Enhanced Neural Differential Distinguisher for Speck32/64 Using Attention Mechanisms and Multi Ciphertext Inputs

Xue Jiang[1,2*], Min Li[1], Makulov Kaiyrbek[3], Valerii Lakhno[2], Sahun Andrii[2]
Email: xjiang1283@163.com, limin@bbc.edu.cn, kaiyrbek.makulov@yu.edu.kz, lva964@nubip.edu.ua,
[1]School of Computer and Information Engineering, Bengbu University, Bengbu 233030, China
[2]Department of Computer systems, networks and Cybersecurity, National University of Life and Environmental Sciences of Ukraine, Kyiv 03041, Ukraine
[3]Department of Computer Science, Caspian University of Technology and Engineering, Almaty 050151, Kazakhstan
E-mail: a.sagun@nubip.edu.ua
[*]Corresponding author

Keywords: EfficientNet, person-job matching measure, deep learning, artificial intelligence, convolutional neural network

Received: December 24, 2024

In recent years, neural network-based differential distinguishers have demonstrated significant advantages in accuracy and effi-ciency over traditional differential distinguishers in symmetric cipher differential analysis. However, when dealing with ciphers involving a higher number of encryption rounds, neural network differential distinguishers still struggle to accurately identify ci-phertext pairs. To address this issue, this study proposes a neural network differential distinguisher model based on attention mechanisms and optimized ciphertext input structures. Specifically, the model first innovates the residual structure within the attention mechanism to maximize the weight of highly discriminative features, enhancing the feature extraction capability of the improved model. Secondly, a multi-scale convolution method is employed, integrating the network structure ideas of RegNet, with the addition of convolutional branches and optimization of activation functions, which further enhances the model's feature ex-traction capability. Finally, a multi-ciphertext input pattern is introduced to improve the input data information, and random key encryption is applied to the input ciphertext structure to construct multi-feature information representations of the ciphertext and encryption functions. The results from 5-8 rounds of experiments on Speck 32/64 indicate that the proposed new neural distinguisher can significantly improve discrimination accuracy to a maximum of 1.65%. On this basis, we carried out an optimization study on the construction method of the multi-ciphertext-pair dataset. The new dataset can increase the accuracy of the distinguisher by 49.16% compared to that of the single-ciphertext-pair case, and can extend the number of attack rounds from 7 to 8.

Povzetek: Raziskava predlaga izboljšan nevronski diferencirnik za Speck32/64, ki s pozornostjo in večvhodnimi šifratnimi pari poveča kvaliteto klasifikacije in število razpoznavnih krogov.

1 Introduction

At the 2019 US Secret Conference, Gohr [1] introduced a novel cryptanalysis approach using a deep learning neural network. Targeting the block cipher Speck, the strategy constructed a differentiator using a neural network, which aimed to exploit machine learning pattern recognition capabilities to enhance cryptanalysis beyond traditional methods. This groundbreaking research ignited significant interest, inspiring numerous cryptographic researchers to delve deeper into the field. Building on Gohr's work, Baksi [2] modified the input data structure from a single ciphertext pair to multiple pairs sharing the same differential value. The application of this approach to large-block ciphers such as ASCON and KNOT yielded impressive results. Researchers have compared the performance of multilayer perceptron (MLP) [3], convolutional neural networks (CNN) [4], and long short-term memory (LSTM) [5] networks for differential analysis and concluded that MLP is the most effective, while the other architectures have limitations in accuracy and training speed.

Aayush Jain [6] further refined Baksi's research by maintaining the original input data structure while optimizing the MLP network. Additionally, following M. Wang's [7] findings, the optimal input difference for the PRESENT cipher was incorporated as a fixed input difference, which resulted in improved differential analysis accuracy for three to five rounds of the PRESENT cipher. Emanuele Bellini [8] integrated cryptographic algorithm characteristics into the neural network architecture to provide prior knowledge. The proposed neural network differentiator comprised two components: a time distinguisher and a feature extractor. The input ciphertext pair was divided into four equal segments, with each segment processed by two dense layers of 32 neurons. This approach aimed to independently extract features from each ciphertext block, minimizing inter-block interactions and aligning more closely with the cryptographic algorithm's structure. In the second part of their research, the authors employed an MLP network for prediction. They introduced a novel neural network distinguisher to conduct a differential analysis on 4–7 rounds of TEA and RAIDEN encryption algorithms, surpassing the performance of three traditional differential distinguishers. Subsequent research has primarily focused on enhancing the neural network's acquisition of prior cryptographic knowledge. For instance, Lijun Lyu [9] integrated mixed-integer linear programming (MILP) to construct a neural network differential distinguisher. By pre-determining the required difference value (δ) using MILP, the authors processed reduced ciphertext and extracted ciphertext pairs with the difference δ as the neural network input, resulting in more regular input data. HengChuan Su [10] transitioned from independent differential pairs to polyhedral differential pairs, utilizing polyhedral differences to establish closer connections between input data, thereby enabling the neural network to extract more ciphertext features. In 2021, Adrien Benamira [11] analyzed Gohr's cryptanalysis research, and by shifting the input from differential pairs to differential values and dynamically adjusting these values during testing, three key conclusions were drawn: (1) a neural network distinguisher's performance is directly proportional to its acquired cryptographic information; (2) the performance of binary differential discriminators is influenced by the penultimate round's differential distribution; (3) while a neural network structure cannot be entirely replaced by other machine learning algorithms, incorporating local network modifications using machine learning techniques can enhance differential analysis accuracy. Benamira's study marked the first exploration of the working principles of neural network distinguishers. In 2022, Hayato Kimura [12] continuously cracked the low-pass cipher algorithm, iteratively refining network parameters based on experimental results to realize white-box attacks. Unlike previous research focusing on improving neural network structure and input data, Kimura's work emphasized interpretability by examining the neural network's decision-making process from a cryptographic perspective. Additionally, neural network differential distinguishers have achieved significant results in the differential analysis of various lightweight encryption algorithms [13-15].

However, there are noticeable issues in the existing research. For instance, the neural network structures used are still relatively simple, and most studies only perform basic parameter tuning on the networks. There is limited research on modifying the neural network structures by incorporating prior knowledge of cryptographic algorithms. The feature information provided by single or multiple fixed differential ciphertext pairs is limited, and the current input data structure fails to offer more feature information to the neural network, which restricts the accuracy of the network in distinguishing between ciphertext pairs.

To address these issues, this paper proposes a differential cryptanalysis model based on an attention mechanism and residual structure. The goal is to enhance the discriminative capabilities of neural network differential distinguishers, broaden the diversity of input data, and strengthen the model's learning and generalization abilities. We optimize the neural network structure by designing an attention-based residual module to enhance feature extraction capabilities and improve the model's discriminative power. We also expand the structure of input data by extending single ciphertext pairs to multiple concatenated ciphertext pairs, combined with encryption and decryption using random keys. This optimizes the input ciphertext structure, enhances the network's ability to learn ciphertext features, and improves the effectiveness of multi-round encryption algorithms. The results from 5-8 rounds of experiments on Speck32/64 indicate that the proposed distinguisher model achieves an improvement in accuracy of 0.16% for individual ciphertext pairs, and 0.85% for multiple ciphertext pairs. The number of rounds recognized can be extended to eight.

The specific contributions of this paper are as follows:

(1) Design of an Attention-based Residual Improvement Scheme: We propose a novel neural network distinguisher model that combines multi-layer convolutions and convolutional branches, exploring the impact of different network structures on the distinguisher's accuracy. This significantly enhances the network's feature extraction capabilities in complex encryption environments.

(2) We propose an innovative model for a multi-scale convolutional enhancement scheme that integrates RegNet's multi-scale convolution method and network design principles. By utilizing stacked 3x3 convolutions and adding convolutional branches, we investigate the impact of these modifications on the accuracy of the distinguisher. This approach significantly enhances the model's feature extraction capabilities.

(3) Development of a multi-ciphertext pair input structure-based encryption method: by utilizing random keys for one-round encryption, we achieve diversity in input data, strengthening the neural network's learning capabilities regarding ciphertext features. This significantly improves the model's generalization capabilities and training efficiency.

The structure of the paper is as follows: Section 2 introduces the fundamentals of the Speck32/64 cipher system, differential analysis, and the design and operation of differential distinguishers. Section 3 delves into the optimization of neural network distinguishers, including a comparative analysis of various neural network architectures, optimization of network parameters, and improvements in ciphertext input format. Section 4 presents the experiments and analysis. Finally, Section 5 provides the conclusion and directions for future work.

2 Preliminaries

2.1 Speck32/64

Speck [16] is a lightweight block cipher introduced by Ray Beaulieu et al. from the National Security Agency (NSA) in June 2013. This study focuses on Speck 32/64, which employs a 32-bit block size and a 64-bit key. The algorithm's core is a combination of modular addition, shift, and bitwise XOR operations. The encryption process for Speck 32/64 is shown in Figure 1. The internal structure of this cipher comprises two 16-bit blocks: a left ciphertext block (Li) and a right ciphertext block (Ri). The subkey for the i-th round is denoted as Ki. The left ciphertext block is right-shifted by 7 bits and then modularly added to the right ciphertext block, followed by an XOR operation with the current round's subkey to produce a new left ciphertext block. The left-shifted right ciphertext block is XORed with this new left block to generate a new right ciphertext block. This process iterates for 22 rounds, resulting in the final ciphertext.

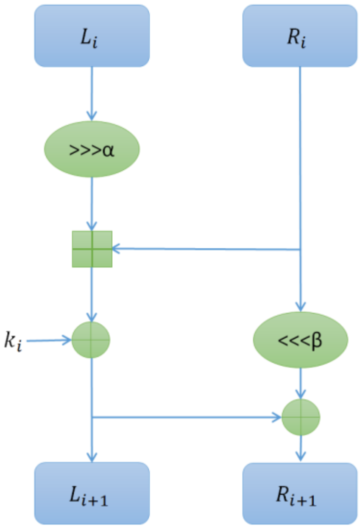

Figure 1: The round function and key schedule algorithm of Speck 32/64.

In one round of Speck cipher encryption, the input data are denoted as (L_i, R_i), and the output data are denoted as (L_{i+1}, R_{i+1}). The encryption process can be described as follows:

$$L_{i+1}=((L_i >>> \alpha) + R_i)\oplus k_i \quad (1)$$

$$R_{i+1} = (R_i <<< \beta) \oplus L_{i+1} \quad (2)$$

where ⊕ represents modular addition, $<<<, >>>$ represent bitwise cyclic shifts to the left and right in the version of Speck 32/64, α = 3, and β = 7.

2.2 How differential cryptanalysis and differential distinguishers work

Differential cryptanalysis is a potent technique for breaking block ciphers. The ability of a block cipher to resist this attack is a critical measure of its overall security. The core principle of differential cryptanalysis involves exploiting the non-uniform probability distribution of ciphertexts resulting from specific input differences [17]. By analyzing these patterns, an attacker can gather information about the key, significantly reducing the search space for potential keys.

Introduced by Biham [18] and Shamir in 1990, differential cryptanalysis is a plaintext attack that differentiates encrypted ciphertext from random data. The method hinges on identifying high-probability differential paths within the cipher's structure and constructing a differential distinguisher based on these paths. Due to its effectiveness against iterative cryptosystems, differential cryptanalysis is a standard tool for analyzing block ciphers. As a result, it is a fundamental metric for evaluating cipher security, and it has made significant contributions to the field of cryptanalysis and cryptographic security.

Differential cryptanalysis is a chosen-plaintext attack technique that exploits the statistical properties of block ciphers [19]. A differential distinguisher identifies high-probability differential patterns within the encryption algorithm, distinguishing valid ciphertext pairs from random ones. This information is used to filter potential key values. Specifically, a distinguisher for $\gamma - 1$ rounds can differentiate valid ciphertext pairs from random ones, enabling a key recovery attack on the full γ-round cipher.

A differential divider refines the distinguisher's output by identifying ciphertext pairs that align with a high-probability differential propagation path. This maximizes the number of "correct pairs" while minimizing the number of "wrong pairs", thereby reducing the key search space. The distinguisher's accuracy is crucial; higher accuracy leads to fewer false positives, a faster key search, and a more effective cryptanalysis.

Recent advancements in deep learning have led to the development of deep learning-based cipher distinguishers capable of achieving high accuracy. These distinguishers can be used to improve subsequent cryptanalytic efforts.

2.3 Differential analysis based on deep learning

Deep learning [20] is a machine learning technique predicated on artificial neural networks (ANNs) that process and analyze data by mimicking the interconnected structure and function of the human brain. It excels in handling complex tasks such as image and speech recognition, as well as natural language processing. Within this domain, convolutional neural networks (CNNs) constitute a critical branch specifically designed for image data processing. CNNs extract image features through convolutional, pooling, and fully connected layers. Convolutional layers employ convolutional kernels (filters) to scan input images and extract local features, while pooling layers reduce dimensionality and computational load, enhancing model generalization.

In 2015, He Kaiming et al. introduced the ResNet model [21], a convolutional neural network architecture incorporating skip connections. The residual module comprises two or more convolutional layers and a skip connection, directly summing the input and output to transmit the shallow network output to deeper layers, mitigating the vanishing gradient problem. Figure 2 illustrates the ResNet network structure. Despite its advancements, ResNet still presents opportunities for enhancing recognition accuracy, computational efficiency, and model parameter count. This study incorporates numerous residual structure design principles into the optimization of the neural network differential distinguisher's architecture.

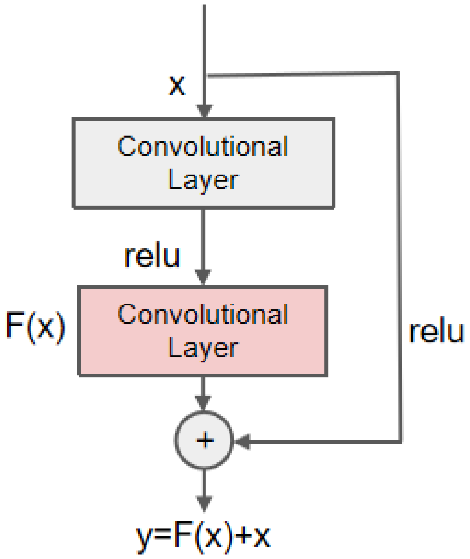

Figure 2: ResNet model structure

In 2019, Gohr pioneered the integration of deep learning with differential cryptanalysis. This study proposes a binary classification method for analyzing ciphertext pairs, building upon existing techniques for Speck encryption. Focusing on the input differential of (0x0040/0000), the goal is to distinguish between genuine and random ciphertext pairs. Real pairs originate from plaintext pairs with a specified difference, while random pairs stem from arbitrary plaintext pairs. When applied to Speck 32/64, this method surpasses traditional approaches in accuracy. Its applicability extends beyond Speck to other lightweight ciphers.

Gohr's neural network architecture, illustrated in Figure 3, comprises four modules: input, initial convolution, residual, and prediction modules. The input module feeds ciphertext data to the initial convolution module, structured as a ResNet network with 1 × 1 convolution kernels for feature extraction. The residual module enhances feature extraction using ten two-convolution neural networks and residual connections to minimize information loss. The prediction module employs multiple fully connected layers to map input features to output labels, classifying pairs as genuine or random and ultimately determining accuracy.

3 Methodology

As shown in Figure 3, the neural network-based differential distinguisher is composed of four modules: the input module (Module 1), initial convolution module (Module 2), residual module (Module 3), and prediction module (Module 4). This study focuses on optimizing and improving the network structure of the input module and the residual module to enhance the classification accuracy of the neural network-based differential distinguisher.

In this section, the structure and advantages of the network model proposed in this study are described in detail. The overall network structure is shown in Figure 4. In the process of improving the neural network differential distinguisher, the focus is primarily on optimizing the residual module and input data structure to enhance the model's distinguishing accuracy and performance. To improve the network architecture, attention-based [22] improvements were introduced to the residual module, enhancing feature extraction capabilities. This allows the neural network to better focus on important features, thus improving accuracy when distinguishing ciphertext pairs. Additionally, integrating RegNet's multi-scale convolution method into the neural network differential discriminator enhances feature extraction capabilities by incorporating convolution branches within the network architecture. By concatenating multiple ciphertext pairs and encrypting with a random key for one round, the input data are extended from single ciphertext pairs to multiple concatenated ciphertext pairs, and differential values are introduced with one round of random key encryption. This improvement not only increases the diversity of input data but also enhances the neural network's ability to learn ciphertext features, thereby improving the model's generalization ability and training efficiency. Based on multiple ciphertext pairs, further optimizations were made to accommodate higher rounds of the Speck encryption algorithm, improving the overall distinguishing accuracy. These improvements, achieved through network structure and input data optimization, significantly enhance the neural network differential distinguisher's performance in analyzing high-round cryptographic algorithms.

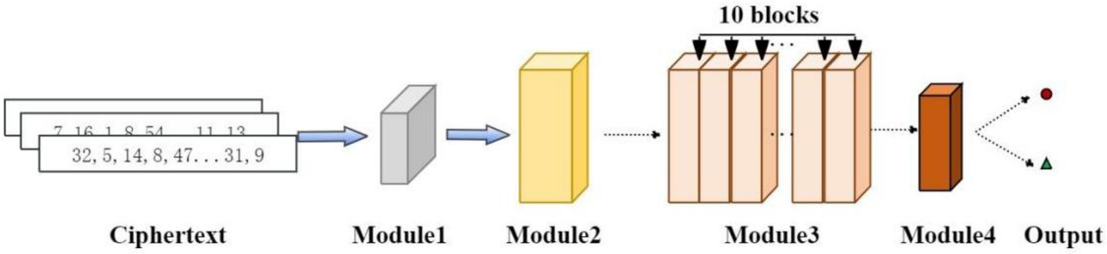

Figure 3: Gohr neural network

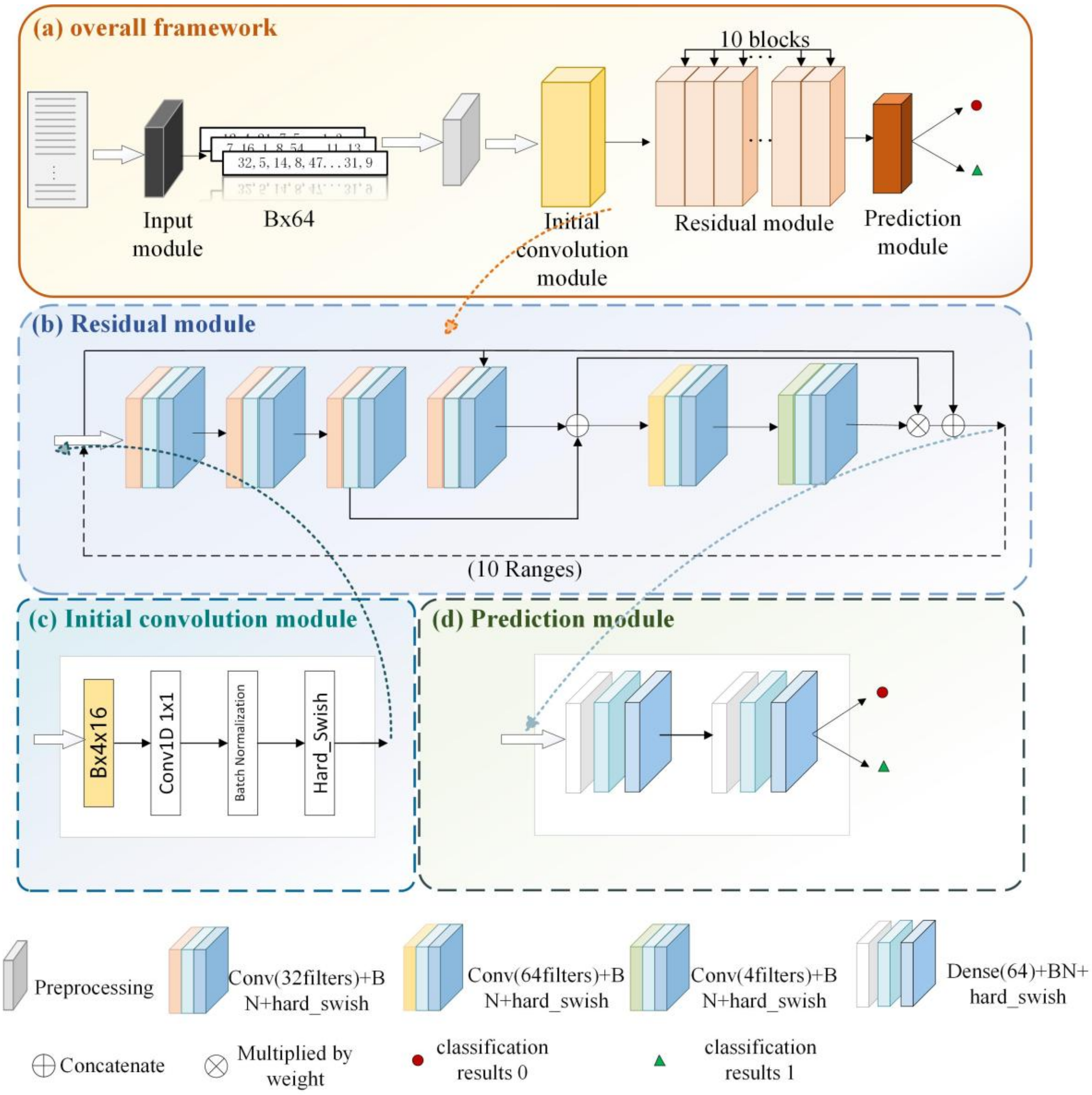

Figure 4: The overall structure of B-C3-HSwish.

3.1 Improvement of network architecture

Based on the original Gohr neural network differential distinguisher, we constructed a more accurate neural network differential distinguisher tailored for Speck 32/64 encryption reduced to 5–7 rounds. The specific improvements are detailed in Sections 3.1.1 and 3.1.2, where the differential classification accuracy for the 5–7 rounds of the Speck encryption algorithm is tested to validate the optimal improvement strategy. In Section 3.2, after determining the optimal structure for the distinguisher, the input ciphertext structure is further optimized to enhance the performance of the distinguisher.

3.1.1. Improvement of residual module based on channel attention

Typically, differential neural distinguishers are constructed using multiple cascaded residual structures [23]. The input of each residual structure is added to the input of the next, which helps reduce the risk of overfitting and improves the model's generalization capability. Meanwhile, the SENet attention mechanism aids the model in better utilizing information across different channels, allowing the network to adaptively adjust the importance of various channels and enhance the focus on significant features, thereby improving the classification performance [24].

In this study, we conducted extensive experimental investigations on the network width and depth of the residual module, aiming to determine the optimal number of filters, the optimal number of residual towers, and the optimal size of convolutional kernels. This was done so that the differential distinguisher could optimally acquire ciphertext feature information and achieve the highest possible distinguishing accuracy. However, after an in-depth analysis of the experimental results, we found that these adjustments did not significantly enhance the model's performance.

Through comprehensive and systematic experimental analysis and comparison, we identified the optimal residual module and its internal residual tower structure, as detailed below. Based on experimental testing, we first selected a residual tower comprising three one-dimensional convolutional layers, with each layer having a convolutional kernel size of 3×3. Additionally, we integrated an attention mechanism module, combining the channel attention extraction structure with the residual tower. After the residual tower learned from the ciphertext data, the outputs from the convolutional layers within the tower were concatenated. This transformed the two-dimensional data (16, 32) into three-dimensional data (16, 32, n). Subsequently, the data was pooled to form a tensor of shape (1, 1, n), which was then passed through a convolutional neural network for further learning and feature extraction. The resulting weights were multiplied with the original concatenated data. The adjusted channels were summed to restore the data to its original two-dimensional form, while maintaining the overall data volume throughout the process. The network structure is shown in Figure 5.

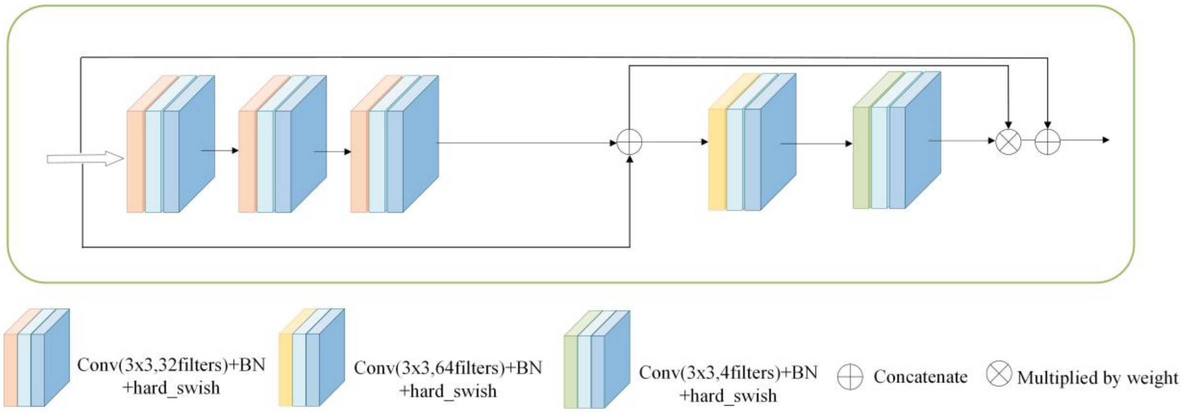
Conv(3x3,32filters)+BN +hard_swish
Conv(3x3,64filters)+BN +hard_swish
Conv(3x3,4filters)+BN +hard_swish
Concatenate
Multiplied by weight

Figure 5: residual tower structures.

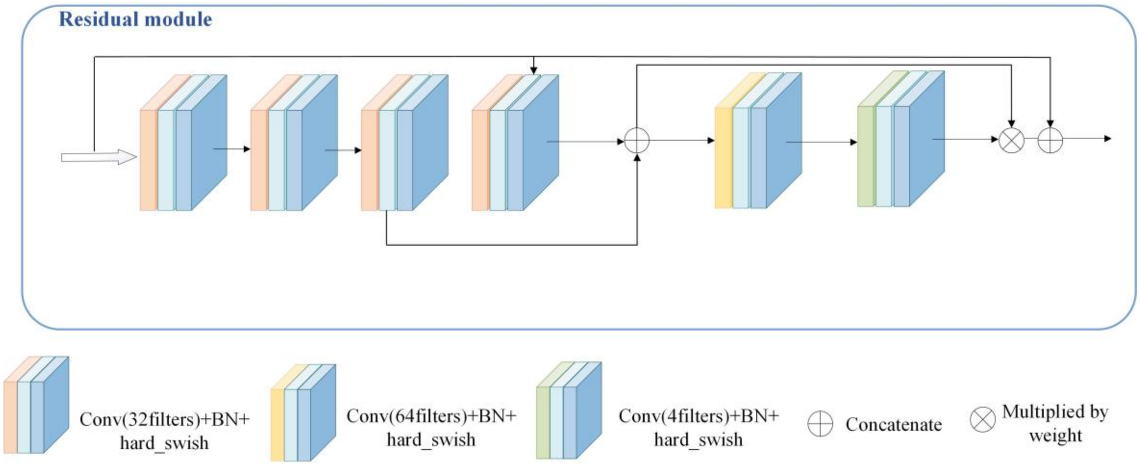
Residual module
Conv(32filters)+BN+ hard_swish
Conv(64filters)+BN+ hard_swish
Conv(4filters)+BN+ hard_swish
Concatenate
Multiplied by weight

Figure 6: Branch-CNN3.

3.1.2. Structure of feature enhancement module based on convolution branch

RegNet is an efficient convolutional neural network architecture renowned for its systematic design and uniform network structure. It facilitates multi-scale learning through diverse building blocks, effectively enhancing the depth and breadth of feature extraction, while also improving learning efficiency through parameter optimization [25]. In this study, we integrate the multi-scale convolution approach of RegNet into a traditional neural network differential distinguisher, using only two stacked 3x3 convolutional layers for feature extraction. The effect of stacking two 3x3 convolutions is equivalent to the receptive field of a single 5x5 convolution. Thus, we adopt a multi-scale convolution strategy, incorporating RegNet's network design principles by adding convolutional branches to the network structure of the differential distinguisher. Additionally, we modify the activation functions to further optimize the feature extraction quality of these branches, thereby enhancing the distinguisher's feature extraction capabilities. The detailed architecture is shown in Figure 6.

3.2 Improvement of the input module

The experiments in the previous section demonstrated that modifying the neural network differentiator architecture can significantly enhance differential discrimination accuracy. However, it was also observed that optimizing the neural network structure alone is insufficient for achieving higher-round differential attacks. To further improve model accuracy and overall performance, this section focuses on optimizing the input dataset. By manipulating the dataset structure, the neural network can extract more ciphertext information and structural characteristics of the encryption function. This leads to higher-quality, more diverse input data, enhancing the model's generalization ability, training efficiency, and overall performance.

Previous research (i.e., [26, 27]) has proposed to change the input mode from a single ciphertext pair to the input of multiple ciphertext pairs concatenated together. Experiments have shown that if the input mode is changed from inputting a single ciphertext pair at a time to inputting multiple concatenated ciphertext pairs, the differential distinguisher will learn more feature information, resulting in a significant improvement in the discrimination accuracy. In this paper, we improve on this idea by using multiple ciphertext pairs, namely p1, p2, ⋯, p32. After encrypting them for one round with a random key, c1, c2, ⋯, c32 are obtained. The differential values of these multiple ciphertext pairs and the differential values after one-round encryption (d1, d2, ⋯, d32) are concatenated and input into the neural network differential distinguisher. This model is named RKMP (Random Key Multi-Cipher Pairs), and is shown in Figure 7.

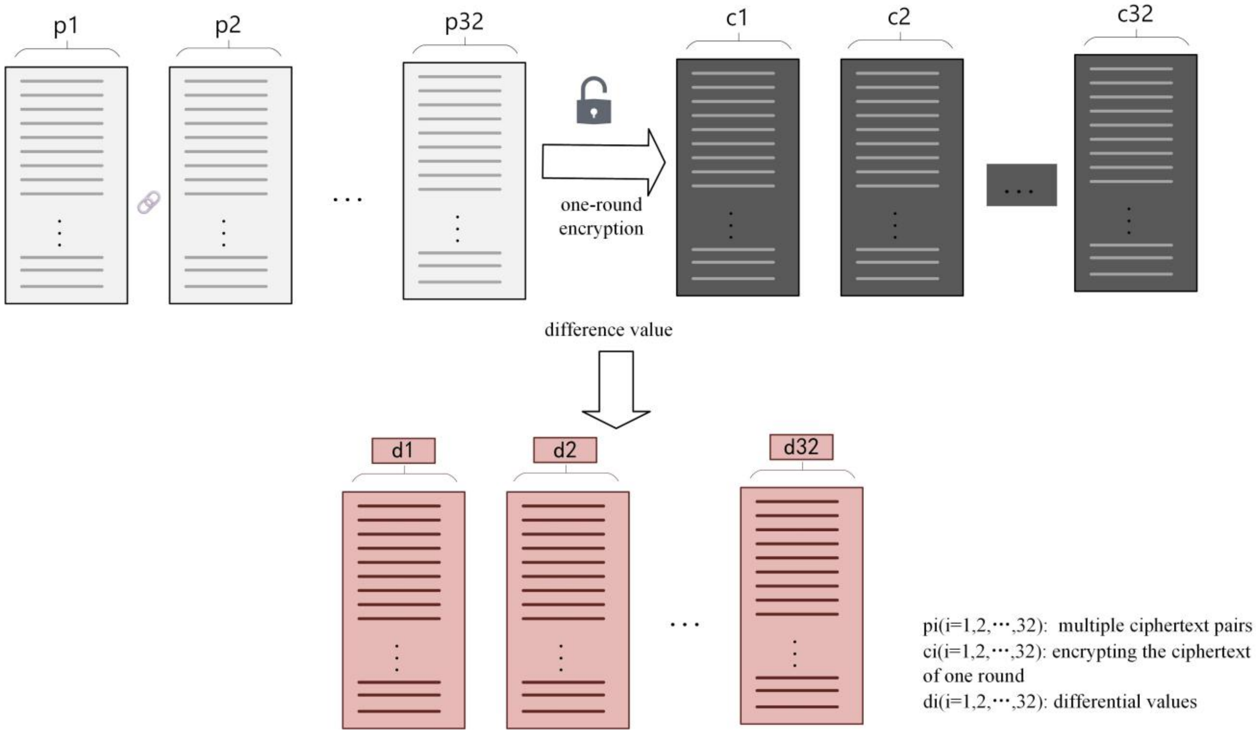
p1
p2
⋯
p32
one-round encryption
c1
c2
⋯
c32
difference value
d1
d2
⋯
d32
pi(i=1,2,⋯,32): multiple ciphertext pairs
ci(i=1,2,⋯,32): encrypting the ciphertext of one round
di(i=1,2,⋯,32): differential values

Figure 7: New input structure RKMP.

4 Experiments

4.1. Dataset

To ensure a fair and consistent experimental environment, the dataset and hyperparameters remain fixed throughout the study. Additionally, the neural network's random seed parameter is stabilized to mitigate the impact of potential floating-point variations on the results.

Network parameters: The neural network was trained on the training set for 200 epochs. The batch size was set to 5000. The Adam algorithm with default parameters in Keras was used to optimize the cross-entropy loss function, with a small penalty for L2 weight regularization (regularization parameter c = 10^{-5}). The learning rate uses a cyclical learning rate, with the learning rate l_i for $epoch_i$ set to $l_i = \alpha + \frac{(n-i) \bmod (n+1)}{n} \cdot (\beta - \alpha)$, where α = 10^{-4}, $\beta = 2 \cdot 10^{-3}$, n = 9 [28]. At the end of each epoch, the obtained network is saved, and the best network is evaluated based on the test set.

Data generation: A fixed random seed Linux random number generator was used to generate the required key, and the sizes of the training and test sets were set to 10^7 and 10^6, respectively. The fixed differential ciphertext pair was obtained by encrypting the plaintext pair with a difference of (0x0040, 0x0000) for n rounds, while the random ciphertext pair was encrypted with uniformly distributed plaintext pairs. The fixed-difference cipher was marked with a Y label of «1», and the random ciphertext was marked with a Y label of «0».

Feature Extraction: The network takes two ciphertexts (C_L^{r1}, C_R^{r1}) and (C_L^{r2}, C_R^{r2}) as inputs, which were fed into the network in the form of $(C_L^{r1}, C_R^{r1}, C_L^{r2}, C_R^{r2})$. It predicts whether the ciphertext pair conforms to the initial difference (0x0040, 0x0000). In the initial convolutional layer, a single-layer convolution operation with a kernel size of 1 is used to extract features from the input ciphertext matrices. The purpose of this step is to mimic the XOR operation in cryptographic computations. Therefore, this convolutional layer learns the convolution of the four-bit pairs that are XORed in cryptographic operations, thereby extracting their features.

Training cost: According to the above basic training plan, in a batch size of 5000 cases, a single GTX 3090 graphics training epoch takes about 90 s. Therefore, a complete training cycle can be finished in less than a day.

4.2. Improvement of residual module

4.2.1. Improvement of residual module based on attention idea

To investigate the effect of the number of convolutional layers inside the residual tower on the performance, we designed three different residual structures (S1, S2, and S3) containing two, three, and four 3*3 one-dimensional convolutional layers, respectively. These structures are shown in Figure 8 (a), (b) and (c). We added the attention mechanism to these basic structures to form three new structures, SeNet-S1, SeNet-S2, and SeNet-S3, as shown in Figure 8 (d), (e), and (f), respectively. The above six residual structures were tested for differentiation against five rounds of Speck ciphertext pairs; the results are shown in Table 1.

Table 1: The accuracy of different distinguishers with the improved model

Network Structure	Network Depth	Accuracy
S1	10	92.95%
S2	10	92.95%
S3	10	92.92%
SeNet-S1	10	93.01%
SeNet-S2	10	93.04%
SeNet-S3	10	93.00%

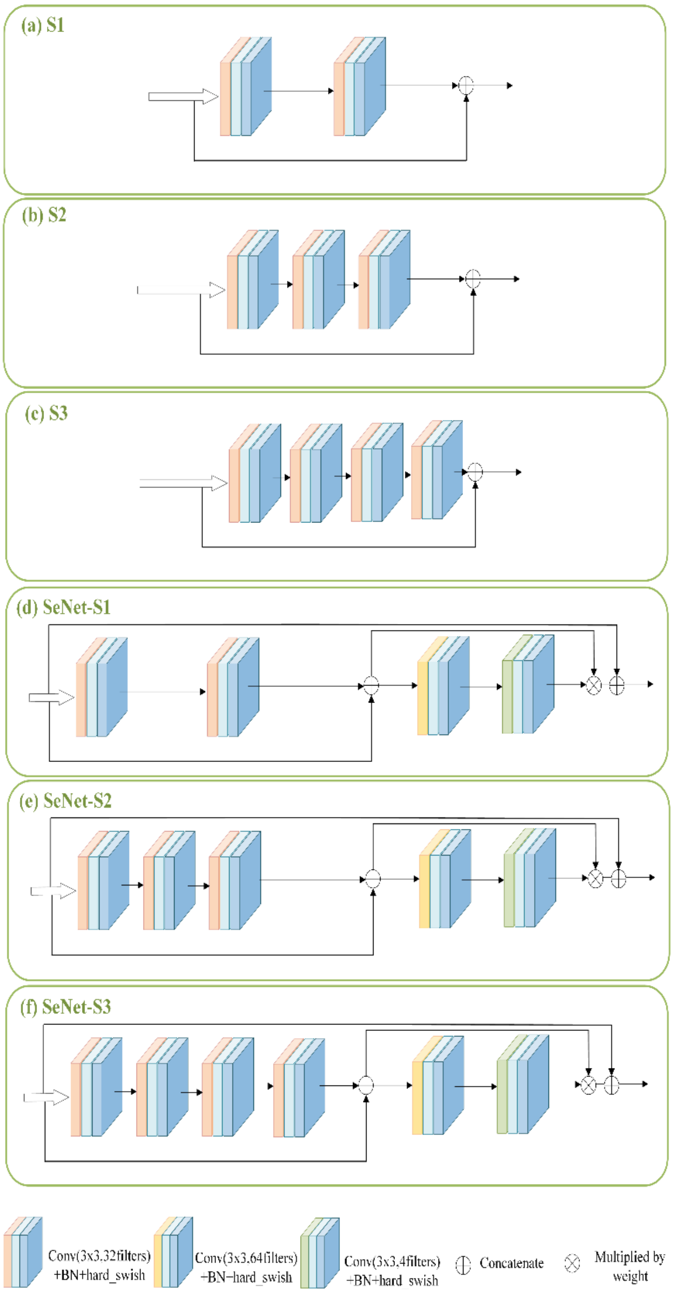
(a) S1
(b) S2
(c) S3
(d) SeNet-S1
(e) SeNet-S2
(f) SeNet-S3
Conv(3x3,32filters) +BN+hard_swish
Conv(3x3,64filters) +BN+hard_swish
Conv(3x3,4filters) +BN+hard_swish
Concatenate
Multiplied by weight

Figure 8: Six residual structures.

As shown in Table 1, SeNet-S2 demonstrates optimal performance in convolution operations. By enhancing feature representation, introducing automatic weight learning, optimizing gradient flow, reducing feature redundancy, integrating local features, and improving network stability, the model significantly outperforms its predecessors. Experimental results indicate that a three-layer convolution strikes an ideal balance between feature extraction, selection, gradient propagation, redundancy reduction, local feature integration, and model flexibility, surpassing both two-layer and four-layer configurations. This equilibrium enables the model to effectively capture complex features while maintaining computational efficiency, resulting in superior classification accuracy.

4.2.2. Structure of feature enhancement module based on convolution branch

The authors experimented with different weight learning structures and neuron counts, as summarized in Table 2.

Table 2: The accuracy of different distinguishers with the improved model

Network Structure	Add a Branch	The Structure Used for Weight Learning	Number of Neurons per Layer	Accuracy
NoBranch-MLP1(SeNet-S3)	NO	MLP	64, 3	93.04%
NoBranch-MLP2	NO	MLP	32, 3	93.01%
NoBranch-CNN1	NO	Convolution	64, 3	93.07%
Branch-MLP1	YES	MLP	64, 4	93.09%
Branch-CNN1	YES	Convolution	128, 4	93.09%
Branch-CNN2	YES	Convolution	256, 4	93.07%
Branch-CNN3	YES	Convolution	32, 4	93.11%

Table 3: Comparison of accuracy of different parameters

Activation Function	Normalization	Accuracy
ReLU	BatchNorm	93.11%
PReLU	BatchNorm	93.06%
Swish	BatchNorm	93.14%
Gelu	BatchNorm	93.14%
ELU	BatchNorm	93.06%
Selu	BatchNorm	92.99%
LeakyReLU (0.3)	BatchNorm	92.96%
Hard_swish	BatchNorm	93.16%
Hard_swish	LayerNorm	92.91%
Hard_swish	InstanceNorm	92.63%

The experimental results in Table 3 demonstrate that incorporating branch structures significantly enhances convolutional network performance, particularly for smaller neuron counts (e.g., 32 and 4). The convolutional networks consistently outperformed the MLPs in these experiments. Increasing network width proved more effective than depth in capturing input data characteristics, leading to improved model performance. Wider networks achieved higher representational capacity with fewer layers, mitigating the gradient vanishing and training challenges associated with deep networks. The shared convolution kernel in the difference distinguisher's convolutional layers effectively detected patterns across the entire input, enhancing model generalization and feature recognition. Consequently, branch addition emerges as a valuable optimization strategy.

4.3. Improving the activation function

Activation functions and normalization are crucial components in neural networks, enhancing expressivity, prediction accuracy, stability, and convergence speed [29]. Building upon the improved distinguisher in Section 3.2, this section investigates activation functions, normalization methods, and their neural network applications. Residual structures typically aim for numerical results within a specific interval in the "residual" branch. However, using the ReLU activation function at the end of this branch leads to non-negative, increasing "residual" values, potentially impacting representational power. To address this, the authors adjusted the activation function's placement or replaced ReLU with Swish, LeakyReLU, or other alternatives. Table 3 presents specific test results for five-round Speck encryption ciphertext pairs.

Normalization can improve the stability and convergence speed of the model and reduce the risk of overfitting. After an experimental comparison and analysis, it is found that the activation technic BatchNorm used in the distinguisher in this section is the optimal solution at present. Therefore, BatchNorm is still used in the distinguisher in this section, and the specific test process is shown in Table 3.

This section focuses on enhancing the neural network differential distinguisher by refining the activation function and normalization operation within the initial convolution module, residual module, and prediction module. Employing activation functions such as Hard_swish and GeLU in place of the ReLU function yields superior outcomes for discrimination. Furthermore, the normalization method significantly impacts the performance of the Hard_swish activation function, with the activation technic BatchNorm outperforming LayerNorm and other alternatives. Consequently, the authors replace the ReLU function with Hard_swish in the initial convolution module, residual module, and prediction module of the Branch-CNN3 architecture while retaining BatchNorm. Figure 9 illustrates the overall structure of this distinguisher.

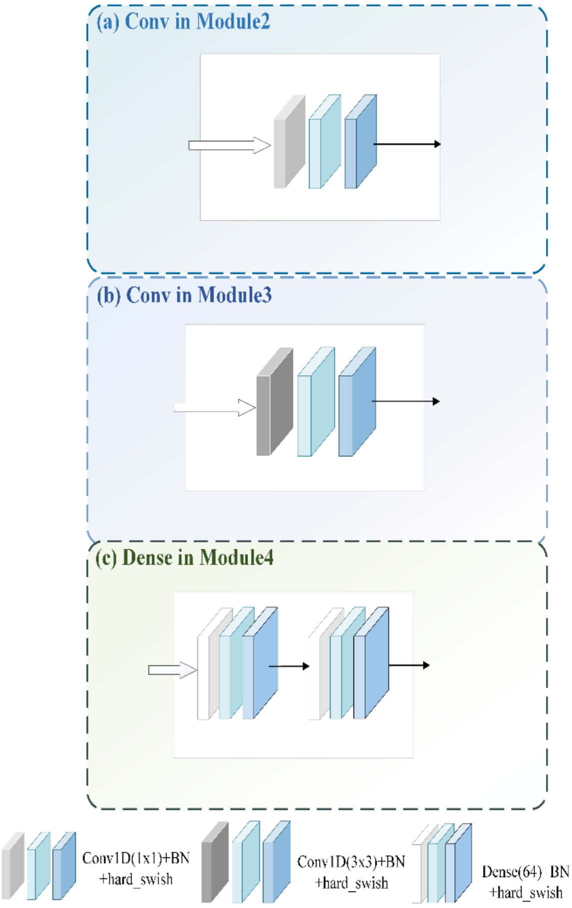

Figure 9: Activation function versus normalization improvement for Branch-CNN3.

Table 4: Comparison of accuracy of various differentiators

Round of Speck 32/64	Neural Distinguisher	Accuracy
5	CSYY22 [14]	92.6%
	Gohr [1]	92.9%
	B-C3-HSwish	93.2%
6	CSYY22 [14]	78.4%
	Gohr [1]	78.8%
	B-C3-HSwish	79.0%
7	CSYY22 [14]	60.7%
	Gohr [1]	61.6%
	B-C3-HSwish	61.7%

4.4. Comparative analysis of the accuracy of different distinguishers

At this point, the overall differential discriminator is completed and named B-C3-HSwish. The detailed structure diagram is shown in Figure 8. It was used to differentiate between five and seven rounds of the Speck encryption algorithm. A comparison of B-C3-HSwish with existing distinguishers is presented in Table 4.

In the 5-7 rounds of the Speck 32/64 encryption discrimination experiment, compared with the Gohr and CSYY22 models, the B-C3-HSwish model exhibited an accuracy advantage in different rounds. In the fifth round, the rate of improvement was approximately 0.32% compared to the Gohr model and approximately 0.65% compared to the CSYY22 model. In the sixth round, these rates were approximately 0.25% and 0.77% respectively. In the seventh round, they were approximately 0.16% and

1.65% respectively. It can be observed that under the conditions of an increasing number of encryption rounds, greater discrimination difficulty, and generally high accuracy levels, the B-C3-HSwish model can consistently maintain its advantage, thus demonstrating its effectiveness as a Speck 32/64 encryption discriminator.

4.5. Improvement of the input module

We conducted a pairwise test experiment to validate the performance of the novel ciphertext structure RKMP (random key multi-cipher pairs) in conjunction with the new splitter B-C3-HSwish, as detailed in Table 5. The results indicate that the B-C3-HSwish distinguisher outperformed the Gohr distinguisher across various round numbers, substantiating the claim that the RKMP ciphertext structure combined with the B-C3-HSwish distinguisher constitutes an optimal pairing. Despite a decrease in accuracy at higher rounds, this combination still provides effective discrimination.

Table 5: Tests of B-C3-HSwish distinguisher combined with the new ciphertext structure RKMP

Round of Speck 32/64	Ciphertext Structure	Neural Distinguisher	Accuracy
7	single-ciphertext pair	Gohr	61.6%
	single-ciphertext pair	B-C3-HSwish	61.7%
	random key multi-cipher pairs	Gohr	91.25%
	random key multi-cipher pairs	B-C3-HSwish	92.03%
8	single-ciphertext pair	Gohr	Unrecognizable
	single-ciphertext pair	B-C3-HSwish	Unrecognizable
	random key multi-cipher pairs	Gohr	63.01%
	random key multi-cipher pairs	B-C3-HSwish	63.32%

5 Conclusions

The focus of this study was to explore how neural networks can be utilized to replace traditional differential distinguishers for ciphertext pair classification. Current neural network-based differential distinguishers exhibit several significant limitations, primarily characterized by a low classification accuracy and a limited number of distinguishable encryption rounds. To address these issues, this study concentrated on two key aspects: the design of the neural network architecture for the differential distinguisher and the design of the input ciphertext structure. Building on extensive experience and existing neural network structures, the authors optimized the network architecture by improving residual modules based on channel attention, multi-scale convolutions, and activation functions in the distinguishers. We also propose a new input data structure by optimizing the input ciphertext structure, enabling the neural network to capture more ciphertext features and encryption structure information. These improvements led to the successful design of a more performant differential distinguisher, achieving an accuracy of up to 92.03% for distinguishing seven rounds of the Speck 32/64 block cipher. Furthermore, the number of distinguishable rounds was extended to eight, with an accuracy of 63.32%. The simulation results validate the effectiveness and superiority of the deep learning-based differential distinguisher design proposed in this paper for the Speck 32/64 cipher system.

In the field of cryptography, where information technology is developing rapidly and data security is of paramount importance, the security assessment of cryptographic algorithms forms the core of data security. The Speck series of cryptographic algorithms are widely used in the Internet of Things (IoT) and embedded systems due to their high efficiency and adaptability to resource-constrained environments. Taking IoT devices as an example, many sensor nodes use Speck 32/64 algorithms to encrypt and transmit data to prevent theft and tampering. However, if the algorithms' security is not secure, attackers may analyze the number of encryptions rounds to crack the encryption, resulting in privacy leakage and illegal access to data, so improving the accuracy of distinguishing between the encryption rounds of Speck 32/64 will help researchers to accurately assess the security of the encryption rounds and find loopholes, thus supporting security hardening of the devices. In embedded systems, such as smart meters and smart home control centers, which also rely on this algorithm and are closely connected to daily life, cracking the encryption will lead to serious consequences, such as confusion in energy management and threat to home security. Therefore, it is of great significance to accurately distinguish the Speck 32/64 encryption rounds to ensure the safe operation of embedded systems and maintain order. The accuracy of the distinction achieved in this study can provide a more reliable basis for the security assessment of the relevant systems in the practical applications and promote the development and improvement of the data security guarantee technology in the real world.

Although this research has made significant progress in designing differential distinguishers based on neural networks, some limitations remain. First, this study primarily focuses on the Speck 32/64 block cipher, and its applicability to other encryption algorithms has not yet been validated. Second, despite improvements in classification accuracy, performance may still be limited when dealing with higher rounds or more complex encryption algorithms. Finally, issues related to computational efficiency and resource consumption in practical applications require further optimization, and future research could explore several directions. First, the model's applicability should be tested across a broader range of encryption algorithms to verify its generalizability; second, the network architecture should be further optimized to enhance its ability to distinguish higher-round encryption; additionally, computational efficiency should be improved to make the model suitable for large-scale real-world applications, which is crucial; lastly, adaptive mechanisms should be explored that allow the model to automatically adjust to different encryption scenarios, which could significantly increase its practical value.

Acknowledgement

This study was supported by the Bengbu University 2024 Key Natural Science Research Project Bengbu University 2024 Key Natural Science Research Project (grant number 2024ZR03zd), and Bengbu University 2024 Natural Science Research Project (grant number 2024ZR03).

References

[1] Gohr, A. (2019). Improving attacks on round-reduced speck32/64 using deep learning. In Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39, 150-179. Springer International Publishing. https://doi.org/10.1007/978-3-030-26951-7_6

[2] Baksi, A., Breier J., Chen Y., Dong X. (2021). Machine learning-assisted differential distinguishers for lightweight ciphers. Design, Automation & Test in Europe Conference & Exhibition (DATE), 176-181. https://doi.org/10.23919/DATE51398.2021.9474092

[3] Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung J, et al. (2021). Mlp-mixer: An all-mlp architecture for vision. Advances in neural information processing systems, 1857, 24261-24272. https://doi.org/10.48550/arXiv.2105.01601

[4] Chua, L. O. (1997). CNN: A vision of complexity. International Journal of Bifurcation and Chaos, 7(10), 2219-2425. https://doi.org/10.1142/S0218127497001618

[5] Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. Neural computation, 31(7), 1235-1270. https://doi.org/10.1162/neco_a_01199

[6] Jain, A., Kohli, V., & Mishra, G. (2020). Deep learning based differential distinguisher for lightweight cipher PRESENT. Cryptology ePrint Archive. 2020/846, https://ia.cr/2020/846

[7] Wang, M. (2008, June). Differential cryptanalysis of reduced-round PRESENT. First International Conference on Cryptology in Africa. 40-49.

[8] Bellini, E., & Rossi, M. (2021). Performance comparison between deep learning-based and conventional cryptographic distinguishers. In Intelligent Computing: Proceedings of the 2021 Computing Conference, 3, 681-701. Springer International Publishing. https://ia.cr/2020/953

[9] Lyu, L., Tu, Y., & Zhang, Y. (2022, May). Improving the Deep-Learning-Based Differential Distinguisher and Applications to Simeck. In 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 465-470. https://doi.org/10.1109/CSCWD54268.2022.9776036

[10] Su, H. C., Zhu, X. Y., & Ming, D. (2021). Polytopic attack on round-reduced simon32/64 using deep learning. In Information Security and Cryptology: 16th International Conference, Inscrypt 2020, Guangzhou, China, December 11–14, 2020, Revised Selected Papers, 12612, 3-20. Springer International Publishing. https://doi.org/10.1007/978-3-030-71852-7_1

[11] Benamira, A., Gerault, D., Peyrin, T., & Tan, Q. Q. (2021). A deeper look at machine learning-based cryptanalysis. In 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, 12696, 805-835. Springer International Publishing. https://doi.org/10.1007/978-3-030-77870-5_28

[12] Kimura, H., Emura, K., Isobe, T., Ito, R., Ogawa, K., & Ohigashi, T. (2022, June). Output prediction attacks on block ciphers using deep learning. In International Conference on Applied Cryptography and Network Security, 248-276. Cham: Springer International Publishing. https://ia.cr/2021/401

[13] Hou, Z., Ren, J., & Chen, S. (2021). Improve neural distinguisher for cryptanalysis. Cryptology ePrint Archive. https://ia.cr/2021/1017

[14] Chen, Y., Shen, Y., Yu, H., & Yuan, S. (2023). A new neural distinguisher considering features derived from multiple ciphertext pairs. The Computer Journal, 66(6), 1419-1433. https://ia.cr/2021/310

[15] Rajan, R., Roy, R. K., Sen, D., & Mishra, G. (2022). Deep Learning-Based Differential Distinguisher for Lightweight Cipher GIFT-COFB. In Machine Intelligence and Smart Systems: Proceedings of MISS 2021, 397-406. Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-16-9650-3_31

[16] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L. (2015). The SIMON and SPECK lightweight block ciphers. 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), 1-6. https://doi.org/10.1145/2744769.2747946

[17] Sabonchi, A. K. S., & Akay, B. (2020). Cryptanalysis of polyalphabetic cipher using differential evolution algorithm. Tehnički Vjesnik, 27(4), 1101-1107. http://dx.doi.org/10.17559/TV-20190314095054

[18] Biham, E., & Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 4, 3-72. https://doi.org/10.1007/BF00630563

[19] Beyne, T., & Rijmen, V. (2022). Differential cryptanalysis in the fixed-key model. In Annual International Cryptology Conference, 13509, 687-716. Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-15982-4_23

[20] Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and trends in Signal Processing, 7(3–4), 197-387. https://doi.org/10.1561/2000000039

[21] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444. https://doi.org/10.1038/nature14539

[22] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 7132-7141. https://doi.org/10.1109/CVPR.2018.00745

[23] Wang, Z., Plaxco, K. W., & Makarov, D. E. (2007). Influence of local and residual structures on the scaling behavior and dimensions of unfolded proteins. Biopolymers: Original Research on Biomolecules, 86(4), 321-328. https://doi.org/ 10.1002/bip.20747.

[24] Li, W., Liu, K., Zhang, L., & Cheng, F. (2020). Object detection based on an adaptive attention mechanism. Scientific Reports, 10(1), 11307. https://doi.org/10.1038/s41598-020-67529-x

[25] Schneider, N., Piewak, F., Stiller, C., & Franke, U. (2017). RegNet: Multimodal sensor registration using deep neural networks. In 2017 IEEE intelligent vehicles symposium (IV), 1803-1810. IEEE. https://doi.org/10.1109/IVS.2017.7995968

[26] Chen, Y., & Yu, H. (2021). A New Neural Distinguisher Model Considering Derived Features from Multiple Ciphertext Pairs. IACR Cryptol. ePrint Arch., 2021, 310. https://ia.cr/2021/310

[27] Hou, Z., Ren, J., & Chen, S. (2021). Improve neural distinguisher for cryptanalysis. Cryptology ePrint Archive. https://ia.cr/2021/1017

[28] Smith, L. (2017). Cyclical Learning Rates for Training Neural Networks. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE. https://doi.org/10.48550/arXiv.1506.01186

[29] Wu, Z., Yu, H., Zhang, L., & Sui, Y. (2023). The Adaptive Quadratic Linear Unit (AQuLU): Adaptive Non-Monotonic Piecewise Activation Function. Tehnički Vjesnik, 30(5), 1469-1485. https://doi.org/10.17559/TV-20230614000735