Related-key Neural Distinguisher on Block Ciphers SPECK-32/64, HIGHT and GOST

[1]Erzhena Tcydenova, [2] Byoungjin Seok and [3*]Changhoon Lee

Abstract

With the rise of the Internet of Things, the security of such lightweight computing environments has become a hot topic. Lightweight block ciphers that can provide efficient performance and security by having a relatively simpler structure and smaller key and block sizes are drawing attention. Due to these characteristics, they can become a target for new attack techniques. One of the new cryptanalytic attacks that have been attracting interest is Neural cryptanalysis, which is a cryptanalytic technique based on neural networks. It showed interesting results with better results than the conventional cryptanalysis method without a great amount of time and cryptographic knowledge. The first work that showed good results was carried out by Aron Gohr in CRYPTO'19, the attack was conducted on the lightweight block cipher SPECK-/32/64 and showed better results than conventional differential cryptanalysis. In this paper, we first apply the Differential Neural Distinguisher proposed by Aron Gohr to the block ciphers HIGHT and GOST to test the applicability of the attack to ciphers with different structures. The performance of the Differential Neural Distinguisher is then analyzed by replacing the neural network attack model with five different models (Multi-Layer Perceptron, AlexNet, ResNext, SE-ResNet, SE-ResNext). We then propose a Related-key Neural Distinguisher and apply it to the SPECK-/32/64, HIGHT, and GOST block ciphers. The proposed Related-key Neural Distinguisher was constructed using the relationship between keys, and this made it possible to distinguish more rounds than the differential distinguisher.

Keywords: *Related-key Attack, Neural Cryptanalysis, Distinguisher, Deep Learning, Lightweight Block Ciphers*

I. Introduction

It is difficult to imagine the development of ICT without cryptography. To ensure that our information is securely transmitted, stored, and used, cryptanalytic algorithms must meet a variety of criteria and be able to resist cryptanalytic attacks. With the rise of the Internet of Things a new direction in cryptography has emerged which is a lightweight cryptography. Since lightweight block ciphers have a smaller block and key sizes and simpler structure than conventional cryptography, it can be more easily targeted by new cryptanalytic attacks. One new attack of growing interest is neural cryptanalysis. It is a new cryptanalysis technique that uses artificial neural networks to conduct a cryptanalytic attack. Neural networks are widely used and have proven to be a great technique in many fields such as image classification or natural language processing and have shown error rates lower than human error [1]. But the performance of neural networks in the field of cryptanalysis has not yet shown significant results, and neural cryptanalysis is still at an early stage. Research in this area has been conducted for more than 10 years, and only recently the results of neural cryptanalysis have improved the results of conventional cryptanalysis. The most significant results were shown in the paper presented in Crypto'19, in which Neural Distinguisher was applied to the lightweight block cipher SPECK-32/64, and this attack showed better accuracy than a conventional differential distinguisher [2]. Cryptanalysis is a very complex task and sometimes takes years to complete, but neural cryptanalysis has been able to show better results in much less time. This neural

[1] *Ph.D. Candidate, Seoul National University of Science and Technology (etcydenova@seoultech.ac.kr)*
[2] *Ph.D., Seoul National University of Science and Technology (sbj7534@seoultech.ac.kr)*
[3*Corresponding Author] *Professor, Seoul National University of Science and Technology (chlee@seoultech.ac.kr)*

Received: Feb. 13, 2023, Revised: Feb. 26, 2023, Accepted: Feb. 27, 2023

distinguisher was constructed using differential characteristics of the cipher and there can be room for improvement by using other cryptanalytic methods or other neural network models.

In this paper, we extend our study on Neural Cryptanalysis [3][4]. First, we apply the Differential Neural Distinguisher (DND) to find if it is successful on ciphers other than SPECK. Target ciphers were chosen among current or past national standard algorithms that have different structures. The target ciphers are HIGHT and GOST block ciphers. Then, we construct the Neural Distinguisher using different neural network models to see if it the performance can be improved. The Neural Distinguisher by Aron Gohr was constructed using a residual neural network model ResNet which the winner of ImageNet Large Scale Visual Recognition Competition (ILSVRC'15). After the competition, more neural network models were introduced that have better performance than ResNet. Neural network models for this study were chosen among the winners of the ILSVRC. ResNext, SENet, AlexNet, and classical Multi-Layer Perception models were chosen for the study. Then, we construct a new Neural Distinguisher using related-key properties - Related-key Neural Distinguisher, and apply it to SPECK, HIGHT and GOST block ciphers. The Related-key Neural Distinguisher was able to distinguish more rounds than the Differential Neural Distinguisher. Also, using differential and related-key characteristics found by paper[5], we apply the Related-key Differential Neural Distinguisher for GOST. It was able to distinguish up to 30 rounds out of 32. This paper is the first to apply a neural distinguisher on block cipher HIGHT and GOST.

The remainder of this paper is organized as follows. Section 2 introduces background and related works of neural cryptanalysis. Section 3 describes application of Aron Gohr's Differential Neural Distinguisher to HIGHT and GOST. In section 4, we describe a method for comparison of neural network models performance on distinguishing attack. In section 5, we propose a new method of constructing a neural distinguisher - Related-key Neural Distinguisher. Section 6 provides experiment results of methods described in sections 3, 4 and 5. And finally, section 7 concludes this paper.

II. Related Works

In the Crypto'19 paper by Aron Gohr [2], a key recovery attack was conducted using a differential neural distinguisher. This marks the first time that conventional cryptanalysis has been combined with neural networks. The study utilized the ResNet [6] model, which was the winning model in the ILSVRC'15 competition. The model took in two 32-bit data blocks as input, and outputted a result determining whether the input was a ciphertext or random data. Each of the 64 nodes in the input layer corresponded to a single bit in the two 32-bit sequences. The dataset was created by encrypting 32-bit plaintexts P_0 and $P_1 = P_0 \oplus (0x0040, 0x0000)$ using the Speck-32/64 encryption algorithm.

Architecture. The input layer is connected to a layer of bit-sliced convolutions with 32 output channels in a channels-first mode. Batch normalization is then applied to the output of these convolutions, followed by rectifier nonlinearities. The result is then passed on to the main residual tower. Each convolutional block contains two layers of 32 filters and each layer first applies the convolutions followed by batch normalization, and rectifier layer. After the final rectifier layer of the block, the output is added to the input of the convolutional block through a skip connection and sent to the next block. The prediction head consists of two hidden layers and a single output unit. The first hidden layer has 64 densely connected units, followed by batch normalization and a rectifier layer. The second hidden layer has 64 ReLu units and is densely connected layer without batch normalization. The final layer has a single output unit with Sigmoid activation function.

The proposed differential distinguisher on Speck outperformed an existing differential distinguisher and the results are shown below in Table 1.

Table 1. Differential Distinguisher on SPECK-32/64

Rounds	Model	Accuracy
5	Neural Distinguisher	0.929
5	Differential Distinguisher	0.911
6	Neural Distinguisher	0.788
6	Differential Distinguisher	0.758
7	Neural Distinguisher	0.616
7	Differential Distinguisher	0.591

Anubhab Baksi proposed Machine Learning-Assisted Differential Distinguishers for Lightweight Ciphers. In the paper, a Multilayer Perceptron was used to construct an all-in-one differential distinguisher and it was applied to six lightweight algorithms from the NIST LWC competition: GIMLI-CIPHER, GIMLI-HASH, ASCON, KNOT-256, KNOT-512, and CHASKEY. The distinguisher had two phases, offline and online, and was constructed using multiple input differences. It was successful in distinguishing 8 rounds of GIMLI-CIPHER and GIMLI-HASH, 3 rounds of ASCON, 10 rounds of KNOT-256, 12 rounds of KNOT-512, and 4 rounds of CHASKEY [7].

Jaewoo So presented Deep Learning-Based Cryptanalysis of Lightweight Block Ciphers. The study proposed a generic deep learning based cryptanalysis tool for finding keys from plaintext-ciphertext pairs and applied it to Simplified DES (SDES), SIMON-32/64, and SPECK-32/64. The results showed that the tool was effective in finding text-based keys with a high probability, although the experiments were limited to a reduced key space of only 64 ASCII characters [8].

Tarun Yadav et al. proposed Differential-ML Distinguisher: Machine Learning based Generic Extension for Differential Cryptanalysis. This approach combines classical differential cryptanalysis with machine learning to perform a distinguishing attack on encryption algorithms. The proposed method extends a r-round differential distinguisher using an s-round neural network distinguisher to attack r+s rounds of the encryption. The method was performed on SPECK-32/64, SIMON-32/64, and GIFT-64/128 ciphers, resulting in the ability to distinguish 9 rounds of SPECK-32/64, 8 rounds of SIMON-32/64, and 8 rounds of GIFT-64/128 [9].

Emanuele Bellini et al. conducted a comparison of the performance of deep learning-based and conventional cryptographic distinguishers. They presented two network architectures for the distinguishers and tested them on the TEA and RAIDEN block ciphers. The results showed that the neural network-based distinguishers outperformed the classical ones without requiring excessive computational resources [10].

In the EUROCRYPT'21 conference, Adrien Benamira et al. presented a study "A Deeper Look at Machine Learning-Based Cryptanalysis". This study aimed to analyze the learning principles of the neural distinguisher proposed by Aron Gohr [2]. The study conducted a series of experiments to determine if the neural distinguisher could learn the features of the cipher and then they tried to interpret the results cryptographically. The experiments were divided into two perspectives: a cryptanalysis perspective and a machine-learning perspective. The results showed that the distinguisher learned not only the difference between input ciphertexts but also the internal differences in the penultimate and antepenultimate rounds. By analyzing the neural network model used for the distinguisher, the authors were able to extract important components and retain almost the same accuracy [11].

2.1. Distinguishing Attack

A distinguishing attack is a type of attack where the attacker is given a "black box" that holds either ciphertext or random data. This type of attack is usually the first step in other cryptographic attacks that aim to decrypt ciphertexts. The purpose of the distinguishing attack is to differentiate between ciphertext generated by the target encryption algorithm using an unknown key, and random bits sequences. The tool used to make this differentiation during a distinguishing attack is called a distinguisher [12].

If there is a distinguishing property that is able to distinguish n-bit cipher C from a random permutation R by having black-box access to the permutation with a probability $p > 2^{-n}$ (greater than the probability of a random permutation), there is exist a distinguishing attack against the cipher C. Given plaintext-ciphertext pairs (P, C) of the full-round block cipher C, the existence of such a distinguisher can result the recovery of the round key.

2.2. Related-key Attack

In a related-key attack, the attacker selects a relationship between a pair of keys, but the keys themselves are not known to the attacker. The data is encrypted using both keys. In a variation where the plaintext is known, the attacker knows the plaintext and ciphertext of the data encrypted using two keys. The objective of the attacker is to find the actual secret keys. It is assumed that the attacker knows or chooses a mathematical relation between the keys [13].

Let's say, the relation is $K_1 = F(K_0)$, where F is a function known or selected by the attacker and (K_0, K_1) are related keys. Related-key differential attack is one of the forms of this attack where the

relation is simply a XOR operation with a constant C: $K_1 = K_0 \oplus C$. This type of attack exploits the properties of difference propagation when plaintexts X_0 and X_1, which can be equal, are encrypted with distinct keys K_0 and K_1 correspondingly. The goal of the attacker is to determine the actual secret keys.

2.3. Lightweight Block Ciphers

HIGHT encryption algorithm is a lightweight and hardware-optimized block cipher with an ARX-based Feistel structure. It uses simple operations such as XOR, addition mod 2^8, and bitwise rotation and has a block size of 64 bits and a key size of 128 bits. The algorithm consists of 32 rounds [14].

The GOST 28147-89 is a Soviet encryption standard that was published in 1989 and it consisted of the block cipher known as Magma or simply GOST. However, it was later replaced by the GOST R 34.12-2015 standard, which also included Magma but with a fixed S-Box and a new block cipher called Kuznyechik. In 2018, the standard was updated once again to become the GOST 34.12-2018, which incorporated both Magma and Kuznyechik. GOST uses a Feistel structure with 32 rounds and has a block size of 64 bits and a key size of 256 bits [15].

2.4. Neural Network Models

In this paper, we employed 5 different neural network models: Multi-layer Perceptron (MLP), AlexNet, ResNet, ResNext, and Squeeze-and-Excitation Network (SENet). The MLP is a simple feedforward artificial neural network that maps inputs to outputs using nonlinear connections between nodes. The network consists of at least three layers: input, hidden, and output. The output of each node is scaled by a weight and passed on to the next layer [16]. However, since it is a fully connected network, it can be inefficient due to the large number of parameters that increase with each layer. On the other hand, Convolutional Neural Networks (CNNs) have smaller weights that are shared, making them easier to train than MLPs. The other models used in this study (AlexNet, ResNet, ResNext, and SENet) are all CNNs.

AlexNet is a first convolutional neural network that was initially designed for image classification, and it showed a great performance in the ILSVRC competition. It has five convolutional layers, interspersed with pooling and normalization layers, followed by three fully connected layers, and finally, the output is processed through a SoftMax loss function. AlexNet was the first to utilize the ReLU non-linearity and it separates normalization layers. To prevent overfitting, it employs dropout instead of regularization [17].

ResNet is a deep convolutional network that was proposed in 2015 and became the winning model of the ILSVRC'15 competition. It solved the issue of convergence and degradation that traditional convolutional networks experienced when built to be too deep. ResNet achieved this by introducing shortcut connections, which convert the network into a residual version. The shortcut connection $H(x) = F(x) + x$ can be utilized when the input and output have the same dimensions. The architecture of ResNet is composed of a large stack of identical residual blocks, each of which contains two 3x3 convolutional layers. There is also an additional convolutional layer at the start of the architecture, and batch normalization is applied after each convolutional layer [6].

ResNext is a deep convolutional network that takes its design from ResNet and was the first runner-up in the ILSVRC'16 competition. It introduces a new idea known as "cardinality" and its effectiveness is due to a building block that aggregates multiple transformations. Instead of going deeper, increasing the "width" (cardinality) has been proven to reduce validation errors [18].

The Squeeze-and-Excitation Network (SENet) is a novel component that can be added to any deep convolutional neural network without increasing computational cost but improving its accuracy. It won the ILSVRC'17 competition and consists of a Global Average Pooling layer (the squeeze part) followed by two fully connected layers with ReLU and Sigmoid activation functions (the excitation part) [19].

III. Application of Differential Neural Distinguisher to HIGHT and GOST

The Differential Neural Distinguisher (DND) was able to distinguish ciphertext from random data up to 7 rounds, with better accuracy than conventional methods, but we can assume that DND is optimized only for the SPECK-32/64 cipher and would not work well on other ciphers, especially those with a Substitution-Permutation Network (SPN) structure. To assess its versatility, the distinguisher was tested on two additional ciphers, HIGHT and GOST.

First, to apply the DND on different ciphers, such as HIGHT and GOST, a dataset needs to be generated for training and validation. The dataset is created by using the Python pseudo-random generator API *urandom* to generate plaintexts P, keys K and labels L. If the label is set to 1, the ciphertext C is generated by encrypting the plaintext P, while the ciphertext C' is generated by encrypting the plaintext P with an input differential Δ. If the label is 0, C and C' are generated randomly. The resulting ciphertext pairs are then converted to binary. The overall algorithm for generating the training and validation data *GenDataDiff* is shown in Algorithm 1.

Algorithm 1. GenDataDiff

Input: Data size: m, number of rounds: n, input differential: Δ
Output: Binary data: D, labels: L

Generate random sequences $P = (P_0, \dots, P_m),\ K = (K_0, \dots, K_m),\ L = (L_0, \dots, L_m)$
for $i = 0;\ i < m; i \leftarrow i + 1$ **do**
  if $L_i == 0$ **then**
    Generate random C_i, C_i'
  else if $L_i == 1$ **then**
    $C_i = Encrypt_{K_i}^n(P_i)$
    $C_i' = Encrypt_{K_i}^n(P_i \oplus \Delta)$
  end if
end for
$D \leftarrow ConvertToBinary(C||C')$
return D, L

To create the training and validation datasets for the DND, Algorithm 1 is utilized. The result of the DND is the highest validation accuracy in distinguishing ciphertext data from random data. Algorithm 2 for the Differential Neural Distinguisher is shown below.

Algorithm 2. DND

Input: Number of rounds: n, input differential: Δ, epochs: e
Output: Best validation accuracy: acc

Train data size = m, validation data size = m', number of rounds = n, tmp = 0
Train data: $D_{train} \leftarrow GenDataDiff(m, n, \Delta)$
Validation data: $D_{val} \leftarrow GenDataDiff(m', n, \Delta)$
for $i = 0;\ i < e; i \leftarrow i + 1$ **do**
  $acc \leftarrow ResNet(D_{train}, D_{val})$
  if $acc > tmp$ **then**
    $tmp \leftarrow acc$
  end if
end for
$acc \leftarrow tmp$
return acc

In differential cryptanalysis, the choice of differential characteristics is crucial. The DND for SPECK utilized differential characteristics that had a high probability of success, which is typically indicated by a low Hamming weight for the difference. The experiments focused on one-bit differences and Algorithm 3 was used to exhaustively search for the best difference. The DND was run for every bit difference, and the difference that provided the best results was selected as the final input difference.

Algorithm 3. Search for input differential

Input: Block size: *s*
Output: Differential with the best accuracy: Δ

Number of rounds = *n*, epochs = *e*, tmp = 0
for $i = 0;\ i < s; i \leftarrow i + 1$ **do**
    $\Delta' \leftarrow 2^i$
    $acc = DND(n, e, \Delta)$
    if $acc > tmp$ **then**
        $tmp \leftarrow acc$
        $\Delta \leftarrow \Delta'$
    end if
end for
return Δ

IV. Comparison of Neural Network Models Performance on Distinguishing attack

The Neural Network that was employed in the DND is the ResNet model, which won the ILSVRC in 2015. ILSVRC is an annual competition that assesses algorithms for large scale image classification and object detection using subsets of the ImageNet dataset. Image classification is the task of identifying what an image represents, and it is done by finding patterns and features that make the image recognizable. After the release of ResNet, several Neural Network models with improved accuracy have been proposed. So, it is possible that using these models might increase the accuracy of the Neural Distinguisher. To explore this possibility, the DND was applied to the ciphers HIGHT and GOST using different NN models (Algorithm 4), including classical MLP, AlexNet, ResNext, and SENet block combined with ResNet and ResNext (SE-ResNet and SE-ResNext). Additionally, since the input for NN models in classification (images) is different from that in the distinguishing attack (encrypted binary data), simple early NN models such as the Multilayer Perceptron (MLP) were also considered for the experiments.

Algorithm 4. NN Models performance comparison

Input: List of models: *models = [MLP, AlexNet, ResNet, ResNext, SE−ResNet, SE–ResNext]*
Output: Model with best accuracy: *M*

Number of rounds: *n*, input differential: Δ, epochs: *e*
Train data size = m, validation data size = m', number of rounds = n
Train data: $D_{train} \leftarrow GenDataDiff(m, n, \Delta)$
Validation data: $D_{val} \leftarrow GenDataDiff(m', n, \Delta)$
$tmp = 0, tmp' = 0$
for i in range *models* **do**
    for $i = 0;\ i < e; i \leftarrow i + 1$ **do**
        $acc \leftarrow models[i](D_{train}, D_{val})$
        if $acc > tmp$ **then**
            $tmp \leftarrow acc$
        end if
    end for
    $acc \leftarrow tmp$
    if $acc > tmp'$ **then**
        $tmp' \leftarrow acc$
        $M \leftarrow models[i]$
    end if
end for
return *M*

V. Related-key Neural Distinguisher

The combination of Neural Distinguisher with differential cryptanalysis resulted in better results than conventional differential distinguisher, indicating that combining Neural Distinguisher with other cryptanalytic methods might also produce similar results and improve the results of the DND. Related-key distinguisher is a type of distinguisher that exploits the relationship between keys to distinguish ciphertexts from random permutations. One of the relationships used in related-key distinguisher is differential, where the differential propagation leads to a particular ciphertext when a plaintext is encrypted with specific related keys. In this study, we construct the Related-key Neural Distinguisher (Figure 1) and apply it to the SPECK-32/64, HIGHT, and GOST ciphers.

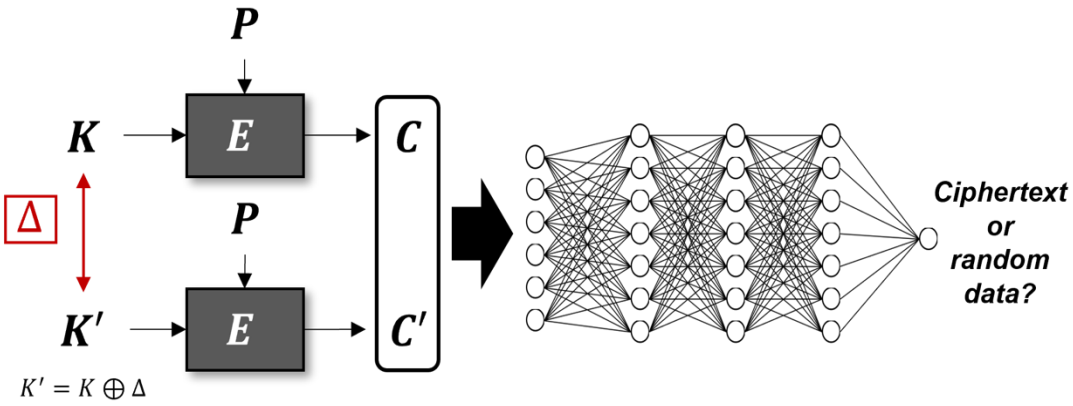

Figure 1. Related-key Neural Distinguisher

Datasets for training and validation of the distinguisher is generated using Algorithm 5. Plaintexts P, keys K and labels L are randomly generated using Python API *urandom*. Ciphertext C is a result of encryption plaintext P with K and ciphertext C' is a result of encryption of plaintext P with $K \oplus \Delta$, if it has label 1. If it is labeled as 0, C and C' are generated by *urandom*. Concatenated ciphertext pairs (C, C') are then converted to binary.

Algorithm 5. GenDataRelKey

Input: Data size: m, number of rounds: n, input differential: Δ
Output: Binary data: D, labels: L

Generate random sequences $P = (P_0, \dots, P_m),\ K = (K_0, \dots, K_m),\ L = (L_0, \dots, L_m)$
for $i = 0;\ i < m; i \leftarrow i + 1$ **do**
  if $L_i == 0$ **then**
    Generate random C_i, C_i'
  else if $L_i == 1$ **then**
    $C_i = Encrypt_{K_i}^{n}(P_i)$
    $C_i' = Encrypt_{K_i \oplus \Delta}^{n}(P_i)$
  end if
end for
$D \leftarrow ConvertToBinary(C||C')$
return D, L

Using Algorithm 5 training and validation datasets for the Related Key Distinguisher (RKD) are generated. Neural Network model M used in this attack is model with best performance found by Algorithm 4. The output of the RKD is best validation accuracy of distinguishing ciphertext data from random data. Algorithm 6 is illustrated bellow.

Algorithm 6. RKD

Input: Number of rounds: n, input differential: Δ, epochs: e
Output: Best validation accuracy: acc

Train data size = m, validation data size = m', number of rounds = n, tmp = 0
Train data: $D_{train} \leftarrow GenDataRelKey(m, n, \Delta)$
Validation data: $D_{val} \leftarrow GenDataRelKey(m', n, \Delta)$
for $i = 0;\ i < e; i \leftarrow i + 1$ **do**
    $acc \leftarrow M(D_{train}, D_{val})$
    if $acc > tmp$ **then**
        $tmp \leftarrow acc$
    end if
end for
$acc \leftarrow tmp$
return acc

The selection of differentials is also crucial for the related-key distinguisher, similar to the differential distinguisher. In the experiments, one-bit differences were utilized and were selected through an exhaustive search process using Algorithm 7.

Algorithm 7. Search for related-key input differential

Input: Key size: k
Output: Differential with the best accuracy: Δ

Number of rounds = n, epochs = e, tmp = 0
for $i = 0;\ i < k; i \leftarrow i + 1$ **do**
    $\Delta' \leftarrow 2^i$
    $acc = RKD(n, e, \Delta)$
    if $acc > tmp$ **then**
        $tmp \leftarrow acc$
        $\Delta \leftarrow \Delta'$
    end if
end for
return Δ

VI. Experiments

6.1. Application of the Differential Neural Distinguisher on HIGHT and GOST

In this experiment, Aron Gohr's Differential Neural Distinguisher is applied to ciphers HIGHT and GOST (Algorithm 2). Dataset was generated using Algorithm 1. Data size for training = 10^6, for validation = 10^5. Input differentials for target ciphers were chosen using Algorithm 3 as follows:

- HIGHT: $\Delta = (0x00800000, 0x00000000)$
- GOST: $\Delta = (0x20000000, 0x00000000)$

Results. For HIGHT the DND with the input differential distinguished ciphertexts from random data up to 9 rounds with accuracy 0.7472. Accuracy of the distinguisher was ≈ 1 up to round 8. The DND on GOST using the input differential was able to distinguish encrypted data from random up to 9 rounds with

accuracy 0.5430. Accuracy of the distinguisher was ≈ 1 up to round 6. Overall results are shown in Table 2.

Table 2. Results of Differential Neural Distinguisher on HIGHT and GOST

Rounds	HIGHT	GOST
1	1.0	1.0
2	1.0	1.0
3	1.0	1.0
4	1.0	1.0
5	1.0	0.9996
6	1.0	0.9917
7	1.0	0.8881
8	0.9990	0.6879
9	0.7472	0.5430

Discussion. The results of the Differential Neural Distinguisher indicate that the approach is not limited to just the SPECK cipher and can be used on other ciphers as well. This was demonstrated by the ability of the DND to distinguish up to 9 rounds of the HIGHT and GOST ciphers. These results suggest that the Neural Distinguisher can be effectively used on ciphers with varying structures.

6.2. Performance comparison of MLP, AlexNet, ResNext, SENet models

In this experiment, the performance of MLP, AlexNet, ResNet, ResNext, SE-ResNet, SE-ResNext neural network models were compared by applying neural distinguisher on SPECK, HIGHT, and GOST. The dataset for all experiments was collected identically with the DND by generating concatenated ciphertext pairs $(C_0||C_1)$. Training data size = 10^6, validation data size = 10^5.

Hyperparameters. Hyperparameters used for all experiments are as follows:

- Learning rate scheduler: Cyclic.
- Learning rate: Step size - 10, Max Lr - 0.002, Min Lr - 0.0001.
- Batch size: 5000.
- Optimizer. Adam.
- Loss function: Mean Square Error.

Results. The accuracy of Differential Neural Distinguisher using different neural network models on SPECK, HIGHT, and GOST are shown in Table 3.

Table 3. Performance of the Differential Neural Distinguisher using different Neural Network models

Cipher	Rounds	MLP	AlexNet	ResNet	SE-ResNet	ResNext	SE-ResNext
SPECK	5	0.8632	0.8970	**0.9049**	0.9022	0.8927	0.8930
	6	0.6468	0.7383	0.7540	**0.7558**	0.7278	0.7289
HIGHT	8	**0.9989**	0.9978	0.9990	0.9988	0.9977	**0.9989**
	9	0.7493	0.7500	0.7465	0.7515	**0.7522**	0.7506
GOST	8	0.6566	0.6803	0.6916	0.6905	0.6906	**0.6923**
	9	**0.5479**	0.5048	0.5410	0.5437	0.5461	0.5421

Discussion. The results of using neural network models with better performance on Image Classification, such as ResNet, SE-ResNet, ResNext, and SE-ResNext, for the Differential Neural Distinguisher were not significantly different from those of simpler models such as MLP and AlexNet. However, for all target ciphers except HIGHT, the more advanced models showed slightly better performance. When it came to HIGHT, all the models performed similarly. This could be due to the smaller 8-bit word size of HIGHT compared to the 16-bit word size of SPECK-32/64 and the 32-bit word size of GOST. The simple MLP model may perform better on ciphers with smaller word sizes. The convolutional networks, such as ResNet and ResNext have pretty similar performance in image classification task and they acted similarly for the distinguisher too. On the other hand, the SENet model,

which includes a Global Average Pooling Layer, performed worse in the distinguishing task than in image classification. This may be because the original ResNet model used in the distinguisher excluded the Global Average Pooling Layer, leading to a drop in performance for SENet. However, it should be noted that the performance of the models could still be improved with proper hyperparameter tuning.

6.3. Application of Related-key Neural Distinguisher on SPECK, HIGHT, and GOST

Dataset was collected by generating concatenated ciphertext pairs $(C_0||C_1)$, where $C_0 = CIPHER_K(P)$, $C_1 = CIPHER_{K\oplus\Delta}(P)$ using target ciphers SPECK, HIGHT and GOST. Training data size = 10^6, validation data size = 10^5. Input differentials were found by comparing results of distinguisher using Algorithm 7. Input differentials used for experiments are as follows:

- SPECK: $\Delta = (0x0040, 0x0000, 0x0000, 0x0000)$
- HIGHT: $\Delta = (0x00000000, 0x8000000000)$
- GOST: $\Delta = (0x0000000000000000, 0x0000000000000000, 0x0000000000000000, 0x0000000000020000)$

Results. The Related-key Neural Distinguisher on SPECK using the input was able to distinguish encrypted data from random up to 9 rounds with accuracy 0.5932. The accuracy of the distinguisher ≈ 1 up to round 6. For HIGHT it was able to distinguish up to 11 rounds with an accuracy of 0.7493. For GOST the distinguisher was able to distinguish encrypted data from random up to 14 rounds with accuracy 0.7134. The accuracy of the distinguisher was ≈ 1 up to round 10. Overall results are shown in Table 4.

Table 4. Results of Related-key Neural Distinguisher on SPECK, HIGHT and GOST

Rounds	SPECK	HIGHT	GOST
1	1.0	1.0	1.0
2	1.0	1.0	1.0
3	1.0	1.0	1.0
4	1.0	1.0	1.0
5	1.0	1.0	1.0
6	1.0	1.0	1.0
7	0.9772	1.0	1.0
8	0.8443	1.0	1.0
9	**0.5932**	0.9998	1.0
10	-	0.9991	1.0
11	-	**0.7493**	0.9995
12	-	-	0.9897
13	-	-	0.8891
14	-	-	**0.7134**

Discussion. Application of new approach Related-key Neural Distinguisher which uses differential relation between keys showed to have better results. By this attack, it was able to attack more rounds compared to Differential Neural Distinguisher. It improved the number of attacked rounds as follows:

- 2 more rounds for SPECK
- 2 more rounds for HIGHT
- 5 more rounds for GOST

By these results, we can say that the distribution of related-key characteristics has higher non-random behavior than differential characteristics for SPECK, HIGHT, and GOST.

6.3.1 Related-key Differential Neural Distinguisher on GOST

The Related-key Differential Neural Distinguisher in this experiment was constructed using information from a related-key differential attack on the full round GOST, which was presented at the FSE'04 conference. This attack was based on a related-key differential distinguisher with a probability of 1 for 24

rounds of the GOST cipher[5]. In our experiment, we built the Related-key Differential Neural Distinguisher by incorporating the related-key and differential characteristics from the FSE'04 paper. The dataset for the distinguisher was created in a similar manner to the related-key differential attack, but with the addition of input differentials for both the plaintext and key. Plaintexts P, keys K, and labels L are randomly generated using Python API *urandom*. Ciphertext C is a result of encryption of plaintext P with K and ciphertext C' is a result of encryption of plaintext $P \oplus \Delta$ with $K \oplus \nabla$ if it has label 1. If it is labeled as 0, C and C' are generated by *urandom*. Concatenated ciphertext pairs ($C||C'$) are then converted to binary. Input differentials used for the experiment are as follows:

- Δ: $(0x00000000, 0x80000000)$
- ∇: $(0x00000000, 0x80000000, 0x00000000, 0x80000000, 0x00000000, 0x80000000, 0x00000000, 0x80000000)$

Results. The Related-key Differential Neural Distinguisher on GOST was able to distinguish up to 30 rounds out of 32 with an accuracy of 0.5928 and the Accuracy of the distinguisher was ≈ 1 up to round 28 rounds, which is 4 more rounds compared to paper [5]. Overall results are shown in Table 5.

Table 5. Results of Related-key Differential Neural Distinguisher on GOST

Rounds	GOST
1	1.0
2	1.0
3	1.0
4	1.0
...	...
24	1.0
25	1.0
26	1.0
27	0.9999
28	0.9889
29	0.8272
30	0.5928

Discussion. Block cipher GOST has a very simple and weak key schedule. Conventional related-key differential distinguisher from [5] for 24 rounds has probability equal to 1 and using these already known characteristics with Neural Distinguisher allowed to distinguish 4 more rounds with accuracy ≈ 1. Even with a random 1-bit input differential, the Related-key Neural Distinguisher was able to distinguish almost half of GOST's rounds without any process of analyzing the cipher. By these results, we can say that Neural Distinguisher is able to learn characteristics of cipher and it shows good performance on ciphers that already have weaknesses.

VII. Conclusions

A distinguishing attack utilizing Neural Networks demonstrated improved performance over conventional distinguishers in terms of both accuracy and ease of use on the SPECK-32/64 cipher. Neural Networks can serve as a less complex alternative to traditional attack methods, as they don't require specific cryptographic analysis knowledge. In this paper, the Differential Neural Distinguisher proposed by Aran Gohr was evaluated on various ciphers besides SPECK and it was found that it is applicable to ciphers with different structures. The DND was able to distinguish up to 9 rounds for the ciphers HIGHT and GOST. Additionally, various neural network models were compared for their performance in the distinguishing task. The usage of neural network models with better performance on Image Classification did not show significant improvement for Neural Cryptanalysis. But performance might be improved by hyperparameter tuning. A new approach to Neural Distinguisher, the Related-key Neural Distinguisher, was introduced and it showed better results compared to the DND. For SPECK, the Related-key Neural Distinguisher distinguished 2 more rounds than the DND, and for HIGHT and GOST, it distinguished 2

and 5 more rounds, respectively. Also, using related-key differential characteristics from the existing research we improved the number of attacked rounds of the distinguisher for GOST. The Related-key Differential Neural Distinguisher was able to distinguish 28 rounds out of 32 with probability 1, which is 4 more rounds compared to the conventional related-key differential distinguisher. This study shows the potential of neural networks in combination with conventional cryptanalysis techniques to become a promising and practical attack method.

VIII. Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1F1A1076468).

IX. References

[1] Dodge, Samuel, and Lina Karam. "A study and comparison of human and deep learning recognition performance under visual distortions." 2017 26th international conference on computer communication and networks (ICCCN). IEEE, 2017.

[2] Gohr, Aron. "Improving attacks on round-reduced speck32/64 using deep learning." Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39. Springer International Publishing, 2019.

[3] E. Tcydenova, “Cryptanalysis of Lightweight Block Ciphers Based on Neural Distinguisher,” MS Thesis, Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul, Korea, 2021.

[4] E. Tcydenova, B. Seok, and C. Lee, “Related-key Neural Distinguisher on Lightweight Block Ciphers SPECK-32/64, HIGHT, SIMECK-32/64 and CHAM-64/128”, KIISC 2021, Yeongnam Branch, Korea Institute of Information Security and Cryptology, 2021.

[5] Ko, Youngdai, et al. "Related key differential attacks on 27 rounds of XTEA and full-round GOST." Fast Software Encryption: 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004. Revised Papers 11. Springer Berlin Heidelberg, 2004.

[6] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

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

[8] So, Jaewoo. "Deep learning-based cryptanalysis of lightweight block ciphers." Security and Communication Networks 2020 (2020): 1-11.

[9] Yadav, Tarun, and Manoj Kumar. "Differential-ml distinguisher: Machine learning based generic extension for differential cryptanalysis." Progress in Cryptology–LATINCRYPT 2021: 7th International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, October 6–8, 2021, Proceedings. Cham: Springer International Publishing, 2021.

[10] Bellini, Emanuele, and Matteo Rossi. "Performance comparison between deep learning-based and conventional cryptographic distinguishers." Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 3. Springer International Publishing, 2021.

[11] Benamira, Adrien, et al. "A deeper look at machine learning-based cryptanalysis." Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I 40. Springer International Publishing, 2021.

[12] Watanabe, Dai, Alex Biryukov, and Christophe De Canniere. "A distinguishing attack of SNOW 2.0 with linear masking method." Selected Areas in Cryptography: 10th Annual International Workshop, SAC 2003, Ottawa, Canada, August 14-15, 2003. Revised Papers 10. Springer Berlin Heidelberg, 2004.

[13] Biham, Eli, Orr Dunkelman, and Nathan Keller. "Related-Key Boomerang and Rectangle Attacks." Eurocrypt. Vol. 3494. 2005.

[14] Hong, Deukjo, et al. "HIGHT: A new block cipher suitable for low-resource device." Cryptographic Hardware and Embedded Systems-CHES 2006: 8th International Workshop, Yokohama, Japan, October 10-13, 2006. Proceedings 8. Springer Berlin Heidelberg, 2006.
[15] Poschmann, Axel, San Ling, and Huaxiong Wang. "256 bit standardized crypto for 650 GE–GOST revisited." Cryptographic Hardware and Embedded Systems, CHES 2010: 12th International Workshop, Santa Barbara, USA, August 17-20, 2010. Proceedings 12. Springer Berlin Heidelberg, 2010.
[16] Popescu, Marius-Constantin, et al. "Multilayer perceptron and neural networks." WSEAS Transactions on Circuits and Systems 8.7 (2009): 579-588.
[17] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Communications of the ACM 60.6 (2017): 84-90.
[18] Xie, Saining, et al. "Aggregated residual transformations for deep neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[19] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

Authors

Erzhena Tcydenova

2018: B.S. in Department of Mathematics and Informatics, Buryat State University
2021: M.S. in Dept. Of Computer Science and Engineering, Seoul National University of Science and Technology
2021 ~ present: Ph.D. Candidate in Dept. Of Computer Science and Engineering, Seoul National University of Science and Technology

Research Interests : Information Security, Cryptography, AI, Blockchain

Byoungjin Seok

2017: B.S. in Dept. Of Computer Science and Engineering, Seoul National University of Science and Technology
2019: M.S. in Dept. Of Computer Science and Engineering, Seoul National University of Science and Technology
2023: Ph.D. in Dept. Of Computer Science and Engineering, Seoul National University of Science and Technology

Research Interests: Information Security, Cryptography, AI, Digital Forensics

Changhoon Lee

2012: Assistant Professor in Dept. Of Computer Science and Engineering, Seoul National University of Science and Technology
2015: Associate Professor in Dept. Of Computer Science and Engineering, Seoul National University of Science and Technology
2020 ~ present: Professor in Dept. Of Computer Science and Engineering, Seoul National University of Science and Technology

Research Interests: Cryptography, Information Security, Cyber Threats Intelligence, Digital Forensics