

# 딥러닝을 활용한 Simon 블록 암호 분석에 대한 연구

성효은<sup>2)</sup>, 김예원<sup>2)</sup>, 강주성<sup>1,2)</sup>, 염용진<sup>1,2)\*</sup>

국민대학교 정보보안암호수학과<sup>1)</sup> / 금융정보보안학과<sup>2)</sup>

{she000, fdt150, jskang, \*salt}@kookmin.ac.kr

## A Study on Cryptanalysis of Block Cipher Simon Using Deep Learning

Hyoeun Seong<sup>1)</sup>, Yewon Kim<sup>1)</sup>, Ju-Sung Kang<sup>1,2)</sup>, Yongjin Yeom<sup>1,2)\*</sup>

Dept. of Information Security, Cryptology, and Mathematics<sup>1)</sup> /  
Financial information security<sup>2)</sup>, Kookmin Univ.

### 요 약

딥러닝은 특정 데이터 집합의 특징을 스스로 학습하여 새로운 데이터에 대한 문제를 해결하는 데 범용성을 갖는다. Gohr는 Crypto2019에서 발표한 Speck 암호분석 논문을 통하여 블록 암호 안전성 분석에 딥러닝 기술이 적용 가능함을 보여주었다. 본 논문에서는 딥러닝 기술이 Speck과 유사한 구조를 가지는 Simon 암호분석에 사용될 경우에도 그 기능이 작동 가능한지를 실험적으로 확인하고, Gohr가 제시한 Key Averaging 알고리즘을 수학적으로 분석한다.

### I. 서 론

딥러닝(deep learning)은 다층 구조의 신경망(neural network) 모델로 데이터의 명시적이지 않은 특성을 스스로 추출하고 학습함으로써 인공지능적인 문제의 해결을 가능하게 한다. 암호분석(cryptanalysis) 분야에도 딥러닝을 기반 기술로 사용하려는 시도들이 있었으며, Crypto2019에서 Gohr[1]가 딥러닝을 라운드 수를 줄인 Speck 암호분석에 활용한 논문을 발표하여 기존의 방식인 차분분석(differential cryptanalysis)보다 높은 정확도로 Speck 암호분석이 가능함을 보였다.

본 논문에서는 [1]에 제시된 Key Averaging 알고리즘을 수학적으로 심층 분석하고, Speck과 유사한 블록 암호 Simon에 대한 딥러닝 기반 분석 가능성을 살펴본다.

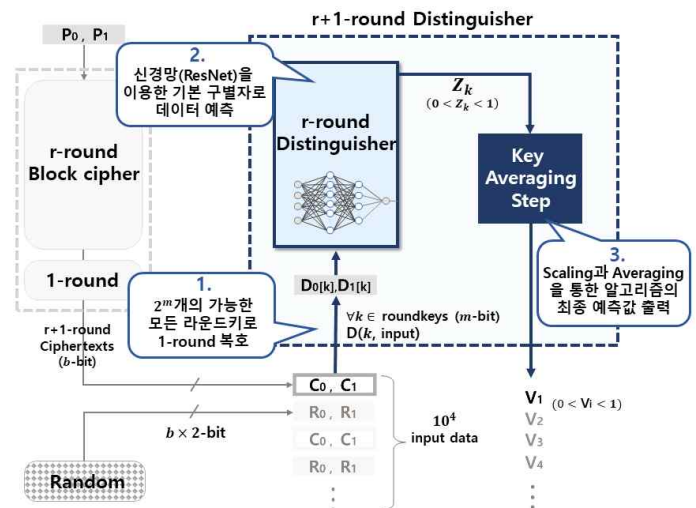
### II. Key Averaging 알고리즘

Gohr[1]는 딥러닝 기술 중 ResNet 구조의 CNN 모델을 사용하였다. 라운드 수를  $r$ 로 줄인 Speck(이하,  $r$ -라운드 Speck)으로 암호화된 데이터와 랜덤하게 생성된 데이터를 식별하는 구별자(distinguisher)로 CNN 모델을 사용한 것이다.  $r$ -라운드 구별자를 사용하여  $(r+1)$ -라운드 암호문 데이터와 랜덤 데이터의 구별 정확도를 높이기 위해 Gohr는 분석 알고리즘의 핵심 단계인 Key Averaging 알고리즘을 제시하였다.

#### 2.1 기본 구별자

기본 구별자는 CNN 모델을 사용하여 특정 평문 차분으로 생성된 암호문 쌍 데이터와 같은 길이의 랜덤 데이터를 구별한다. 모델 훈련에는  $10^7$  개, 테스트에는  $10^6$  개의 데이터가 각각 사용된다. 훈련된 구별자의 출력값  $Z \in (0, 1)$  가 0.5보다 크면 암호문 데이터로 구별하고, 그렇지 않은 경우엔 랜덤 데이터로 분류한다.

#### 2.2 Key Averaging 알고리즘이 적용된 구별자



[그림 1] Key Averaging 알고리즘 동작 과정

[그림 1]은 Key Averaging 알고리즘을 이용한 구별자로  $10^4$ 개의 입력 데이터를 암호문 데이터와 랜덤 데이터로 구별하는 과정을 설명한다. 하나의 입력 데이터를 가능한 모든  $(r+1)$ -라운드 키 후보로 복호화한 값들을 훈련된 딥러닝 모델인 기본 구별자를 통과시킨 후, Key Averaging 단계를 거친다. 최종 출력값  $V_i \in (0, 1)$  가 0.5보다 클 때, 입력 데이터를  $(r+1)$ -라운드 암호문 데이터로 분류한다.

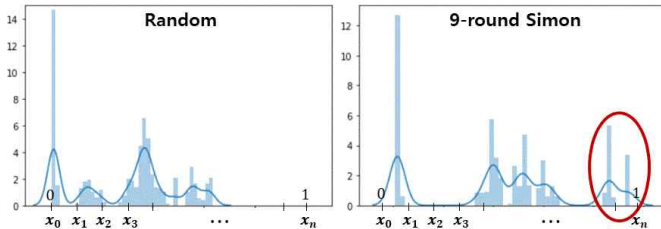
#### 2.3 Key Averaging 단계의 내부 함수 분석

Key Averaging 단계는 다음 수식과 같이  $Z_k$ 의 범위를 조정하는 (scaling) 내부 함수  $f$ 로 구별자의 성능을 향상시킬 수 있다.

$$(0, 1) \xrightarrow{f} (0, \infty) \xrightarrow{f^{-1}} (0, 1)$$

$$Z_k \xrightarrow{z'_k = \frac{Z_k}{1-Z_k}} z = \sum_{k=0}^{2^m-1} z'_k \xrightarrow{v = \frac{z}{1+z}} V$$

하나의 랜덤 데이터와 9-라운드 Simon 암호문 데이터를 각각의 입력으로 할 때,  $2^m$  개의 라운드 키 후보  $k$ 에 대한 예측값  $Z_k$ 의 분포는 [그림 2]와 같은 차이를 보인다. 이때,  $f$ 에 의해 우측 분포에 표시된 부분의 가중치(weight)를 크게 해줌으로써 두 분포의 구별 확률을 높인다.



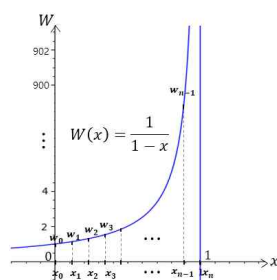
[그림 2] 신경망 예측값  $Z_k$ 의 분포의 예

Key Averaging 단계에서 가중치의 효과를 검증해 보기 위하여,  $Z_k$ 의 범위  $(0, 1)$ 을 균등하게 분할한 값들을  $x_0 = 0, x_1, x_2, \dots, x_n = 1$ 라 하자.  $x_i$ 의 가중치 값에 상관없이 전체  $Z_k$ 값이 Averaging 될 경우의 결과는 다음과 같다.

$$V = \frac{1}{2^m} \sum_{k=0}^{2^m-1} Z_k \quad (1).$$

$V_R, V_S$ 가 각각 랜덤 데이터, 암호문 데이터에 대한 알고리즘 출력값이라고 할 때, 두 값은 큰 차이가 나지 않아 전체  $(r+1)$ -라운드 구별자의 정확도가 낮아지게 된다.

한편, [그림 2]의 표시된 부분과 같이 암호문 데이터 분포에서만 나타나는 1에 가까운 값들에 큰 가중치를 두는 식 (2)로 Weighted Averaging 방법을 적용함으로써 [그림 2]와 같은 두 분포를 좀 더 효과적으로 구별하는 것이 가능할 것이다.



$$Z = \frac{1}{2^m} \sum_{k=0}^{2^m-1} W(Z_k) \cdot Z_k \quad (2).$$

실제로 [1]에서는 좌측과 같은 함수 합수를 가중치 함수  $W$ 로 사용하였다.

내부 함수를  $f(Z_k) = W(Z_k) \cdot Z_k$ 라 하면 식 (3)에서와 같이 역함수를 취해줌으로써 결과값의 범위를  $(0, 1)$ 로 환원시키는 것이 가능하다.

$$V = f^{-1} \left( \frac{1}{2^m} \sum_{k=0}^{2^m-1} f(Z_k) \right) \quad (3).$$

그 결과 암호문 데이터에 대한 알고리즘 출력값  $V_S$ 가 더 높은 값을 가져 알고리즘의 정확도가 향상된다.

[그림 2]와 같은 분포를 가지는 9-라운드 Simon 암호문 데이터에 대한 식 (1)의 우리의 실험적 결과는  $V_S = 0.37 (< 0.5)$ 로 알고리즘이 데이터를 랜덤 데이터로 오판하는 결과를 보였으나, 식 (3)의 결과는  $V_S = 0.93 (> 0.5)$ 으로 알고리즘이 데이터를 올바르게 구별하였다.

### III. 딥러닝 기반 Simon 암호분석

본 논문의 Simon 암호분석 실험에는 NVIDIA GeForce TITAN X GPU를 사용하였으며, 기본적인 소스 코드는 [3]을 참고하였다. 기본 구별자로 사용된 CNN 모델의 은닉층은 32 커널로 이루어진 convolution layer 21층과 64 unit들로 이루어진 dense layer 2층으로 구성하였다. 한 모델의 200 epochs 훈련에 약 7시간 정도가 소요되었고, 훈련된 모델을 사용한 Key Averaging 알고리즘은 1시간 이내에 수행 가능하였다.

#### 3.1 실험 과정

Simon의 각 라운드별로 먼저  $r$ -라운드 기본 구별자를 생성한 후 Key Averaging 알고리즘을 적용한  $(r+1)$ -라운드 구별자의 정확도를 측정하였다. 실험 데이터의 평균 차분은 Simon 차분분석 시 차분 특성 확률이 가장 높다고 알려진  $0x0000/0020$ 을 사용하였다[2].

#### 3.2 실험 결과

		Speck		Simon	
구별자		기본 구별자	Key Averaging 알고리즘	기본 구별자	Key Averaging 알고리즘
라운드	5	0.927	-	-	-
	6	0.787	0.796	-	-
	7	0.611	0.633	0.938	-
	8	-	-	0.746	0.816
	9	-	-	0.607	0.659
	10	-	-	0.500	0.557

[표 1] 구별자의 정확도 측정 결과

[표 1]은 라운드별 구별자의 정확도를 측정한 결과로, 정확도는 구별자 입력 데이터들 중 올바르게 구별한 데이터의 비율을 나타낸다. Speck 암호문 구별자의 경우 [1]에 기재된 정확도 범위 내에서 측정됨을 확인하였다. 또한, 해당 방법을 Simon 암호문의 구별자로 사용한 결과 Speck 암호의 5, 6, 7-라운드 구별자와 Simon 암호의 7, 8, 9-라운드 구별자가 유사한 정확도로 구별하는 것을 관찰할 수 있었다. Simon의 경우 Speck 암호보다 2라운드 더 높은 9-라운드까지 구별 가능하였다.

### IV. 결론

본 논문에서는 딥러닝 모델을 Simon 암호분석에 사용 가능함을 확인하였고 [1]에 제안된 Key Averaging 알고리즘의 내부 함수의 기능에 대해 수학적으로 분석하고 실험적으로 확인하였다. Simon 암호분석에 기존에 사용된 기법을 적용한 결과 Speck 암호분석에서의 결과보다 2라운드 더 높은 라운드까지 적용 가능하다는 결과를 얻을 수 있었다. 향후에 각 암호 알고리즘의 특성에 적합한 Key Averaging 알고리즘의 내부 함수에 대하여 데이터 분포를 효율적으로 구별하기 위한 관점의 연구가 필요할 것으로 사료된다.

### 참고 문헌

- [1] Gohr Aron, "Improving Attacks on Round-Reduced Speck32/64 Using Deep Learning," Annual International Cryptology Conference. Springer, Cham, 2019.
- [2] Ray Beaulieu, Douglas Shors, and Jason Smit. "The Simon and Speck Families of Lightweight Block Ciphers." National Security Agency, 2013.
- [3] Gohr Aron, "gohr/deep\_speck", 2020, ([https://github.com/agohr/deep\\_speck](https://github.com/agohr/deep_speck)).