Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.30, NO.3, Jun. 2020 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2020.30.3.337

딥러닝 기반 S-Box 설계정보 분석 방법 연구*

김 동 훈,[1†] 김 성 겸,[1] 홍 득 조,[2‡] 성 재 철,[3] 홍 석 희[4]
[1,4]고려대학교 (대학원생, 교수), [2]전북대학교 (교수), [3]서울시립대학교 (교수)

An Study on the Analysis of Design Criteria for S-Box Based on Deep Learning*

Dong-hoon Kim,[1†] Seonggyeom Kim,[1] Deukjo Hong,[2‡]
Jaechul Sung,[3] Seokhie Hong[4]
[1,4]Korea University (Graduate student, Professor),
[2]Chonbuk National University (Professor), [3]University of Seoul (Professor)

요 약

CRYPTO 2019에 발표된 Gohr의 연구결과는 딥러닝 기술이 암호분석에 활용될 수 있음을 보여주었다. 본 논문에서는 특정 구조를 가진 S-box를 딥러닝 기술이 식별할 수 있는지 실험한 결과를 제시한다. 이를 위해, 2가지 실험을 수행하였다. 첫 번째로는, 경량암호 설계에 주로 사용하는 *Feistel* 및 *MISTY*, *SPN*, *multiplicative inverse* 구조를 가진 S-box의 DDT 및 LAT로 학습 데이터를 구성하고 딥러닝 알고리즘으로 구조를 식별하는 실험을 수행하여 구조는 물론 라운드까지 식별할 수 있었다. 두 번째로는 *Feistel* 및 *MISTY* 구조가 특정 라운드까지 의사난수성을 보이는지에 대한 실험을 통해 이론적으로 제시된 라운드 수 보다 많은 라운드 수에서 *random*한 함수와 구분할 수 있음을 확인하였다. 일반적으로, 군사용 등 고도의 기밀성 유지를 위해 사용되는 암호들은 공격이나 해독을 근본적으로 차단하기 위해 설계정보를 공개하지 않는 것이 원칙이다. 본 논문에서 제시된 방법은 딥러닝 기술이 이처럼 공개되지 않은 설계정보를 분석하는 하나의 도구로 사용 가능하다는 것을 보여준다.

ABSTRACT

In CRYPTO 2019, Gohr presents that Deep-learning can be used for cryptanalysis. In this paper, we verify whether Deep-learning can identify the structures of S-box. To this end, we conducted two experiments. First, we use DDT and LAT of S-boxes as the learning data, whose structure is one of mainly used S-box structures including *Feistel*, *MISTY*, *SPN*, and *multiplicative inverse*. Surprisingly, our Deep-learning algorithms can identify not only the structures but also the number of used rounds. The second application verifies the pseudo-randomness of and structures by increasing the nuber of rounds in each structure. Our Deep-learning algorithms outperform the theoretical distinguisher in terms of the number of rounds. In general, the design rationale of ciphers used for high level of confidentiality, such as for military purposes, tends to be concealed in order to interfere cryptanalysis. The methods presented in this paper show that Deep-learning can be utilized as a tool for analyzing such undisclosed design rationale.

Keywords: Cryptanalysis, Deep-learning, Symmetric key, S-box structure

Received(03. 03. 2020), Modified(04. 29. 2020), Accepted(05. 04. 2020)
* 본 논문은 2019 동계 학술대회에 발표한 최우수논문을 개선 및 확장한 것임
* 본 연구는 고려대 암호기술 특화연구센터(UD170109ED)를 통한 방위사업청과 국방과학연구소의 연구비 지원으로 수행되었습니다.

† 주저자, dhkim85@korea.ac.kr
‡ 교신저자, deukjo.hong@jbnu.ac.kr (Corresponding author)

I. 서 론

암호는 Kerckhoffs 원리에 따라 설계원리가 공개되어도 안전하게 구현되어 오직 키를 통하여서만 암호 안전성을 보장받도록 설계한다. 그러나, 군이나 국가 기관에서 사용하는 암호는 해독을 방지하기 위해 설계정보를 일체 미공개 하고 있다. 이러한 암호를 분석하기 위한 키나 내부연산 복원 등을 수행하기 위해서는 암호 구조나 라운드와 같은 초기정보 획득이 중요하다.

블록암호의 구성요소 중 하나인 S-box는 치환(substitution)을 통해 입력값을 특정 출력값으로 변경시켜 비선형적 성질을 만들어주는 역할을 한다. NSA(National Security Agency)의 DES 및 Skipjack 암호의 경우, 비선형 부분이 설계된 S-box의 구체적인 설계원리를 미공개하여 해독을 원천적으로 차단하고 있으며, 학계에서는 이러한 설계원리를 밝히기 위한 연구가 진행되고 있다.

Biruykov 및 Perrin[1]은 Skipjack의 S-box 설계원리 복원을 위해 다양한 방법을 사용하였다. 그 중 Jackson Pollock's Pattern Recognition 기법은 DDT 및 LAT 등 암호분석에 사용되는 요소를 이미지화 하여 특정한 패턴을 찾고 S-box의 설계원리를 식별하는 방법이다.

예를들어 이 방법으로 3~5라운드 *Feistel*, *MISTY* 구조와 *multiplicative inverse* ($=x^{-1}$) 및 3라운드 *SPN*, 랜덤순열의 DDT를 Fig. 1.과같이 나타내면, 3라운드 *Feistel*에서는 랜덤순열과 육안으로 구별되는 대각선 및 격자형의 띠가 식별되고 4라운드에서는 격자형의 띠는 찾아보기 어려운 반면 대각선 형태의 띠는 남아있는 것이 식별된다. 또한, 5라운드가 되면 랜덤순열과 구별이 되지 않는 것을 볼 수 있다. *MISTY*구조에서도 랜덤순열과 구별할 수 있는 격자형의 띠 모양이 3라운드에서 식별 가능하고 4라운드에서는 격자의 분포가 희미해지는 것을 볼 수 있다.

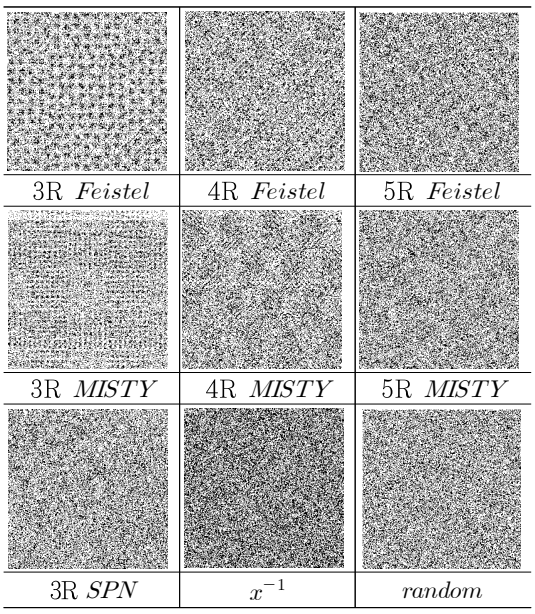

Fig. 1. DDT's Jackson Pollock's Pattern

또한, 5라운드로 확장시킬 경우에는 랜덤순열과 거의 유사한 모양이 보이는 것을 알 수 있다. 3라운드 *SPN* 구조의 경우에는 랜덤순열과 구별되는 패턴이 존재하는 듯 보이나 육안으로 명확하게 식별하기에는 어려웠으며, x^{-1}의 경우에는 랜덤순열과 육안으로는 거의 구별이 불가능하였다. 이처럼, 각 함수들은 적은 라운드에서는 육안으로 랜덤순열과 구조가 구별 가능한 특징점을 나타내고 있으나 라운드가 올라갈수록 육안으로는 식별되지 않는 것을 알 수 있다.

최근의 딥러닝 기술은 이미지나 데이터들의 특징점들을 식별하는데 효과적이다. 본 논문에서는 이러한 딥러닝 기술을 이용하여 다양한 구조의 식별 및 각 구조의 라운드 수 증가에 따른 의사난수성 분석을 시도하였다.

이를 위해, 경량암호에 많이 사용되는 *Feistel* 및 *MISTY*, *SPN*, x^{-1}구조의 확장된 S-box를 생성하고 대응하는 DDT와 LAT를 구성하였다. 다음으로는 이를 학습 데이터로 구성하고, 딥러닝 알고리즘으로 학습을 시켜서 구별해본 결과 구조뿐만 아니라 라운드 수까지도 구별하였다. 또한, 이론적으로 랜덤순열과 구별 가능함이 증명된 *Feistel* 및 *MISTY* 구조의 DDT 및 LAT를 라운드 수 별로 학습시켜 의사난수성을 확인해 본 결과 8라운드까지는 랜덤순열과 구별가능한 것을 확인하였다.

이를 통해, 본 연구결과는 딥러닝 기술을 이용하여 암호알고리즘의 공개되지 않은 설계정보를 분석하는 것이 가능하다는 것을 보여준다.

본 논문은 구성은 다음과 같다. Ⅱ장에서 연구에 사용된 확장 S-box들의 설계원리와 DDT, LAT에 대해서 소개하였으며, 딥러닝 기술을 이용한 암호분석 분야에 대해서 기술하였다. Ⅲ장에서는 딥러닝 기술을 이용하여 여러 가지 구조들에 대해서 분류하는 방법을, Ⅳ장에서는 *Feistel* 및 *MISTY* 구조에 대

해서 의사난수성 확인 방법과 실험 내용을 기술하였으며, V장에서 실험결과를 분석해보았다. 마지막 장에서는 본 논문의 결론을 도출하였다.

II. 배경지식

2.1 경량성을 고려한 확장 S-box

S-box를 설계할 때에는 x^{-1}와 같은 수학적 연산이나 *Feistel*·*MISTY*·*SPN* 등과 같이 암호에서 사용되는 구조를 사용한다. 한편, 최근 경량화된 시스템들의 등장으로 보안을 위하여 사용하는 암호 경량화에 대한 요구에 따라 다양한 경량화 전략이 반영된 암호가 제안되고 있다. 안전성 측면에서는 x^{-1} 구조의 S-box가 가장 안전한 것으로 알려져 있으나, 구현비용이 큰 단점을 지니고 있어 *Feistel*·*MISTY*·*SPN* 구조를 이용하여 4-bit 형태로부터 확장한 8-bit S-box가 경량화 전략에서 자주 활용된다.

2.1.1 Feistel(Balanced) 기반 확장 S-Box

Robin, CRYPTON S-box에서 사용된 *Feistel* 기반 확장방법은 Fig. 2에서 보는 것처럼 $2n$bit 입력값을 좌·우 $n$bit씩 분리하고 비선형 연산 S_n을 수행한 후, 라운드를 지날 때마다 좌·우 값을 교환한다. 1라운드를 지나면 $R_1 = S_1(R_0) \oplus L_0$, $L_1 = R_0$ 값을 가진다. R 라운드 *Feistel*의 구조를 갖는 S-box의 집합을 $\mathcal{F}^R$로 표현하며, 내부에서 구성된 S-box들을 $S_i (1 \le i \le R)$로 표기하고 하나의 $\mathcal{F}^R$에는 동일한 순열의 S_i를 사용한다. 본 논문에서는, $n = 4$로 설정하여 4-bit S_i를 이용해서 8-bit로 확장된 모델을 사용하였다. 각 S_i는 4-bit 랜덤 permutation을 선택하여 구성하였다.

Feistel 구조는 Luby와 Rackoff[2] 및 Patarin[3]에 의해 이론적으로 6라운드 이상에서 랜덤순열과 구별이 어려운 것으로 알려져 있다.

2.1.2 MISTY(Balanced) 기반 확장 S-Box

MISTY 구조는 *Feistel*과 동일하게 $2n$비트 입력 후, n비트씩 나누어지며 *Feistel*과 달리 좌·우를 번갈아 가면서 비선형 연산 S_n에 입력되는 구조를 가진다. 따라서 1라운드에서는 *Feistel*과 유사한 형태를 가지지만 2라운드부터는 전혀 다른 구조의 형태를 보인다. 아래에는 *Feistel*과 *MISTY*의 2라운드 값이다.

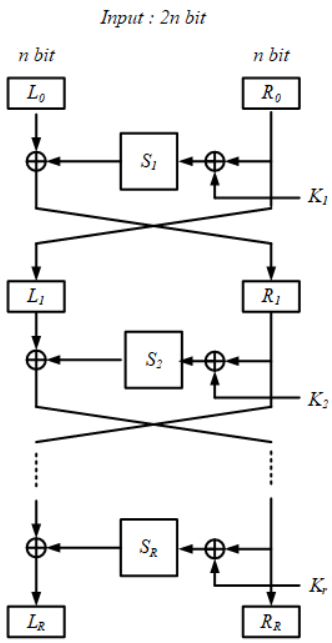

Fig. 2. Feistel Extension

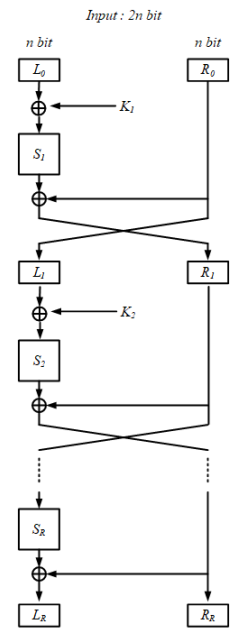

Fig. 3. MISTY Extension

$$Feistel: L_2 = S_1(R_0)\oplus L_0$$
$$R_2 = S_2(S_1(R_0)\oplus L_0)\oplus R_0$$
$$MISTY: L_2 = S_1(L_0)\oplus R_0$$
$$R_2 = S_2(R_0)\oplus S_1(L_0)\oplus R_0$$

R라운드 *MISTY*의 구조를 갖는 S-box의 집합은 $\mathcal{M}^R$로 표현하며, 내부 S-box는 *Feistel*과 동일한 S_i로 표현한다. *MISTY*구조도 Gilbert와 Minier[4]에 의해서 4라운드부터 랜덤순열과 구별이 어려운 것이 알려져 있다.

2.1.3 SPN 기반 확장 S-Box

ICEBERG, KHAZARD S-box에서 사용된 *SPN*구조는 Fig. 4에서 보는 것처럼 $2n$bit 입력값을 좌·우 $n$bit씩 분리하고 비선형 연산인 SR_i 에서 치환을, 구현 효율성을 위하여 내부 P-box에서는 bit-permutation 과정을 통해서 혼돈과(confusion) 확산(diffusion) 효과를 준다. 본 논문에SL_i, 서는 라운드 *SPN*으로 확장한 S-box의 집합은 $\mathcal{SP}^R$로 표현하며, 비선형 연산은 SL_i, SR_i $(1 \le i \le R)$로 표현하고 동일한 순열을 적용한다. 또한, bit- permutation은 라운드마다 다르게 사용한다. 또한, $n=4$로 설정하여 4-bit SL_i, SR_i을 통해 8bit로 확장된 모델을 사용하였다.

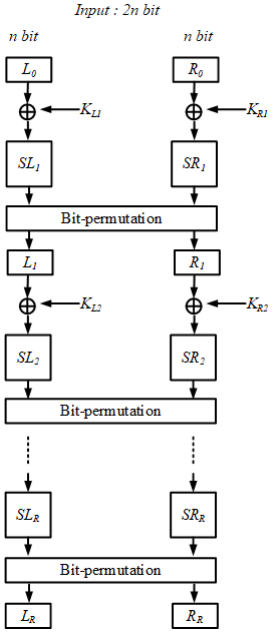

Fig. 4. SPN Extension

2.1.4 Multiplicative Inverse 기반 확장 S-Box

AES S-box에 사용된 x^{-1}구조는 $GF(2^8)$에서 입력값에 대해서 x^{-1}를 기반으로 입력값과 출력값에 Affine layer를 덧붙여 설계하였다. 이를 구현하기 위해, AES 구조에서 사용한 형태를 적용해서 모델을 구성하였다.[5] 이 구조에서는 구조 생성시마다 임의의 Affine layer A_1, A_2들을 생성하여 출력값이 매번 달라지도록 구현하였다. 본 논문에서는 x^{-1}로 확장한 S-box들의 집합을 $\mathcal{J}$로 표현한다.

$$f: GF(2^8) \rightarrow GF(2^8)$$
$$I: x \mapsto x^{256-2}$$

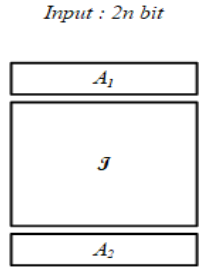

Fig. 5. Multiplicative Inverse

2.2 DDT & LAT

2.2.1 DDT(Differential Distribution Table)

블록암호의 대표적인 공격으로 알려진 차분공격[6]은 S-box의 입력쌍의 차분($\triangle P = P_1 \oplus P_2$)과 이에 대응하는 출력쌍의 차분($\triangle C = C_1 \oplus C_2$)의 관계가 특정 확률을 가지고 있는 성질을 이용하여 암호키를 추측하는 공격기법이다. 이때, 모든 입력 값들에 대한 차분과 모든 출력 값들에 대한 차분들의 갯수를 합산하여 분포를 나타낸 테이블이 DDT이다.

$s:\{0,1\}^n \rightarrow \{0,1\}^n$일 때, $2^n \times 2^n$크기의 DDT 계수 $d_{i,j}$를 구하는 방법은 아래 식과 같다.

$$d_{i,j} = \#\{ x \in \{0,1\}^n \mid s(x\oplus i)\oplus s(x) = j\}$$

한편, DDT의 계수들은 2의 배수값을 가지며 분포 특성상 2, 4, 8 등 작은 수들이 128, 256 등 큰

값보다 상대적으로 많은 비율을 차지한다. 이로 인하여, DDT를 실제 Jackson Pollock's Pattern으로 나타내어 보면 Fig. 6에서 보는 것처럼 0, 2, 4 등 낮은 수의 부분들이 검정색으로 표시되어 육안으로 식별할 수 없다. 본 논문에서는 이를 육안으로 식별하기 위해서 값이 있는 부분은 255로 표현하고 값이 없는 부분은 0으로 치환한 *DDT를 통해 확인해 보았다.

그 이유는 Fig. 1에서 언급한 것처럼 사람의 육안으로도 구별 가능한 적은 라운드의 확장된 S-box를 딥러닝 구별자가 구분할 수 있는지와, 육안으로는 구별이 어려운 많은 라운드의 확장된 S-box까지도 구별 가능한지를 알아보고 싶었기 때문이다. 이후, 딥러닝 구별자에는 실제 DDT data를 입력하여 실험하였으며, 본 논문에서는 DDT 구별자의 집합을 $\mathcal{D}_D$ 로 표현한다.

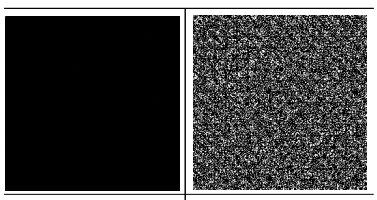

Fig. 6. Jackson Pollock's Pattern recognition

2.2.2 LAT(Linear Approximation Table)

블록암호의 또 다른 공격 중 하나인 선형공격[7]은 S-box 사이의 특정 입·출력 관계를 통해 선형근사식을 구성하고 이러한 관계를 통해 키를 추측하는 공격방법이다. 이때, 선형식을 구성하는 다양한 관계가 바이어스라고 불리는 기준점으로부터 얼마나 차이가 나는지를 계산하여 나타낸 표가 LAT이다. $2^n \times 2^n$ LAT의 계수값 $c_{i,j}$를 구하는 식은 아래와 같다.

$$c_{i,j} = \#\{x \in 0,1\}^n | x \bullet i = s(x) \bullet j\} + 2^{n-1}$$

본 논문에서는 LAT 구별자 집합을 $\mathcal{D}_L$ 로 표현한다.

2.3 딥러닝을 활용한 암호분석

딥러닝 기술은 인공지능의 한 분야로 특정 데이터 집단을 입력하면 여러 개의 procesing layer를 통하여 데이터를 학습하고 집단이 공통적으로 가지고 있는 통계적 특징을 계산하여 데이터를 분류하거나 회귀하는 기술이다. 최근 들어 GPU 기술이 발전하면서 병렬처리가 가능하게 되었고 컴퓨터 성능의 발전으로 많은 양의 데이터를 학습하고 분류할 수 있게 되면서 활발하게 연구가 진행되고 있는 분야이다. 암호분석 분야에서는 과거 머신러닝을 이용하여 암호문을 통해 암호알고리즘의 종류를 식별하는 연구들이 주로 수행되었으나 실제 높은 수준의 정확도를 보여주지 못하였다.[8][9] 또한, 딥러닝 기술이 발전된 이후에는 부채널 분야에서 다양한 방법으로 활용되고 있으며[10], CRYPTO 2019에서 Ahron Gohr가 딥러닝을 이용한 Speck 32/64을 차분공격을 수행한 논문을 발표[11]하는 등 암호분석에도 적용되고 있다.

III. 다양한 구조의 S-box 식별

우선, *Feistel*·*MISTY*·*SPN* 구조의 3~5라운드로 확장한 S-box와 x^{-1}로 확장한 S-box까지 총 10가지 case를 딥러닝 모델이 구별 가능한지 실험해 보았다.

3.1 데이터 셋 준비

실험은 8-bit $\mathcal{F}^R$, $\mathcal{M}^R$, $\mathcal{SP}^R$의 라운드 값을 3~5까지 증가시키며 DDT와 LAT를 구성하고, $\mathcal{I}$도 동일하게 DDT와 LAT를 구성하였다.

내부 4bit S-box S_i는 2가지 종류로 구성하였다. 첫째, 같은 S-box를 사용한 경우로 $\mathcal{F}_e^R$, $\mathcal{M}_e^R$, $\mathcal{SP}_e^R$으로 표현한다. 두 번째는 동일 S-box에 라운드 키를 포함한 경우로 $\mathcal{F}_k^R$, $\mathcal{M}_k^R$, $\mathcal{SP}_k^R$로 표기한다.

이를 위해 1epoch당 각 case별로 10,000개씩 100,000개의 데이터를 생성하였다. train 데이터는 전체 데이터의 70%인 70,000개를, validation 데이터는 전체 데이터의 30%인 30,000개로 구분하여 사용하였다. test data는 case별로 5,000개씩 50,000개의 데이터를 생성하여 실험하였다.

3.2 딥러닝 네트워크 구성

딥러닝 네트워크는 다중 분류에 좋은 성능을 보이고 있는 CNN(Convolution Neural Network)을 사용하였다. Fig.7에서 보는 것처럼 Conv2D를 4개 층으로 구성하였으며, 각 층마다 Maxpooling을 통해 feature를 반으로 줄여주었다. 각 layer를 통과한 이미지는 Fig.8처럼 특정 feature에 딥러닝 모델

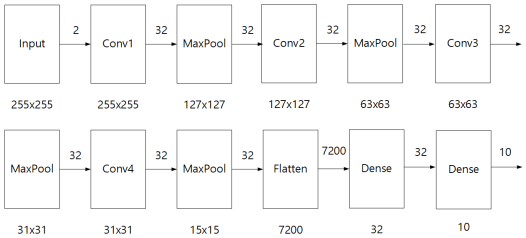

Fig. 7. Network Structure of first experiment

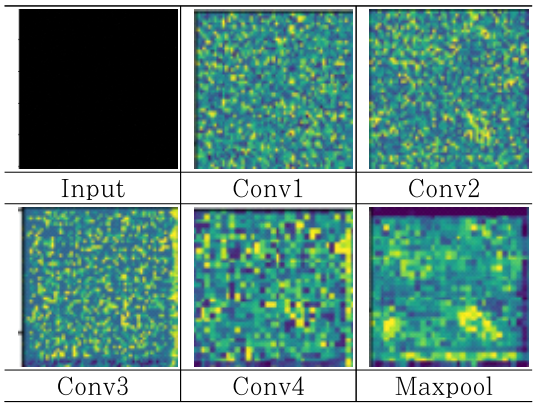

Fig. 8. Pooling images of 3R Feistel DDT

이 중점적으로 인식하는 노란색 부분이 많이 식별되게 된다. 마지막 Dense는 2개 층으로 구성하여 출력하였으며, 필터는 32, epoch는 50, batch size는 100으로 설정하였다. Convolution 계층마다 activation 함수는 relu를 사용하였으며, 마지막 Dense에서는 다중분류에서 사용되는 softmax 함수를 사용하였다. optimizer는 RMSprop를, loss function은 다중 분류에 적합한 categorical crossentropy를 사용하였다.

3.3 실험 환경

실험에는 NVIDIA GeForce GTX 1080 Ti(11GB) GPU, Intel Core i7-9700 CPU와 32GB RAM을 탑재한 PC 1대로 사용하였다. 학습 데이터를 생성하는데 약 12분이 소요되었으며, 1epoch당 학습을 수행하는데 약 9.7GB의 GPU 메모리를 사용하였으며 학습에는 약 15분이 소요되었다.

IV. *Feistel* 및 *MISTY* 구조 의사난수성 분석

이 장에서는 이론적으로 특정 라운드 이상에서 의사난수성을 만족하는 것이 증명된 *Feistel*과 *MISTY*의 라운드를 증가시키면서 확장된 S-box를 구성하고 딥러닝 모델이 식별할 수 있는지를 실험해보았다.

4.1 데이터 셋 준비

실험은 8-bit $\mathcal{F}^R$,$\mathcal{M}^R$의 라운드 값을 3~8까지 증가시키며 DDT 및 LAT를 구성하고, 내부 4bit S-box S_i는 3가지 종류로 구성하였다. 첫째, 같은 S-box를 사용한 경우로 $\mathcal{F}_e^R$, $\mathcal{M}_e^R$으로 표현한다. 두 번째는 동일 S-box에 라운드 키를 포함한 경우로 $\mathcal{F}_k^R$, $\mathcal{M}_k^R$으로 표현한다. 또한, 대조군으로는 랜덤함수를 사용하고 $\mathcal{R}$로 표기한다.

학습을 위한 입력 데이터는 $\mathcal{F}_e^R$ vs $\mathcal{R}$, $\mathcal{M}_e^R$ vs $\mathcal{R}$, $\mathcal{F}_k^R$ vs $\mathcal{R}$, $\mathcal{M}_k^R$ vs $\mathcal{R}$, 4가지 상황에 대해서 3~8 라운드 별로 총 12개의 상황을 구성하였다. 또한, DDT ($\mathcal{D}_D$), LAT($\mathcal{D}_L$)와, 두 개의 테이블을 합친 DDT & LAT($\mathcal{D}_{D\|L}$)까지 총 3개의 테이블 구별자를 통하여 36가지 상황을 실험하였다. 본 논문에서 $\mathcal{D}_{D\|L}$에 대해서도 실험한 이유는 딥러닝 모델은 back propagation을 통해 weight를 보정하는 능력이 뛰어나기 때문에 두 테이블을 합하여 하나로 구성할 경우 각 테이블의 장점을 반영하여 weight를 훨씬 더 잘 보완할 것으로 가정했기 때문이다. 이로 인해 기존 구별자보다 훨씬 더 좋은 구별자의 역할을 할 것이라 생각했다. 입력 데이터는 1epoch당 $\mathcal{F}^R$,$\mathcal{M}^R$, $\mathcal{R}$ 구조별로 1,000개씩 총 2,000개를 생성하고 딥러닝 모델을 이용하여 학습시킨다. 훈련 데이터는 입력 데이터의 50%인 1,000개를, 검증 데이터는 30%인 600개를, test 데이터는 20%인 400개를 사용하였다.

Table 1. Preprocessing input data

categorize	$\mathcal{D}_D$	$\mathcal{D}_L$	$\mathcal{D}_{D\|L}$
$\mathcal{F}_e^3$ vs $\mathcal{R}$	per 1epoch input data : 2,000(each 1,000) train data : 1,000(each 500) validation : 600(each 300) test : 400(each 200)		
$\mathcal{M}_e^3$ vs $\mathcal{R}$			
$\mathcal{F}_k^3$ vs $\mathcal{R}$			
$\mathcal{M}_k^3$ vs $\mathcal{R}$			
…			
$\mathcal{M}_k^8$ vs $\mathcal{R}$			

4.2 딥러닝 네트워크 구성

첫 번째 실험과 동일하게 CNN Conv2D를 4개 층으로 구성하였으며, 학습간 과적합이 발생하여 dropout을 각 층마다 추가하였다. 또한, 1개층마다 Maxpooling을 통해 feature의 개수를 줄였으며, 입력된 데이터는 Shuffle함수를 통해 랜덤과 *Feistel*·*MISTY* 구조를 섞어서 동일한 구조가 연속으로 학습되거나 평가되어 결과에 영향을 미치는 상황을 최소화시켰다.

Convolution 계층마다 activation 함수는 relu를 사용하였으며, 마지막 Dense에서는 sigmoid를 사용하였다. optimizer는 이진 분류에 적합한 binary crossentropy를 사용하였다.

4.3 실험환경

첫 번째 실험과 동일하게 PC 1대를 사용하였으며, 학습에는 약 3분이 소요되었다. 1개의 case당 학습을 수행하는데 약 1.2GB의 GPU 메모리를 사용하였다.

V. 실험결과

5.1 다양한 확장구조에 대한 구조식별

실험은 10개 case를 분류하는 모델이 1개의 case를 임의로 선택하였을 때 정답을 맞추는 정확도를 10%라고 할 때, 딥러닝 모델이 학습된 데이터를 바탕으로 test data가 case를 구분하지를 확인해 보았다. 학습한 결과 과적합 없이 검증 정확도와 손실률이 거의 비슷한 수준을 보이며 학습되었다.

【$\mathcal{F}_e^R$, $\mathcal{M}_e^R$, $\mathcal{SP}_e^R$, $\mathcal{I}$: 동일한 S-box 사용】

$\mathcal{F}_e^R$,$\mathcal{M}_e^R$,$\mathcal{SP}_e^R$ 의 경우에는 3라운드에서 Jackson Pollock's pattern이 구조별로 상이한 모양을 나타내고 있어 쉽게 구분할 수 있을 것으로 판단하였다. 그러나, 5라운드부터는 거의 랜덤순열과 동일한 모양을 보이고 있어 육안으로는 구분이 불가능 하였다.

반면, 딥 러닝 모델을 통해서 구분해본 결과 Fig. 9.에서 보는 것처럼 학습단계에서 8epoch부터 80%이상으로 구분을 하였으며, 테스트 셋을 통해서 구별해본 결과 $\mathcal{D}_D$는 90.4%, $\mathcal{D}_L$을 사용했을 경우에는 82%의 높은 정확도로 구조를 구분하였다. 이 실험

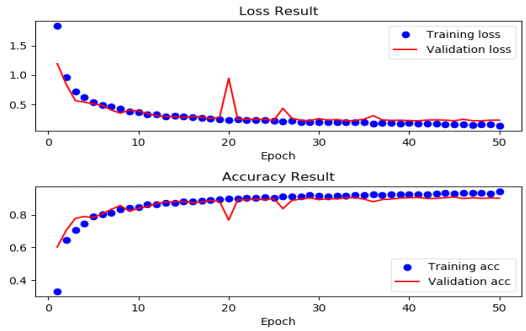

Fig. 9. Result of identify structure using equivalent S-box(DDT)

을 통해서 각 구조에 대한 정보가 DDT·LAT에 포함되어 테이블 계수들 사이의 관계나 연산에서 딥러닝이 구별 가능한 특징점으로 나타난다는 것을 확인하였다.

특히, 딥러닝 구별자는 동일한 구조에 다른 라운드를 가진 경우에 대해서도 서로 구별하였다. 일반적으로는 라운드를 지날수록 암호의 확산 효과가 커지면서 암호를 구별하기 어려운 효과를 주는 것으로 알려져 있다. 이 실험을 통하여 라운드에 대한 정보가 DDT·LAT에 포함되어 딥러닝 구별자가 식별할 수 있는 특징점으로 나타난다는 것을 확인하였다.

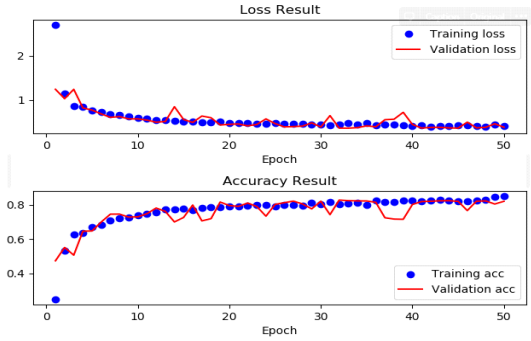

Fig. 10. Result of identify structure using equivalent S-box(LAT)

【$\mathcal{F}_k^R$, $\mathcal{M}_k^R$, $\mathcal{SP}_k^R$, $\mathcal{I}$: 동일 S-box에 라운드키 적용】

안전한 S-box를 설계하기 위해서는 내부 S-box를 라운드별로 다르게 사용하는 것이 가장 이상적이나, 암호 설계시 가용성 문제로 동일한 S-box에 라운드 키를 적용하여 설계한다. 이를 적용하여 $\mathcal{F}_k^R$, $\mathcal{M}_k^R$, $\mathcal{SP}_k^R$는 $s^r \oplus round\ key$를 통해 라운드 별로

다른 S-box를 적용한 효과를 가진다고 가정하였다. 키를 적용할 경우 확산효과가 커지면서 구별이 어려울 것으로 예상하였으나, 실험결과에서는 $\mathcal{D}_D$ 가 86.1%, $\mathcal{D}_L$은 80.9%의 높은 정확도로 구조를 구분하였다.

두 실험을 분석해볼 때 $\mathcal{D}_D$가 $\mathcal{D}_L$보다 약 10% 정도 좋은 정확도를 보이는 것을 보아 DDT가 LAT보다 구조식별에 더 용이한 것을 알 수 있었다.

이 실험결과를 바탕으로 실제 암호들의 S-box DDT를 구한 후 훈련된 $\mathcal{D}_D$ 를 바탕으로 예측해보았다. 표 3에서 보는 것처럼 x^{-1}의 경우에는 다른 구조와 쉽게 구별하는 것을 알 수 있었으며, *Feistel*, *SPN* 의 경우에는 구조는 식별하였으나 라운드 정보는 식별하지 못하였다. 한편, 일부 구조에 대해서는 라운드 정보를 식별하지 못했는데 이는 S-box를 선택할 때 암호학적 성질이 좋은 S-box를 선택하기 때문이라고 추측된다.

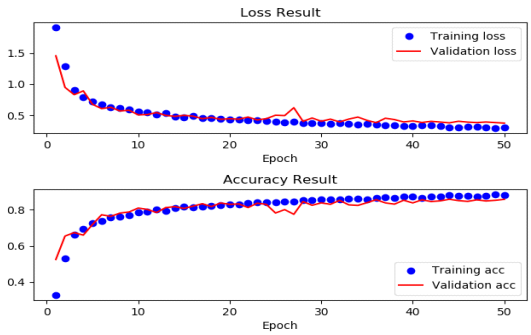

Fig. 11. Result of identify structure using different S-box(DDT)

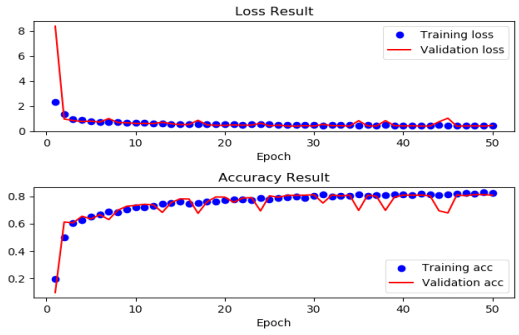

Fig. 12. Result of identify structure using different S-box(LAT)

Table 2. Identification accuracy rate in extension S-box structures

(accuracy(%))

Category	$\mathcal{D}_D$	$\mathcal{D}_L$
$\mathcal{F}_e^R$ vs $\mathcal{M}_e^R$ vs $\mathcal{SP}_e^R$ vs $\mathcal{I}$	90.4	82
$\mathcal{F}_k^R$ vs $\mathcal{M}_k^R$ vs $\mathcal{SP}_k^R$ vs $\mathcal{I}$	86.1	80.9

Table 3. Predictation of real S-box structures

Cipher	S-box Structure	Predict
AES	x^{-1}	x^{-1}
Clefia	x^{-1}	x^{-1}
CRYPTON	3r *Feistel*	4r *Feistel*
CS_cipher	3r *Feistel*	3r *Feistel*
ICEBERG	3r *SPN*	4r *SPN*
KHAZARD	3r *SPN*	4r *SPN*

5.2 *Feistel*·*MISTY* 임의 함수와의 구분

두 번째 실험으로는 이론적으로 특정 라운드까지 랜덤순열과 구분이 증명된 *Feistel*·*MISTY*구조에 대해서 딥러닝 구별자가 각 구조의DDT와 LAT를 몇 라운드까지 의사난수와 구별하는지에 대해서 실험해보았다. 실험결과는 $\mathcal{F}^R$ vs $\mathcal{R}$, $\mathcal{M}^R$vs $\mathcal{R}$ 중 정확히 한 가지 구조를 구분할 확률이 50% 일 때, Test data의 정답을 딥러닝 모델이 맞추는 정확도로 구별 성공을 판단하였다. 각 상황에 대한 실험결과는 아래와 같다.

【$\mathcal{F}_e^R$, $\mathcal{M}_e^R$: 동일한 S-box 사용】

$\mathcal{F}_e^R$, $\mathcal{M}_e^R$의 경우 Jackson Pollock's Pattern이 5라운드 이상부터 랜덤순열과 눈으로 식별 불가한 것에 비해 딥러닝 모델은 $\mathcal{F}_e^R$의 경우 8라운드까지 높은 확률로 각 구조를 식별하였으며, $\mathcal{M}_e^R$의 경우에는 6라운드까지 구조를 식별하였다. 이는 첫 번째 실험과 동일하게 DDT, LAT 계수들 사이의 관계나 구조의 라운드가 반복되면서 발생하는 특정 연산에서 딥러닝이 랜덤한 함수와 구별 가능한 특징점이 존재한다는 것을 확인하였다.

【$\mathcal{F}_k^R$, $\mathcal{M}_k^R$: 동일한 S-box ⊕ 라운드 key】

라운드 키를 적용해본 결과 $\mathcal{F}_k^R$은 6라운드까지 구별을 하였으며, $\mathcal{M}_k^R$은 $\mathcal{D}_D$ 의 경우 5라운드에서,

$\mathcal{D}_L$ 및 $\mathcal{D}_{D||L}$의 경우에는 4라운드까지 구별을 하는 것을 알 수 있었다. 이를 볼 때, 첫 번째 실험과 동일하게 차분 특성을 지닌 $\mathcal{D}_D$가 선형 특성을 나타내는 $\mathcal{D}_L$보다 구조를 식별하는데 좋은 성능을 보인다는 것을 알 수 있었다. 또한, $\mathcal{D}_{D||L}$의 경우 우리의 예상과는 달리 정확도가 낮은 $\mathcal{D}_L$와 유사한 모습을 보였다. 이것으로 볼 때, $\mathcal{D}_{D||L}$ 모델은 정확도가 상대적으로 낮은 $\mathcal{D}_L$의 영향을 많이 받는 것을 알 수 있었다. 또한, $\mathcal{F}^R$ vs $\mathcal{R}$ 과 $\mathcal{M}^R$ vs $\mathcal{R}$의 경우를 분석해보면 상대적으로 $\mathcal{F}^R$ vs $\mathcal{R}$의 경우가 좋은 정확도를 보였다. 이는, *MISTY*설계간 선형공격 및 차분공격에 저항성을 가지도록 설계하였다[12]고 알려져 있는데, 이 실험에서 사용한 특성이 LAT와 DDT인 점이 영향을 미친 것으로 판단된다.

Table 4. Maximum Identification Round S-box Structure

(max round(accuracy(%)))

Category	$\mathcal{D}_D$	$\mathcal{D}_L$	$\mathcal{D}_{D\|\|L}$
$\mathcal{F}_e^R$ vs $\mathcal{R}$	8R(93)	8R(91)	8R(92.5)
$\mathcal{M}_e^R$ vs $\mathcal{R}$	6R(68.3)	6R(69)	6R(70.5)
$\mathcal{F}_k^R$ vs $\mathcal{R}$	6R(65)	6R(61.5)	6R(63.3)
$\mathcal{M}_k^R$ vs $\mathcal{R}$	5R(85.5)	4R(100)	4R(100)

VI. 결 론

이번 실험을 통하여 S-box의 알려지지 않은 설계 원리를 딥러닝 모델이 식별할 수 있는지를 알아보았다. *Feistel*, *MISTY*, *SPN*, x^{-1}구조를 사용한 확장 S-box의 경우 구조는 물론이고 라운드까지 식별 가능한 것을 실험적으로 확인하였으며, 이론적으로 랜덤순열과 구분되는 것이 알려진 *Feistel* 구조의 경우 발표된 라운드 수보다 2라운드 증가한 경우까지, *MISTY* 구조는 1라운드 증가한 경우까지 식별 가능함을 실험적으로 알 수 있었다. 딥러닝의 특성상 구별자가 구별에 활용한 특징점을 정확히 알 수는 없지만, 각 구조별 특성에서 나오는 불능 차분이나 연산과정에서 발생하는 LAT, DDT 계수들 간의 관계를 통하여 S-box를 식별하는 것으로 추측해본다.

최근 경량암호에 대한 수요가 많아지면서 작은 bit S-box가 암호설계간 많이 사용되고 있다. 이러한 경량암호 분석간 우리가 설계한 도구를 사

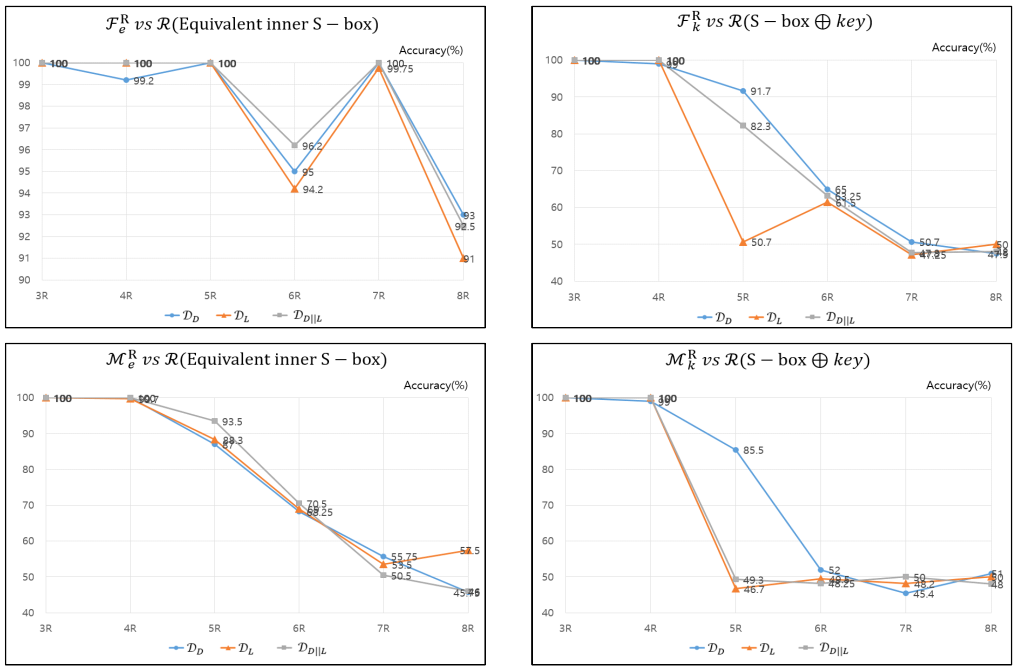

Fig. 13. Result Identification S-box Structure

용시 설계 원리가 미공개된 S-box에 대한 근본적인 정보를 획득할 수 있다. 이러한 정보는 암호분석을 위한 역공학을 수행할 경우 하나의 Trapdoor 역할을 할 수 있을 것으로 생각된다. 또한, S-box 설계 후 검증과정에서 우리의 딥러닝 구별자 도구를 사용한다면 설계자가 생각하지 못한 특성을 찾아내어 암호의 안전성을 강화하는데 활용할 수 있을 것으로 생각된다.

한편, 실험적 측면에서 LAT와 DDT 특성을 혼합한 경우 오히려 실험에 부정적 영향을 주는 것도 식별하였는데 차후 암호분석에 활용할 수 있는 좋은 계기가 되었다. 향후에는 도출된 결과를 바탕으로 모델을 최적화 하여 더 큰 S-box나 많은 라운드, 다른 설계원리들에 대해서도 정교하게 식별 가능한 딥러닝 구별자를 구현하는 방법에 대한 연구를 진행해 볼 예정이다.

References

[1] BIRYUKOV, Alex; PERRIN, Léo. On reverse-engineering S-Boxes with hidden design criteria or structure. In: Annual Cryptology Conference. Springer, Berlin, Heidelberg, 2015. p. 116-140.

[2] LUBY, Michael; RACKOFF, Charles. How to construct pseudorandom permutations from pseudorandom functions. SIAM Journal on Computing, 1988, 17.2: 373-386.

[3] PATARIN, Jacques. Generic attacks on Feistel schemes. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2001. p. 222-238.

[4] GILBERT, Henri; MINIER, Marine. New results on the pseudorandomness of some blockcipher constructions. In: International Workshop on Fast Software Encryption. Springer, Berlin, Heidelberg, 2001. p. 248-266.

[5] DAEMEN, Joan; RIJMEN, Vincent. The design of Rijndael. New York: Springer-verlag, 2002.

[6] BIHAM, Eli; SHAMIR, Adi. Differential cryptanalysis of DES-like cryptosystems. Journal of CRYPTOLOGY, 1991, 4.1: 3-72.

[7] MATSUI, Mitsuru. Linear cryptanalysis method for DES cipher. In: Workshop on the Theory and Application of of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1993. p. 386-397.

[8] TAN, Cheng; JI, Qingbing. An approach to identifying cryptographic algorithm from ciphertext. In: 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN). IEEE, 2016. p. 19-23.

[9] DE MELLO, Flávio Luis; XEXÉO, José AM. Identifying Encryption Algorithms in ECB and CBC Modes Using Computational Intelligence. J. UCS, 2018, 24.1: 25-42.

[10] KWON, Donggeun, et al. Improving Non-Profiled Side-Channel Analysis Using Auto-Encoder Based Noise Reduction Preprocessing. Journal of the Korea Institute of Information Security & Cryptology, 2019, 29.3: 491-501.

[11] GOHR, Aron. Improving Attacks on Round-Reduced Speck32/64 Using Deep Learning. In: Annual International Cryptology Conference. Springer, Cham, 2019. p. 150-179.

[12] MATSUI, Mitsuru. New block encryption algorithm MISTY. In: International Workshop on Fast Software Encryption. Springer, Berlin, Heidelberg, 1997. p. 54-68.

〈저자소개〉

김 동 훈 (Dong-hoon Kim) 학생회원
2009년 2월: 육군3사관학교 전자공학과 졸업
2019년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 정보보호, 대칭키 암호분석, 암호정책

김 성 겸 (Seonggyeom Kim) 학생회원
2016년 8월: 한양대학교 수학과 졸업
2016년 9월~2018년 8월: 고려대학교 정보보호대학원 석사
2019년 3월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 암호 알고리즘 설계 및 분석, 대칭키 암호, 난수발생기

홍 득 조 (Deukjo Hong) 종신회원
1999년 8월: 고려대학교 수학과 학사
2001년 8월: 고려대학교 수학과 석사
2006년 2월: 고려대학교 정보보호대학원 박사
2006년 3월~2007년 12월: 고려대학교 정보보호기술연구센터 연구교수
2007년 12월~2015년 8월: 국가보안기술연구소 선임연구원
2015년 9월~현재: 전북대학교 IT정보공학과 부교수
〈관심분야〉 암호 알고리즘 설계 및 분석

성 재 철 (Jaechul Sung) 종신회원
1997년 8월: 고려대학교 수학과 학사
1999년 8월: 고려대학교 수학과 석사
2002년 8월: 고려대학교 수학과 박사
2002년 8월~2004년 1월: 한국정보보호진흥원 선임연구원
2004년 2월~현재: 서울시립대학교 수학과 전임강사, 조교수, 부교수, 교수
〈관심분야〉 암호 알고리즘 설계 및 분석

홍 석 희 (Seokhie Hong) 종신회원
1995년: 고려대학교 수학과 학사
1997년: 고려대학교 수학과 석사
2001년: 고려대학교 수학과 박사
1999년 8월~2004년 2월: ㈜시큐리티 테크놀로지 선임연구원
2003년 3월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U. Leuven ESAT/SCD-COSIC 박사후 연구원
2005년 3월~2013년 8월: 고려대학교 정보보호대학원 부교수
2013년 9월~현재: 고려대학교 정보보호대학원 정교수
〈관심분야〉 대칭키 및 공개키 암호 알고리즘, 부채널 공격 및 대응기법, 디지털 포렌식