용하면, 암호문의 길이는 고정적으로 128비트가 된다. 이는 약 20비트에 불과한 해당 평문 구간에 비해 암호문이 6배 이상 커지는 결과를 초래한다. 반면, 형태 보존 암호를 적용하면 평문과 동일한 길이 및 형식을 유지할 수 있어 기존 데이터베이스 시스템에 대한 구조적 변경이나 추가 용량 확보 없이 효율적인 데이터 보호가 가능하다.

NIST 표준으로 지정된 암호는 FF1, FF3-1가 있다. FF1과 FF3-1는 각각 10라운드와 8라운드로 구성되며 블록 크기와 키 크기는 각각 32비트와 128비트이다. 또한 Feistel 구조로 설계되었으며, 내부 라운드 함수로 암호화 함수 (예: AES)를 사용하며, 해당 암호화 알고리즘은 변경 될 수 있다. 위의 두 암호는 유사한 점도 있지만, FF1은 FF3-1보다 더 높은 라운드를 가짐으로서 상대적으로 더 안전하며 FF3-1은 FF1에 비하여 데이터 처리량이 더 높다는 이점이 있다.

제 2 절 차분 암호 분석(Differential Cryptanalysis)

차분 암호 분석은 블록 암호의 대표적인 암호 분석 방법 중 하나이다. 입력 차분 (δ)은 평문 쌍 (P_0, P_1) 간의 XOR이고, 출력 차분 (Δ)은 암호문 쌍 간의 XOR이다. [수식 2-1]과 같이 C_0와C_1은 각각 P_0와P_1을 암호화 (E)한 결과이다. 출력 차분 (Δ)은 C_0와C_1을 XOR하여 얻을 수 있다. 여기서 차분 특성은 입력 차분과 출력 차분의 쌍 (δ, Δ)을 의미한다. 이상적인 암호는 임의의 입력 차분에 대해 암호화된 출력 차분이 고르게 분포되어야 하지만, 취약한 암호 알고리즘은 특정 차분에 대응하는 일정한 출력 차분을 보인다. 입력 차분에 따른 출력 차분의 발생 확률이 랜덤 확률보다 높다면, 그 암호문은 랜덤과 구별이 가능해진다. 이러한 특성은 암호화가 진행되더라도 유지되며, 확률적으로 분석할 수 있다.

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

[수식 2-1] 차분 암호 알고리즘

제 3 절 인공 신경망

딥러닝 네트워크는 여러 층으로 구성되며, 각 층은 다수의 뉴런으로 이루어져 있고 뉴런은 이전 층에서 전달된 가중치를 합산한 후, 활성화 함수를 통해 최종 출력을 계산한다. 이 과정은 입력 층에서 시작해 각 층을 거치며 반복된다.

네트워크는 손실 함수를 활용해 예측된 출력과 실제 레이블 간의 차이를 최소화하며 학습한다. 이 과정에서 최적화 함수를 사용하여 효과적으로 학습을 진행한다. 네트워크가 훈련되면, 훈련된 가중치를 사용하여 예측을 수행할 수 있으며 잘 훈련된 네트워크는 훈련되지 않은 데이터에 대

해서도 강력한 예측 성능을 보인다.

제 4 절 차분 암호 분석을 위한 신경망 구별자

신경망은 주어진 입력 차분에 대해 특정 출력 차분을 확률적으로 만족시킬 수 있기 때문에, 구별자 공격에 효과적인 해결책이 될 수 있다. 이와 관련된 연구는 주로 Gohr의 연구를 기반으로 진행되었으며, 대상 암호와 입력 차분에 초점을 맞추고 있다. CRYPTO'19에서 제안된 Gohr의 연구에서는 라운드 수가 축소된 SPECK32/64에 대한 최초의 신경망 구별자가 소개되었고, 이 구별자는 최대 7라운드 동안 암호화된 데이터와 무작위 데이터를 성공적으로 구별하였다. 또한, 전이 학습을 통해 분석 가능한 라운드가 최대 8라운드까지 확장되었다. Baksi et al.의 또 다른 연구에서는 다중 입력 차분과 단일 입력 차분을 고려한 두 가지 신경망 구별자 모델을 제안하여 GIMLI, ASCON, KNOT, Chaskey 암호에 적용하였다. 제안된 MLP 기반 신경망 구별자는 8라운드 GIML, 3라운드 ASSCON, 10/12 라운드 KNOT (256/512비트), 4라운드 Chaskey를 성공적으로 구별하였다.

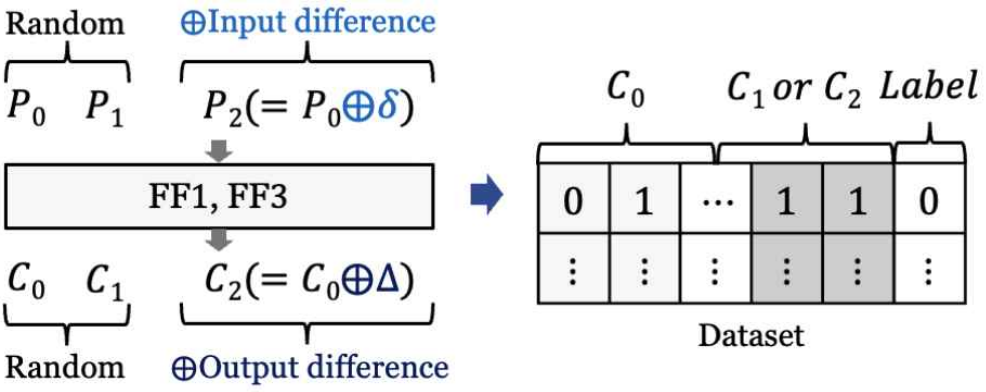
제 3 장 제안 기법

본 장에서는 FPE 방식 (FF1 및 FF3-1)을 위한 신경망 구별자를 설명한다. 신경망 구별자는 Baksi et al.의 최신 논문을 기반으로 하였으며, FPE 방식에 대한 신경망 구별자는 Dunkelman et al.의 ePrint'20 논문을 참조하였다. 본 연구에서는 FPE 방식의 차분 특성을 분석하고 이를 활용해 두가지 구현 모델 (ModelOne 및 ModelMul)을 제시한다. 두 모델은 사용된 입력 차분의 유형에 따라 구분되며, 상세한 설계과 구현 방법은 본 장 1절, 2절에서 구체적으로 다룬다.

제 1 절 ModelOne: 단일 입력 차분

1) Dataset

[그림 3-1]은 ModelOne의 단일 입력 차분을 사용하여 전체 생성 과정과 생성된 데이터 셋을 나타낸다. 먼저 랜덤 평문 P_0, P_1를 준비하여 입력 차분을 적용한 ($P_n = P_0$ XOR δ_n) P_2를 얻는다. 그 후, P_0, P_1을 암호화하여 C_0, C_1를 얻는다. 이는 사전에 정의된 차분 특성 ($C_n = C_0$ XOR Δ_n)을 만족하지 않는 일반적인 경우에 해당하며 $C_0 \| C_1$에 라벨 0을 할당한다. 반면, 특정 차분 특성 ($P_n = P_0$ XOR δ_n)을 만족하는 평문P_0, P_2는 암호화를 거쳐 C_0, C_2로 변환되며, 해당 쌍은 특정 확률에서 차분 특성 ($C_n = C_0$ XOR Δ_n)을 충족하는 관계를 가지므로 $C_0 \| C_2$에는 라벨 1을 할당한다.



[그림 3-1] Model One 차분 데이터 셋

이러한 접근은 FPE를 통해 암호화된 데이터의 원래 형식을 유지함으로 데이터 무결성, 규제 준수, 시스템간 호환성을 보장한다. 본 논문에서는 숫자 (0-9) 및 소문자 (a-z) 도메인이라는 두 가지 형식적 특성을 가진 데이터 셋을 고려하며, 암호문 쌍을 비트 단위로 연결한 구조 (예: $C_0 || C_1$ 혹은 $C_0 \| C_2$)를 활용한다. 또한, 입력 차분 데이터 셋으로는 0x0 ‖ K (K는 0x0~0xF 범위의 16진수)를 적용하였는데, 이는 Dunkelman의 방정식에 근거하여 선택한 것으로 내부 함수 (예: SKINNY, SPECK, AES)와 독립적이므로 다양한 FPE 구현에 유연하게 적용할 수 있다.

2) Architecture and Traning

ModelOne은 연접된 랜덤 데이터 ($C_0 \| C_1$) 또는 암호 데이터 ($C_0 \| C_2$)를 입력으로 받아 이를 랜덤 (라벨 0)또는 암호 (라벨 1)로 분류한다. 데이터 셋의 암호문 쌍에 포함된 각 비트는 입력 계층의 각 뉴런에 할당되며, 이후 입력 계층의 출력이 은닉 계층을 통과한다. 출력 계층에서는 시그모이드 활성화 함수를 적용하여 0과 1사이의 최종 값을 계산한다. 이 최종 값과 실제 값 (0 또는 1)간의 손실을 계산한다.

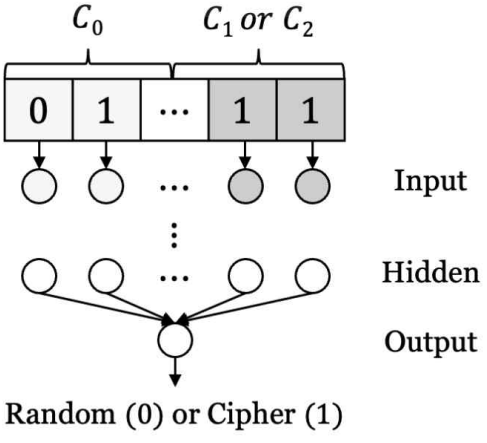
[그림 3-2] 및 [알고리즘 3-1]은 단일 차분을 사용하는 ModelOne의 전체 프로세스를 보여주며 입력 데이터를 구별하기 위한 학습이 올바르게 수행될 경우, 해당 모델은 FF1 및 FF3-1에 대한 신경망 구별자로 작동

할 수 있다. 이때, 단일 입력 차분에 대한 유효한 구별자로 작동하기 위해서는 랜덤 예측 확률인 $\frac{1}{2}$이상의 정확도를 달성해야 한다.

Algorithm 3-1 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 $C_2 \leftarrow FPE_{enc}(P_0, P_1, \text{and} P_2)$ ▷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
    Abort DL ▷$a = \frac{1}{2}$
end if

[알고리즘 3-1] ModelOne Training procedure

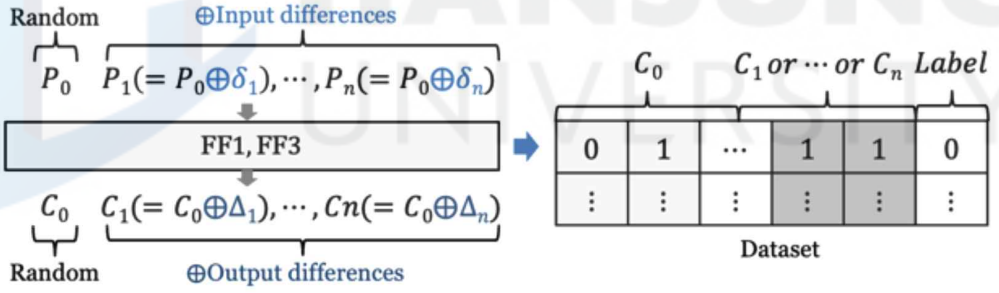


[그림 3-2] ModelOne의 시스템 다이어그램

제 2 절 ModelMul: 다중 입력 차분

1) Dataset

[그림 3-3]은 ModelMul의 다중 입력 차분을 사용하여 전체 생성 과정과 생성된 데이터 셋을 나타낸다. ModelOne과 유사하게, 임의의 평문 P_0을 생성 후 입력 차분δ_n을 적용하여 평문 P_n을 생성 ($P_n = P_0$ XOR δ_n)하고, 이를 암호화하여 암호문 C_n을 생성한다. 암호문 C_0와 C_n을 연결하여 ($C_0 \| C_n$) 학습 데이터로 사용한다. 이때, C_n이 δ_n에 해당하면 이를 클래스 n-1로 할당한다 (예: δ_3에 대응하는 C_3은 클래스 2로 분류). 다중 입력 차분을 사용하는 구별자에서도 숫자 도메인 (0-9)과 소문자 도메인 (a-z)을 사용하며, 입력 차분 0x0||K (K는 0x0-0xF)를 활용한다. ModelMul은 여러 차분 특성 (예: 0x01, 0x02, 0x08)을 학습하여 어떤 차분이 사용되었는지 구별할 수 있다.



[그림 3-3] ModelMul 차분 데이터 셋

2) Architecture and Training

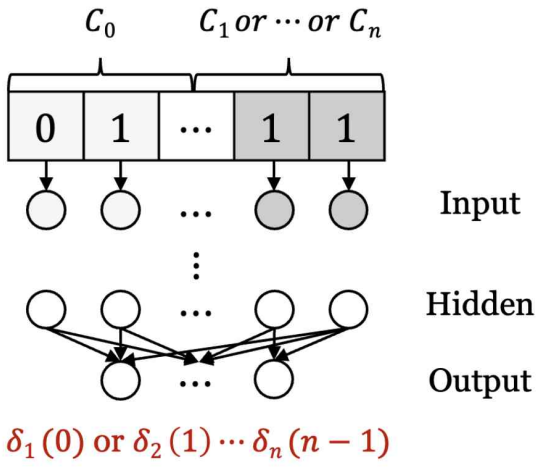
[그림 3-4] 및 [알고리즘 3-2]는 다중 입력 차분을 사용하는 ModelMul의 시스템 로직을 보여준다. 이 모델에서 입력 차분으로 δ_0, δ_1, ...,δ_{n-1} (n>2)을 선택한다. 훈련 단계에서 딥러닝 모델이 출력에서 특정 패턴(즉, 차분 특성)을 학습하여 ModelMul은 다중 입력 차분을 구별할 수 있게 된다. ModelOne이 랜덤 차분과 단일 차분만을 구별할 수 있는 반면, ModelMul은

여러 차분 특성을 만족하는 데이터를 구분하는 데 활용된다. n개의 입력 차분이 사용될 경우, 유효한 구별자로 작동하려면 랜덤 데이터 확률($\frac{1}{n}$)을 초과하는 정확도가 필요하며 학습 정확도가 $\frac{1}{n}$을 넘으면 모델은 암호문 출력에서 패턴을 찾아내고, 이를 기반으로 차분 공격이 진행된다. 반대로 학습 정확도가 $\frac{1}{n}$ 이하라면 모델은 작동을 중단한다. 즉, ModelMul은 차분 특성을 만족하는 암호문 쌍을 입력으로 받아 사용된 입력 차분에 따라 이를 분류한다. 이를 통해 암호 데이터에서 사용된 입력 차분을 효과적으로 구별할 수 있다.

Algorithm 3-2 ModelMul: Training procedure

1: Training Data $TD \leftarrow [\ ]$ ▷Empty state
2: Choose randim plaintext P ▷Step 2
3: Ciphertext $C \leftarrow FPE_{enc}(P)$ ▷FPE_{enc}means FF1 or FF3 encryption
4: for i from 0 to $n-1$ do
5: $P_i \leftarrow P \oplus \delta_i$
6: $C_i \leftarrow FPE_{enc}(P_i)$
7: Append TD with $(C_i \oplus C, i)$ ▷$C_i \oplus C$ is from class i
8: end for
9: Repeat from Step 2
10: Train DL model with TD
11: $a \leftarrow$ Output of trained DL model ▷a is training accuracy
12: if $a > \frac{1}{n}$ then
13: Continue the training procedure
14: else
15: Abort DL model ▷$a = \frac{1}{n}$
14: end if

[알고리즘 3-2] ModelMul Training procedure



[그림 3-4] ModelMul 시스템 다이어그램

제 3 절 Model One, ModelMul : 하이퍼파라미터

1) ModelOne Hyper-Parameter

[표 3-1]은 ModelOne (FF1 및 FF3-1)의 하이퍼파라미터를 제시한다. ModelOne의 경우, Epoch는 각각 20과 15로 설정되며, 모든 노드가 완전히 연결된 형태의 레이어 (Dense)가 사용된다. ModelOne은 입력 데이터를 랜덤 데이터와 암호 데이터로 구별하는 이진 분류 작업을 수행하므로, 손실 함수로 이진 Crossentropy 함수를 사용한다. 또한, 성능이 우수한 것으로 알려진 Adam 최적화 함수가 모델에 적용되며, 학습률은 학습 과정에서 동적으로 조정된다 (초기 학습률은 0.001이며, 세부 조정을 위해 학습률을 0.0001까지 감소시킨다).

2) ModelMul Hyper-Parameter

[표 3-1]은 ModelMul (FF1 및 FF3)의 하이퍼파라미터를 제시한다. ModelMul의 경우, Epoch는 각각 20과 15로 설정되며, ModelOne과 동일한 Dense레이어가 사용된다. 그러나 ModelMul은 출력 차분을 만족하는

여러 암호문 쌍을 분류하므로 다중 클래스 분류를 수행한다.

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)	Sigmoid(output)
Batch size	32	
Hidden layers	5/4 hidden layers (with 64 / 128 units)	
Parameters	173,956 / 74,497	173,956/75,787

[표 3-1] 신경망 구별자 모델의 하이퍼파라미터

HANSUNG UNIVERSITY

제 4 장 실험 및 평가

제 1 절 실험 환경

본 실험은 Ubuntu 20.04.5 LTS와 Tesla T4 (GPU) 12GB RAM을 지원하는 클라우드 컴퓨팅 플랫폼인 Google Colaboratory에서 수행되었다. 프로그래밍 환경으로는 TensorFlow 2.12.0과 Python 3.9.16이 사용되었다.

제 2 절 성능 평가 (ModelOne)

본 절에서 [표 4-1], [표 4-2]를 보면 숫자 도메인의 경우, FF1 및 FF3-1에서 입력 차분으로 0x0F / 0x08을 사용할 때, ModelOne은 10라운드 / 8라운드까지 데이터를 효과적으로 구별할 수 있으며, 각각 0.85 / 0.98의 높은 정확도를 달성했다. 다른 입력 차분을 사용할 경우, 0x0F / 0x08보다 상대적으로 낮은 정확도를 보인다.

소문자 도메인의 경우, 평문과 암호문의 경우의 수가 증가함에 따라 FF1, FF3-1에서 ModelOne은 최대 2라운드까지 데이터를 구별할 수 있으며, 0x0F / 0x08를 사용할 때 0.522 / 0.55의 정확도를 달성하는데, 이는 숫자 도메인에서보다 다소 낮은 수치이다. 입력 차분 0x03 / 0x01을 사용할 경우, 0x0F / 0x08에 비해 0.1 / 0.35 정도 낮은 정확도를 보인다. 본 실험은 특정 암호에 특정 차분일 때 데이터를 높은 확률로 예측할 수 있음을 확인하였다.

0x	Number(10Rounds)				Lowercase(2Rounds)			
	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

[표 4-1] FF1 ModelOne 결과표

0x	Number(8Rounds)				Lowercase(2Rounds)			
	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	0.519	0.519	0.523	0.023
07	0.756	0.739	0.74	0.24	0.529	0.529	0.529	0.029
08	0.987	0.976	0.977	0.477	0.554	0.554	0.554	0.054
09	0.962	0.942	0.941	0.441	0.543	0.543	0.549	0.049
0A	0.969	0.953	0.951	0.451	0.534	0.534	0.532	0.032
0B	0.97	0.965	0.966	0.466	0.526	0.526	0.522	0.022
0C	0.97	0.959	0.959	0.459	0.536	0.536	0.539	0.039
0D	0.968	0.965	0.966	0.466	0.524	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	0.524	0.54	0.524	0.024

[표 4-2] FF3-1 ModelOne 결과표

제 3 절 성능 평가 (ModelMul)

[표 4-3]은 ModelMul의 입력 차분 데이터셋에 대한 세부 정보이다. ModelMul은 입력 차분 0x0‖K를 사용한다. 각 데이터셋은 사용된 입력 차분 쌍에 따라 설정되며, 각 클래스는 $2^{18.6097}$개의 데이터로 구성된다. FF1, FF3-1에서 가장 적합한 차분으로 간주되는 0x0F, 0x08을 고정 차분으로 설정하며 서로 다른 입력 차분에 대해 데이터를 확장하면서 데이터셋을 생성하였다. 유효한 정확도는 사용된 입력 차분의 개수에 따라 정해진다. 예를 들어, 입력 차분 세 개를 사용하는 경우, 0.3333 ($=\frac{1}{3}$) 이상의 정확도를 달성해야 해당 모델이 유효하다고 할 수 있다.

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	>0.0.76
I13		01 ~ 0E	>0.071
I14		01 ~ 0F	>0.066

[표 4-3] ModelMul 입력 차분 데이터셋 세부 정보

I1 ~ I14 (입력 차분의 다양한 조합)에 대해 실험을 수행하였으며, 숫자 도메인과 소문자 도메인 모두에서 유효한 정확도를 달성하였다. 단일 입력 차분을 사용하는 ModelOne과 마찬가지로, ModelMul도 0x0‖K 차분을 구별할 수 있기 때문에 FF1과 FF3-1에 대하여 유효한 구별자로 작동한다. 아래 [표 4-4]와 [표 4-5]는 각각 FF1과 FF3-1의 입력 차분 데이터셋에 따른 ModelMul의 실험 결과를 보여준다.

Datset	Number (8 Rounds)				Lowercase (2 Rounds)			
	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	0.027
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

[표 4-4] FF1 ModelMul 결과표

HANSUNG UNIVERSITY

Datset	Number (8 Rounds)				Lowercase (2 Rounds)			
	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

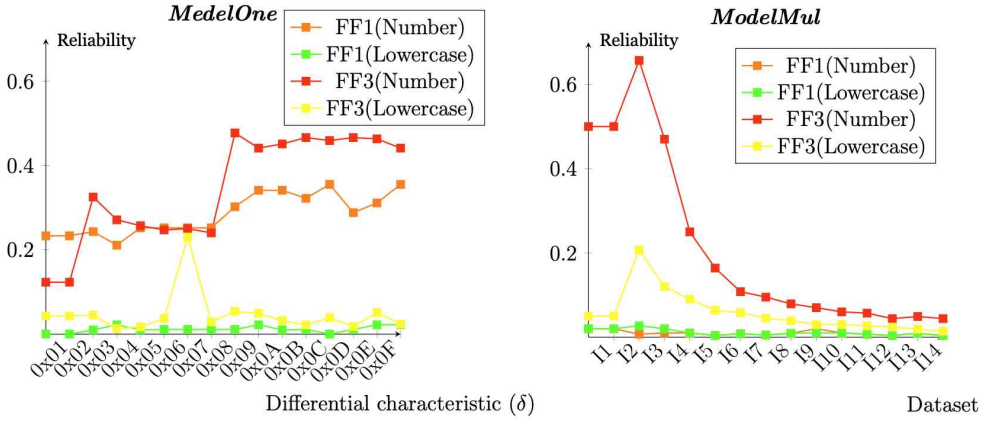
[표 4-5] FF3-1 ModelMul 결과표

I1 ~ I14중, I2은 숫자 및 소문자 (domain) 영역에서 가장 높은 신뢰도를 보였다 (신뢰도는 테스트 정확도와 검증 정확도를 의미). 본 연구 결과에 따르면, 사용되는 차분의 특성의 수가 증가할수록 신뢰도는 감소하는 경향을 보인다. 이러한 현상은 구분해야 할 차분 특성이 많을수록 해결해야 할 문제가 복잡해지기 때문인 것으로 판단된다 (일반적으로 데이터가 복잡해질수록 더 복잡한 모델 구조가 필요). 또한 입력 차분에 따라 최적의 신경망 구조가 존재할 것으로 판단된다.

제 4 절 각 차분 특성과 데이터셋에 따른 신뢰도 표

[그림 4-1]은 각 차분 특성과 데이터셋에 따른 신뢰도를 보여준다. Dunkelman의 연구에 따르면, 0x08 차분 특성은 형태보존암호 (FPE)에서 가장 우수한 차분 특성으로 밝혀졌으며, 0x01과 0x02는 상대적으로 열악한 차분 특성으로 나타난다. FF1에서 ModelOne은 입력이 0x0F일 때 두 도메인 모두에서 가장 높은 신뢰도를 보인다. FF3-1에서 ModelOne은 Dunkelman 등의 결과와 같이 0x08 차분 특성일 때 가장 높은 신뢰도를 보였다. 또한 FF3-1에서 ModelMul은 I2 데이터셋을 사용할 경우 두 도메인 모두에서 가

장 높은 신뢰도를 나타낸다.



[그림 4-1] 차분 특성과 데이터셋에 따른 신뢰도

HANSUNG UNIVERSITY

제 5 장 결론 및 향후 연구 방안

본 연구에서 FF1 및 FF3-1에 대한 최초의 신경망 구별자 모델을 제안한다. 입력 차분을 분류하는 방식에 따라 모델 유형을 이진 분류(ModelOne)과 다중 분류 (ModelMul)로 구분하였다. ModelOne의 경우, 0x0F 차분 특성을 사용할 때 10라운드에서 약 0.85의 높은 정확도를 달성하였으며, 0x08 차분 특성을 사용할 때는 8라운드에서 약 0.98에 달하는 높은 정확도를 기록하였다. 또한, 소문자 도메인에서는 최대 2라운드까지 구분 가능 하였다. ModelMul은 모든 경우에서 정확도가 유효 정확도를 상회하였으며, 특히 I2 데이터 셋을 활용했을 때 가장 높은 신뢰도를 보였다. 이와 같은 결과에 본 구현에서 기존 연구와 다른 내부 암호화 함수를 사용했음에도 불구하고 차분 특성과 그에 따른 확률이 유지되는 경향을 보여주는데, 이는 입력 차분 0x0||K가 내부 암호화 함수에 종속되지 않는다는 점을 시사한다. 따라서 본 연구에서 제안한 구별자는 FF3-1 변형에도 충분히 적용 가능할 것으로 예상된다. 향후 연구에서는 ModelMul을 더 넓은 도메인에 대해 훈련시키는 것을 목표로 한다. 모델의 범용성을 높이기 위해서는 모델 최적화뿐만 아니라 다양한 도메인 데이터를 폭넓게 활용하는 것이 중요하므로, 이를 중심으로 연구를 진행할 계획이다. 또한 이번 연구에서는 실험 환경의 제약으로 인해 대용량 데이터나 확장된 도메인 기반 데이터 활용에 어려움이 있었으나, 향후 실험 환경 개선을 통해 신뢰도 높은 검증을 수행할 예정이다.

참 고 문 헌

1. 국외문헌

Heys, H.M. A tutorial on linear and differential cryptanalysis . Cryptologia 2002, 26, 189-221. [CrossRef]

Taye, M.M. Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. Computers 2023, 12, 91. [CrossRef]

Khaloufi, H.; Abouelmehdi, K.; Beni-Hssane, A.; Rustam, F.; Jurcut, A.D.; Lee, E.; Ashraf, I. Deep learning based early detection framework for preliminary diagnosis of COVID-19 via onboard smartphone sensors. Sensors 2021, 21, 6853. [CrossRef] [PubMed]

Ammer, M.A.; Aldhyani, T.H. Deep learning algorithm to predict cryptocurrency fluctuation prices: Increasing investment awareness. Electronics 2022, 11, 2349. [CrossRef]

Lamothe-Fernández, P.; Alaminos, D.; Lamothe-López, P.; Fernández-Gámez, M.A. Deep learning methods for modeling bitcoin price. Mathematics 2020, 8, 1245. [CrossRef]

Essaid, M.; Ju, H. Deep Learning-Based Community Detection Approach on Bitcoin Network. Systems 2022, 10, 203. [CrossRef]

Zhu, S.; Li, Q.; Zhao, J.; Zhang, C.; Zhao, G.; Li, L.; Chen, Z.; Chen, Y. A Deep-Learning-Based Method for Extracting an Arbitrary Number of Individual Power Lines from UAV-Mounted Laser Scanning Point Clouds. Remote Sens. 2024, 16, 393. [CrossRef]

Lata, K.; Cenkeramaddi, L.R. Deep learning for medical image cryptography: A comprehensive review. Appl. Sci. 2023, 13, 8295. [CrossRef]

Kim, H.; Jang, K.; Lim, S.; Kang, Y.; Kim, W.; Seo, H. Quantum Neural Network Based Distinguisher on SPECK-32/64. Sensors 2023, 23, 5683. [CrossRef] [PubMed]

Kim, H.; Lim, S.; Kang, Y.; Kim, W.; Kim, D.; Yoon, S.; Seo, H. Deep-learning-based cryptanalysis of lightweight block ciphers revisited. Entropy 2023, 25, 986. [CrossRef] [PubMed]

Gohr, A. Improving attacks on round-reduced speck32/64 using deep learning. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 18-22 August 2019 ; Springer: Berlin/Heidelberg, Germany, 2019; pp. 150-179.

Baksi, A. Machine learning-assisted differential distinguishers for lightweight ciphers. In Classical and Physical Security of Symmetric Key Cryptographic Algorithms; Springer: Berlin/Heidelberg, Germany, 2022; pp. 141-162.

Baksi, A.; Breier, J.; Dasu, V.A.; Hou, X.; Kim, H.; Seo, H. New Results on Machine Learning-Based Distinguishers. IEEE Access 2023, 11, 54175-54187. [CrossRef]

Jain, A.; Kohli, V.; Mishra, G. Deep learning based differential distinguisher for lightweight cipher PRESENT. arXiv 2020, arXiv:2112.05061.

Rajan, R.; Roy, R.K.; Sen, D.; Mishra, G. Deep Learning-Based Differential Distinguisher for Lightweight Cipher GIFT-COFB. In Machine Intelligence and Smart Systems: Proceedings of MISS 2021; Springer: Berlin/Heidelberg, Germany, 2022; pp. 397-406.

Mishra, G.; Pal, S.; Krishna Murthy, S.; Prakash, I.; Kumar, A. Deep Learning-Based Differential Distinguisher for Lightweight Ciphers GIFT-64 and PRIDE. In Machine Intelligence and Smart Systems: Proceedings of MISS 2021; Springer: Berlin/Heidelberg, Germany, 2022; pp. 245-257.

Chen, Y.; Yu, H. A New Neural Distinguisher Model Considering Derived Features from Multiple Ciphertext Pairs. IACR Cryptol. ePrint Arch. 2021, 2021, 310.

Benamira, A.; Gerault, D.; Peyrin, T.; Tan, Q.Q. A deeper look at machine learning-based cryptanalysis. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, 17-21 October 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 805-835.

Hou, Z.; Ren, J.; Chen, S. Cryptanalysis of round-reduced Simon32 based on deep learning. Cryptol. ePrint Arch. 2021, 2021, 362.

Yadav, T.; Kumar, M. Differential-ml distinguisher: Machine learning based generic extension for differential cryptanalysis. In Proceedings of the International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, 6-8 October 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 191-212.

Yue, X.; Wu, W. Improved Neural Differential Distinguisher Model for Lightweight Cipher Speck. Appl. Sci. 2023, 13, 6994. [CrossRef]

Haykin, S. Neural Networks and Learning Machines, 3/E; Pearson Education India: Delhi, India, 2009.

Stallings, W. Format-preserving encryption: Overview and NIST specification. Cryptologia 2017, 41, 137-152. [CrossRef]

Jang, W.; Lee, S.Y. A format-preserving encryption FF1, FF3-1 using lightweight block ciphers LEA and, SPECK. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 30 March-3 April 2020; pp. 369-375.

Kim, H.; Kim, H.; Eum, S.; Kwon, H.; Yang, Y.; Seo, H. Parallel Implementation of PIPO and Its Application for Format Preserving Encryption. IEEE Access 2022, 10, 99963-99972. [CrossRef]

Dunkelman, O.; Kumar, A.; Lambooij, E.; Sanadhya, S.K. Cryptanalysis of Feistel-based format-preserving encryption. Cryptol. ePrint Arch. 2020, 2020, 1311.

HANSUNG UNIVERSITY

ABSTRACT

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

Kim, Duk-Young

Major in Convergence Security

Dept. of Convergence Security

The Graduate School

Hansung University

Distinguishing data that satisfy the differential characteristic from random data is called a distinguisher attack. At CRYPTO'19, Gohr presented the first deep-learning-based distinguisher for round-reduced SPECK. Building upon Gohr's work, various works have been conducted. Among many other works, we propose the first neural distinguisher using single and multiple differences for format-preserving encryption (FPE) schemes FF1 and FF3. We harnessed the differential charac- teristics 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. 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.

【Key words】 Differential Cryptanalysis, AES Encryption, Distinguisher Attack, Format-Preserving Encryption, FPE, Deep Learning

HANSUNG UNIVERSITY