Deep Learning Assisted Differential Cryptanalysis for the Lightweight Cipher SIMON

Wenqiang Tian[*] and Bin Hu
PLA SSF Information Engineering University
62 Kexue Road, Zhengzhou 450001 China
[e-mail: wchtien@126.com, hb2110@126.com]
[*]Corresponding author: Wenqiang Tian

Received October 19, 2020; revised December 16, 2020; accepted January 19, 2021; published February 28, 2021

Abstract

SIMON and SPECK are two families of lightweight block ciphers that have excellent performance on hardware and software platforms. At CRYPTO 2019, Gohr first introduces the differential cryptanalysis based deep learning on round-reduced SPECK32/64, and finally reduces the remaining security of 11-round SPECK32/64 to roughly 38 bits. In this paper, we are committed to evaluating the safety of SIMON cipher under the neural differential cryptanalysis. We firstly prove theoretically that SIMON is a non-Markov cipher, which means that the results based on conventional differential cryptanalysis may be inaccurate. Then we train a residual neural network to get the 7-, 8-, 9-round neural distinguishers for SIMON32/64. To prove the effectiveness for our distinguishers, we perform the distinguishing attack and key-recovery attack against 15-round SIMON32/64. The results show that the real ciphertexts can be distinguished from random ciphertexts with a probability close to 1 only by $2^{8.7}$ chosen-plaintext pairs. For the key-recovery attack, the correct key was recovered with a success rate of 23%, and the data complexity and computation complexity are as low as 2^8 and $2^{20.1}$ respectively. All the results are better than the existing literature. Furthermore, we briefly discussed the effect of different residual network structures on the training results of neural distinguishers. It is hoped that our findings will provide some reference for future research.

Keywords: Deep Learning, Cryptography, Lightweight Cipher, Differential Cryptanalysis, SIMON

This work was supported by National Natural Science Foundation of China (Grant Nos. 61802438).

http://doi.org/10.3837/tiis.2021.02.012 **ISSN : 1976-7277**

1. Introduction

Differential cryptanalysis has been introduced in 1990 by Biham and Shamir [1] to break the Data Encryption Standard (DES) block cipher, and today it has been considered as one of the most basic cryptanalysis methods. The vital step in differential cryptanalysis is to find the differential characteristics of the cryptographic primitives, so that an attacker can construct differential distinguishers to carry out the differential attack. On this basis, cryptanalysts have proposed improved methods such as multiple differential cryptanalysis [2], truncated differential cryptanalysis [3], and impossible differential cryptanalysis [4, 5].

The development of deep learning brings some new ideas for cryptanalysis. Deep Learning has made great improvements in recent years on many difficult tasks, especially in machine translation, image recognition and automatic driving. In the field of cryptography, existing work using machine learning techniques mainly focuses on side-channel analysis [6-8] and cryptographic implementations [9-11]. However, the successful applications of deep learning in other fields have inspired researchers that deep learning may be able to find hidden rules or non-randomness that have not been found under some classic cryptanalysis methods, so as to conduct black-box cryptanalysis.

At CRYPTO 2019, Aron Gohr [12] creatively showed how to teach neural networks to exploit differential properties of round-reduced SPECK32/64. Under the Markov assumption, Gohr first computed the full differential distribution table of round-reduced SPECK32/64 with a fixed input difference. Hence, the all-in-one differential characteristics for 5-, 6-, 7-, 8-round SPECK32/64 are reported. After this, he presented differential distinguishers based on deep residual neural networks that achieve better accuracies than the analogous classical distinguishers using the full differential distribution table. Besides, based on a variant of Bayesian optimization, he developed a highly selective key search policy which, together with the neural distinguishers, can be used to reduce the remaining security of 11-round Speck32/64 to roughly 38 bits. The inspiring attempt of this work opened a forward-looking study for the application of deep learning in black-box cryptanalysis, more and more cryptanalysts are beginning related research, such as [13]. In [14], the author breaks the full rounds of SIMON32/64 with “text key” using deep learning. However, the security of SIMON with random keys under neural differential analysis has not been evaluated. At ESORICS 2020, Hou et al. [15] use deep learning to achieve linear attack on DES with plain-cipher pairs. This is another important advance in the application of deep learning in cryptanalysis.

Constructing an effective neural network is the most critical point of cryptanalysis using deep learning. In [12], the author chooses a ResNet (Residual Network) with ReLU (Rectified Linear Unit) before addition to train the differential distinguishers. The residual network is proposed by He Kaiming et al. at CVPR 2016 [16], and they compared several variants of the residual network with various usages of activation in [17]. However, the network adopted in [12] is not considered the best performing network in [17]. Therefore, on the issue of training neural differential distinguishers, which network works best is worthy of study. In addition, whether the deep learning assisted differential cryptanalysis can be generalized to other ciphers to achieve better results than the conventional differential cryptanalysis is also unknown.

Our Contributions References [12] show that for non-Markov ciphers, neural distinguishers perform better than conventional differential distinguishers, because the differential transition matrix calculated under the Markov assumption is often inaccurate in this case. Therefore, after proving that SIMON is a non-Markov cipher, we apply the method of training neural differential distinguishers to the SIMON cipher. As a result, we present 11-, 12-, 13-round neural differential distinguishers of SIMON32/64 with the highest accuracies to date.

Secondly, to prove the advantages of our distinguishers, we use them to perform the 15-round distinguishing attack and 15-round key-recovery attack on SIMON32/64. Our high-accuracy neural distinguishers make the chosen plaintexts required for successful distinguishing attacks being roughly 2^{21} times lower than an analogous classical distinguisher in [18]. For the key-recovery attack, we successfully recover the 15-round subkey with $2^{20.1}$ computation complexity and 2^8 data complexity. This is currently the minimum complexity required to attack 15-round of SIMON32/64. To our knowledge, this is the first time that the security of SIMON32/64 with random keys is evaluated under neural differential cryptanalysis.

Lastly, to study which kind of residual network works best, we use five variants of the residual network and various parameters to train neural differential distinguishers of SIMON and SPECK respectively. We conclude that the structure of the residual network will affect the convergence rate of training, but the impact on training accuracy and validation accuracy is negligible (on the order of magnitude 10^{-4}).

2. Preliminaries

2.1 Brief Description of SIMON Cipher

SIMON is a family of lightweight block ciphers designed by Beaulieu et al. in 2013 [19]. The family consists of ciphers having a range of block size $2n$ and key size k: 32/64, 48/72, 48/96, 64/96, 64/128, 96/96, 96/144, 128/128, 128/192, and 128/256. In this paper, we mainly focus on SIMON32/64 and abbreviate it as SIMON32. SIMON has a Feistel-like construction, which uses a simple round function composed of three operations: bitwise XOR($\oplus$), bitwise AND(&), and left circular shift(□). Let (L_i, R_i) be the input of the i-th round of SIMON. Then the output of the i-th round is (L_{i+1}, R_{i+1}), and (L_{i+1}, R_{i+1}) is computed as follows:

$$L_{i+1} = F(L_i) \oplus R_i \oplus K_i, R_{i+1} = L_i, \tag{1}$$

where

$$F(x) = ((x \lll 1) \wedge (x \lll 8)) \oplus (x \lll 2). \tag{2}$$

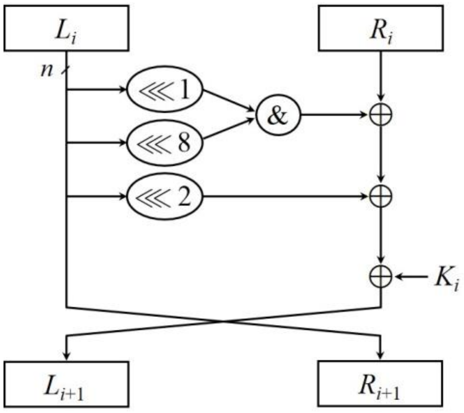
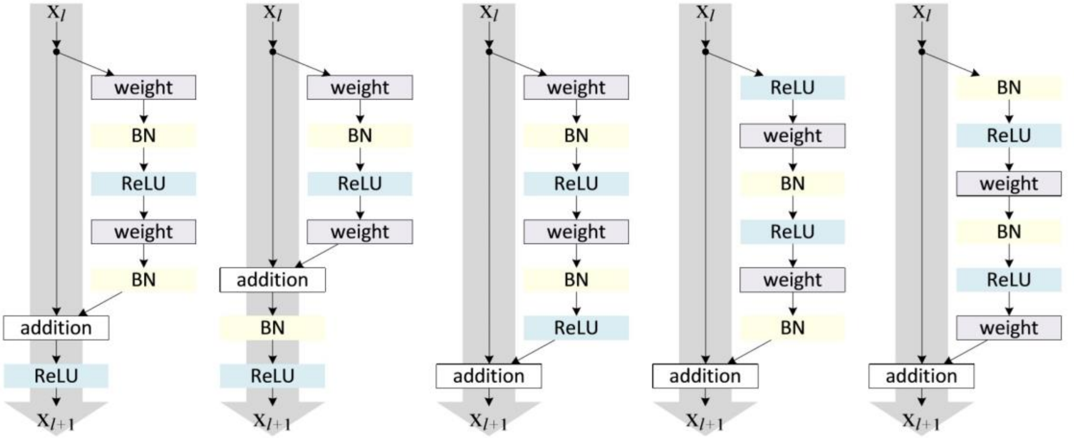


Fig. 1. The round function of SIMON

The round function of SIMON is shown in **Fig. 1**. As the key schedule is irrelevant to our differential analysis, we omit its description and refer the readers to [19]. Since SIMON is of great significance for protecting the information security of Internet of Things devices, there are various papers published on the cryptanalysis of it [20-26].

2.2 Brief Description of Deep Residual Neural Network

Deep residual neural network, which is one of the most effective and widely used convolutional neural networks (CNN) currently, is proposed by He Kaiming et. al at CVPR 2016 [16]. The core idea is to add an "identity shortcut connection" to a normal convolutional neural network, skipping one or more convolutional layers, to solve the problem of gradient disappearance when training convolutional neural network models.



(a) original (b) BN after addition (c) ReLU before addition (d) ReLU-only pre-activation (e) full pre-activation

Fig. 2. Five residual block structures [17]

The basic unit of the residual network is called the residual block. The original residual block has a structure in **Fig. 2**(a). The batch normalization (BN) is adopted after each weight layer, and ReLU is used after BN except that the last ReLU in a residual block is after elementwise addition. **Fig. 2**(b-e) show several variants of the residual block by rearranging the activation functions of ReLU and BN.

He et al. compared the performance of the five residual block structures by testing the error rate of networks using each residual block above on a classification task. The results show that Net(e) performs significantly better than other networks, while Net(b) and Net(c) perform the worst. More details of the deep residual network can be found in papers [16, 17].

For three reasons, we choose the residual neural network to train distinguishers. First, in the differential cryptanalysis, we hope that the neural network can learn the ciphertext differences obtained by XORing the ciphertext pairs. The residual neural network has been proved to be able to accomplish this task well. Second, we experimented with various network models such as fully connected networks and convolutional neural networks, and the residual neural network performed best. Third, in reference [12], the author did similar work, and his best result was also obtained by residual neural network.

3. Neural Differential Attack on SIMON32

3.1 Non-Markov Property of SIMON

The concept of Markov ciphers is first introduced by Lai, Massey and Murphy [27] at Eurocrypt 1991.

Definition 1 (Markov Chain [27]). *Given a sequence of discrete random variables* $v_0, v_1, \cdots, v_r$ *is a Markov chain, if for* $0 \le i < r$,

$$P(v_{i+1} = \beta_{i+1} \mid v_i = \beta_i, v_{i-1} = \beta_{i-1}, \cdots, v_0 = \beta_0) = P(v_{i+1} = \beta_{i+1} \mid v_i = \beta_i). \tag{3}$$

If $P(v_{i+1} = \beta \mid v_i = \alpha)$ *is independent of i for all α and β, the Markov chain is called homogeneous.*

Definition 2 (Markov Cipher [27]). *An iterated cipher with round function* $Y = f(X, K)$ *is a Markov cipher if there is a group operation* $\otimes$ *for defining differences such that, for all choices of* $\alpha(\alpha \ne 0)$ *and* $\beta(\beta \ne 0)$,

$$P(\Delta Y = \beta \mid \Delta X = \alpha, X = \gamma) \tag{4}$$

is independent of γ when the subkey K is uniformly random.

Theorem 1 [27]. *If an r-round iterated cipher is a Markov cipher and the r-round keys are independent and uniformly random, then the sequence of differences* $\Delta X = \Delta Y_0, \Delta Y_1, \cdots, \Delta Y_r$, *is a homogeneous Markov chain. Moreover, this Markov chain is stationary if ΔP is uniformly distributed over the non-neutral elements of the group.*

According to Theorem 1, the probability of an r-round differential characteristic can be computed as,

$$P(\Delta Y_1 = \beta_1, \Delta Y_2 = \beta_2, \cdots, \Delta Y_r = \beta_r \mid \Delta X = \beta_0) = \prod_{i=1}^{r} P(\Delta Y_1 = \beta_i \mid \Delta X = \beta_{i-1}). \tag{5}$$

However, for non-Markov ciphers, we cannot use the equation above to compute the probability of a characteristic anymore, as the differences in the former rounds usually have a significant effect on the differences of the latter rounds.

Theorem 2. *The cipher SIMON is a non-Markov cipher.*

Proof. Define the group operation $\otimes$ as bitwise XOR. The round function of SIMON is denoted as

$$R_k(x_0, x_1) = (y_0, y_1) = (F_k(x_0) \oplus x_1, x_0). \tag{6}$$

The function $F_k(x)$ is as follows,

$$F_k(x) = (x \ll 1) \wedge (x \ll 8) \oplus (x \ll 2) \oplus k. \tag{7}$$

For the given input (x0, x1) input difference $\alpha = (\alpha_0, \alpha_1)$ and output difference $\beta = (\beta_0, \beta_1)$,

$$\begin{aligned} P(\Delta Y=\beta \mid \Delta X=\alpha, X=\gamma) &= P(R_k(x_0 \oplus \alpha_0, x_1 \oplus \alpha_1) \oplus R_k(x_0, x_1) = (\beta_0, \beta_1)) \\ &= P((F_k(x_0 \oplus \alpha_0) \oplus F_k(x_0) \oplus \alpha_1, \alpha_0) = (\beta_0, \beta_1)). \end{aligned} \tag{8}$$

Hence,

$$P(\Delta Y=\beta \mid \Delta X=\alpha, X=\gamma) = \begin{cases} 0, & \alpha_0 \neq \beta_1, \\ P(F_k(x_0 \oplus \alpha_0) \oplus F_k(x_0) = \alpha_1 \oplus \beta_0), & \alpha_0 = \beta_1. \end{cases} \tag{9}$$

Therefore, the round function $R_k(x_0, x_1)$ has Markov property if and only if the function $F_k(x)$ has Markov property.

We have

$$F_k(x_0 \oplus \alpha_0) = ((x_0 \oplus \alpha_0) \lll 1) \wedge ((x_0 \oplus \alpha_0) \lll 8) \oplus ((x_0 \oplus \alpha_0) \lll 2) \oplus k, \tag{10}$$

$$F_k(x_0) = (x_0 \lll 1) \wedge (x_0 \lll 8) \oplus (x_0 \lll 2) \oplus k. \tag{11}$$

As all operations are bitwise, the distributive law holds for XOR, that is

$$(x \oplus y) \lll i = (x \lll i) \oplus (y \lll i), \tag{12}$$

$$(x \oplus y) \wedge z = (x \wedge z) \oplus (y \wedge z). \tag{13}$$

Then we can get,

$$\begin{aligned} F_k(x_0 \oplus \alpha_0) \oplus F_k(x_0) = ((x_0 \lll 1) \wedge (\alpha_0 \lll 8)) \oplus ((x_0 \lll 8) \wedge (\alpha_0 \lll 1)) \\ \oplus ((\alpha_0 \lll 1) \wedge (\alpha_0 \lll 8)) \oplus (\alpha_0 \lll 2). \end{aligned} \tag{14}$$

Obviously, the probability of $F_k(x_0 \oplus \alpha_0) \oplus F_k(x_0) = \alpha_1 \oplus \beta_0$ is related to x_0. Therefore, $F_k(x)$ does not satisfy the Markov property, i.e., the cipher SIMON is a non-Markov cipher.

Since SIMON is not a Markov cipher, the differential cryptanalysis under the Markov assumption is not accurate, and the theoretical results will differ from the real situation to some extent. In contrast, neural distinguishers constructed with deep learning have no such disadvantages.

3.2 Neural Distinguishers of SIMON32

Paper [12] provides a method for constructing distinguishers using differential characteristics, which can be simply described as follows.

1. Generating random and uniformly distributed keys K_i and plaintext pairs P_i with a fixed input difference as well as a vector of binary-valued real/random labels L_i.
2. To produce training and validation data for the k-round cipher, the plaintext pair P_i was then encrypted for k rounds if L_i was set, while otherwise the ciphertext was generated by encrypting the freshly generated random plaintexts.
3. Pre-process ciphertext pairs to fit the format required by the neural network and start training.

Using this method, Gohr trained distinguishers for 5-, 6-, 7-, 8-round SPECK32/64 that achieve better accuracies than the differential distinguishers using the full differential distribution table. There are two main reasons for the improved results. One is that SPECK is not strictly a Markov cipher, and the other is that the neural distinguisher can exploit features besides the differential distribution table.

In this paper, we extend the method for training neural distinguishers to SIMON cipher, and the specific procedure as **Algorithm 1**.

Algorithm 1 Training Neural Distinguishers

```
Input: Objective cipher $E$, neural network $Net$, an input difference $\delta$, number of chosen plaintext
       pairs $n$.
Output: a neural distinguisher $N$.
 1: $TD \leftarrow (\cdot)$
 2: $P \leftarrow n$ randomly generated plaintexts
 3: $K \leftarrow n$ randomly generated keys
 4: $L \leftarrow$ a randomly generated vector of $n$ binary labels
 5: for $i = 0$ to $n - 1$ do
 6:     if $L[i] = 0$ then
 7:         $P_i' \leftarrow P_i \oplus \delta$
 8:     else
 9:         $P_i' \leftarrow$ a freshly generated random plaintext
10:     else if
11:     $C_i \leftarrow E(P_i, K_i)$
12:     $C_i' \leftarrow E(P_i', K_i')$
13:     Append $TD$ by $(L[i], (C_i, C_i'))$
14: end for
15: Train $Net$ with $TD$
16: $Net$ training reports accuracy $acc$
17: if $acc > 0.5$ then
18:     $N \leftarrow Net$ after training
19: else
20:     Abort
21: end if
```

According to **Algorithm 1**, we use $\Delta = 0x0000/0400$ as the input difference (based on a differential characteristic given in [25]) to train a depth-10 residual net like the one in [12], and finally get 7-, 8-, 9-round neural differential distinguishers of SIMON32 with high accuracies. Combined with a 4-round differential transition $0x0001/4404 \rightarrow 0x0000/0400$, we can extend them to 11-, 12-, 13-round distinguishers at a fairly low additional cost. The results are in **Table 1** and **Table 2**. Our 13-round distinguisher is by far the most accurate. Prior to this, the accuracies of the best 13-round distinguishers were $2^{-30.22}$ in [21] and $2^{-28.79}$ in [18].

It should be noted that accuracy is used here as a measure of the effectiveness of the distinguishers, because it is naturally related to the distinguishing advantage of classical cryptographic distinguishers. Certainly, the median key rank and training loss are also valid measures in an attack, but they do not significantly improve the results compared to accuracy. A detailed discussion of deep learning and neural net is out of the scope of this work, and interested readers may refer to paper [12] and relevant books such as [28, 29].

In **Table 1**, **t**he 7-, 8-, 9-round neural differential distinguishers for SIMON32. *N*7-*N*9 are neural distinguishers using ciphertext pairs with chosen-plaintext difference 0x0000/0400 for 7, 8, 9 rounds. Accuracy is the probability that a sample will be correctly recognized. The true positive rate is the probability that a positive sample can be recognized as true, and the true negative rate is the probability that a negative sample can be recognized as false.

Table 1. The 7-, 8-, 9-round neural differential distinguishers for SIMON32

Rounds	Distinguisher	Accuracy	True Positive Rate	True Negative Rate
7	*N*7	0.9826	0.9986	0.9671
8	*N*8	0.7497	0.7231	0.7798
9	*N*9	0.6320	0.5297	0.7681

In **Table 2**, the 11-, 12-, 13-round distinguishers for SIMON32 obtained by adding the 4-round differential transition in front of the 7-, 8-, 9-round neural differential distinguishers. The differential characteristics are represented as hexadecimal, and *N*7/*N*8/*N*9 are neural distinguishers using ciphertext pairs with chosen-plaintext difference 0x0000/0400 for 7, 8, 9 rounds. *p* represents the differential transition probability of the truncated difference and the true positive rate of neural distinguishers. The last row is the probability of the full distinguishers.

Table 2. The 11-, 12-, 13-round distinguishers for SIMON32

Rounds	Differential Distinguisher	***p***
0 1 2 3 4	(0x0001, 0x4404) (0x4400, 0x0001) (0x1000, 0x4400) (0x0400, 0x1000) (0x0000, 0x0400)	1 2^{-2} 2^{-2} 2^{-2} 2^{-2}
4 → 11/12/13	*N*7/ *N*8/ *N*9	0.9986/0.7231/0.5297
$\sum_{acc}$	0.9986/0.7231/0.5297 × 2^{-10}	

3.3 Distinguishing Attack Against SIMON32

We have extended our neural 9-round neural distinguisher to a 13-round distinguisher by prepending a 4-round differential transition $0x0001/4404 \rightarrow 0x0000/0400$ with a probability of 2^{-10} (presented in **Table 2**). We can perform a distinguishing attack against 15-round SIMON32.

The 13-round distinguisher can be extended by another round at no additional cost, since no key addition happens in SIMON before the first nonlinear operation. Consider the plaintexts $P = (p_0, p_1)$ and $P' = (p'_0, p'_1)$, and denote the ciphertexts after one round of SIMON as $C = (c_0, c_1)$ and $C' = (c'_0, c'_1)$. Then we have

$$\begin{aligned} c_0 &= (p_0 \ll 1) \wedge (p_0 \ll 8) \oplus (p_0 \ll 2) \oplus p_1 \oplus k, \\ c_0^{'} &= (p_0^{'} \ll 1) \wedge (p_0^{'} \ll 8) \oplus (p_0^{'} \ll 2) \oplus p_1^{'} \oplus k, \end{aligned} \quad (15)$$

$$\begin{aligned} c_1 &= p_0, \\ c'_1 &= p'_0. \end{aligned} \quad (16)$$

Hence, the difference $(\Delta c_0, \Delta c_1)$ of C and C' is

$$
\begin{aligned}
\Delta c_0 &= c_0 \oplus c_0' \\
&= ((p_0 \ll 1) \wedge (p_0 \ll 8) \oplus (p_0 \ll 2) \oplus p_1) \oplus ((p_0' \ll 1) \wedge (p_0' \ll 8) \oplus (p_0' \ll 2) \oplus p_1'), \quad (17)
\end{aligned}
$$

$$\Delta c_1 = c_1 \oplus c_1' = p_0 \oplus p_0'. \quad (18)$$

Obviously, $(\Delta c_0, \Delta c_1)$ is independent of the subkey k and only related to plaintext pair (P, P'). Therefore, the 13-round distinguisher can be extended by an additional round by choosing the appropriate plaintext pairs to make the difference 0x0001/4404. Using this distinguisher, the 15-round distinguishing attack is performed as **Algorithm 2**.

Algorithm 2 Distinguishing Attack Against SIMON32

Input: Objective cipher *Oracle*, a 13-round neural distinguisher *N*, number of chosen plaintext pairs *n*.

Output: The output of *Oracle* is *True* (real) or *False* (random).

```
$(P, P') \leftarrow n$ random plaintext pairs that validate the difference 0x0001/4404 after one round of encryption.
$(C, C') \leftarrow Oracle(P, P')$
$s \leftarrow 0$
for $i = 0$ to $n - 1$ do
    for $k$ in Subkeys do
        $(D_i[k], D_i'[k]) \leftarrow \text{DecryptOneRound}((C_i, C_i'), k)$
        $v_i[k] \leftarrow N(D_i[k], D_i'[k])$
        $v_i[k] \leftarrow v_i[k] / (1 - v_i[k])$
    end for
    $v_i \leftarrow \text{Average}([v_i[k], k \in \{Subkeys\}])$
    $v_i \leftarrow v_i / (1 + v_i)$
    if $v_i > 0.5$ then
        $s \leftarrow s + 1$
        if $s > 1$ then
            return True
        end if
    end if
end for
return False
```

We explain **Algorithm 2** in detail. To distinguish 15-round SIMON32 with a 13-round distinguisher, we first generate *n* random plaintext pairs that validate the differential transition 0x0001/4404 → 0x0000/0400 after one round of encryption. Then we go through all subkeys to decrypt the output of oracle with one round of SIMON32. Put the partially decrypted ciphertexts $(D_i[k], D_i'[k])$ into the 13-round distinguisher, which will report a score $v_i[k]$ between 0 and 1. If and only if the oracle is indeed 15-round SIMON32 and the subkey k is correct, $v_i[k]$ should be greater than 0.5 and closer to 1. Otherwise, $v_i[k]$ is going to be less than 0.5 and close to 0. Then we summarize the results into a score v_i for the ciphertext pairs by transforming the scores into real-vs-random likelihood ratios and computing the average value. Finally, the number of v_i greater than 0.5 is greater than 1, we believe the oracle is

SIMON32.

In the practical attack, we use *k* probabilistic neutral bits [30] to boost the signal from this distinguisher. The neutral bits can create from each plaintext pair a plaintext structure consisting of 2^k plaintext pairs that conform to the same differential transition $0\text{x}0001/4404 \to 0\text{x}0000/0400$, and thus reduce the number of chosen plaintexts by 2^k times. Obviously, the more plaintexts an attacker chooses, the higher the attack success rate will be. We set *n* to $2^{8.7}$ to ensure a success rate of more than 99%, so the data complexity is $2^{8.7}$ chosen-plaintext pairs. With a probabilistic neutral bit set consisting of bits 17, 19, 21, 23, 25 of the cipher states (with a probability very close to 1), $2^{13.7}$ plaintext pairs that conform same differential transition can be generated.

The computation complexity mainly consists of two parts. One is the partial decryption (line 6 in **Algorithm 2**), which is equivalent to

$$2^{13.7} + \frac{1}{15} \times 2^{16} \times 2^{13.7} \approx 2^{25.8} \tag{19}$$

times of 15-round SIMON32 encryption at most. The other is the cost of running the neural networks (line 7 in **Algorithm 2**). According to the experimental statistics, the time required to run the neural network of the 9-round distinguisher is about 15 times that required to run 15-round SIMON32 encryption. So, the cost of running the neural networks is equivalent to

$$15 \times 2^{16} \times 2^{13.7} \approx 2^{33.6} \tag{20}$$

times of 15-round SIMON32 encryption. It should be noted that this is the equivalent computational complexity only on the CPU. In reality, the neural network runs on GPU, and the time required is negligible compared with the encryption and decryption on CPU. In our experiments, a distinguishing attack was performed less than a minute on a PC with Intel i7-9700 CPU and Nvidia RTX 2080Ti GPU.

According to the principle of distinguishing attack, there are two types of misjudgment. One is to judge the real output as random data. The probability is

$$\alpha = \sum_{i=0}^{1} \mathrm{C}_{2^{13.7}}^{i} \cdot (TPR \times 2^{-10})^{i} \cdot (1 - TPR \times 2^{-10})^{2^{13.7}-i} \approx 0.008063, \tag{21}$$

where TPR = 0.5297 is the true positive rate of *N*9, and 2^{-10} is the differential transition probability of the truncated difference $0\text{x}0001/4404 \to 0\text{x}0000/0400$.

The other is to judge the random data as the real output. The probability is

$$\beta = 1 - \sum_{i=0}^{1} \mathrm{C}_{2^{13.7}}^{i} \cdot (FPR \times 2^{-32})^{i} \cdot (1 - FPR \times 2^{-32})^{2^{13.7}-i} \approx 9.248158 \times 10^{-13}, \tag{22}$$

where FPR = 1-TNR = 0.2319 is the false positive rate of *N*9, and 2^{-32} is the probability that a random ciphertext pair conforms to the truncated difference $0\text{x}0001/4404 \to 0\text{x}0000/0400$.

Therefore, the distinguishing advantage is

$$adv = 1 - \alpha - \beta \approx 0.991937. \tag{23}$$

3.4 Key-recovery Attack Against SIMON32

To show the utility of our neural distinguishers, we construct a partial-key recovery attack based on the 13-round distinguisher. The basic idea is that for each plaintext pair with the difference 0x0001/4404, we decrypt the resulting ciphertexts under all final subkeys and rank each partially decrypted ciphertext using our neural distinguisher. Then we combine scores returned for individual ciphertext pairs into a score for the key, and finally sort the keys in descending order according to their score. A brief description of attack steps is as follows and the details can be seen in **Algorithm 3**.

1. Generate *n* random chosen-plaintext pairs $P_0,\cdots,P_{n-1}$ such that the output difference of the first round is $\Delta = 0\text{x}0001/4404$. Obtain the corresponding ciphertext pairs $C_0,\cdots,C_{n-1}$.
2. For each last-round subkey *k*, decrypt the C_i under *k* to get $D_i^k = (D_i[k], D_i^{'}[k])$.
3. Use the 13-round differential distinguisher to get scores $v_i[k]$ for each partially decrypted ciphertext pair D_i^k .
4. For each *k*, combine the scores $v_i[k]$ into one score v_k.

Sort the keys in descending order according to their score v_k.

Algorithm 3 Key-Recovery Attack Against SIMON32

Input: Objective cipher *Oracle*, a 13-round neural distinguisher *N*, number of chosen plaintext pairs *n*.

Output: A list of candidate keys *CK*.

```
$(P, P') \leftarrow n$ random plaintext pairs that validate the difference 0x0001/4404 after one round of encryption.
$(C, C') \leftarrow Oracle(P, P')$
$CK \leftarrow \{\cdot\}$
for $k$ in Subkeys do
    for $i = 0$ to $n - 1$ do
        $(D_i[k], D_i^{'}[k]) \leftarrow \text{DecryptOneRound}((C_i, C_i^{'}), k)$
        $v_i[k] \leftarrow N(D_i[k], D_i^{'}[k])$
    end for
    $v_k \leftarrow \sum_{i=0}^{n-1} \log_2(v_i[k]/(1 - v_i[k]))$
    Append $(k, v_k)$ to CK.
end for
sort CK in descending order of $v_k$.
return CK
```

Similar to distinguishing attack, we first generate *n* random plaintext pairs that validate the difference 0x0001/4404 after one round of encryption. Then for each possible subkey, we use it to decrypt all ciphertexts with one round of SIMON32. Put the partially decrypted ciphertexts $(D_i[k], D_i^{'}[k])$ into the 13-round distinguisher and get a score $v_i[k]$ between 0 and 1. Then we use the real-random likelihood ratio

$$v_k = \sum_{i=0}^{n-1} \log_2(v_i[k]/(1 - v_i[k])) \tag{24}$$

to combine the scores of individual decrypted ciphertext pairs into a score for the key. It is proved to be effective in [12].

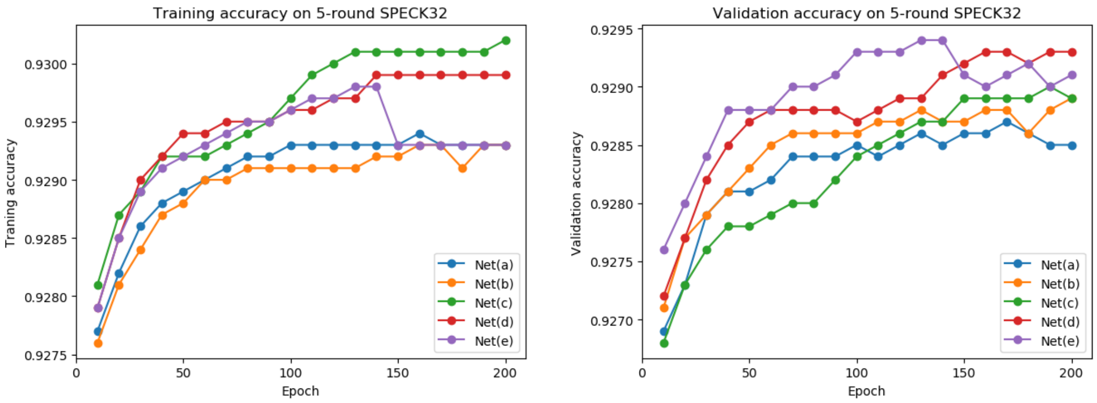
With the same neutral bits set, we conduct a lot of experiments, and finally set n to 2^8 to balance the success rate and complexity. The data complexity is 2^8 chosen-plaintext pairs. The computation complexity of operations on CPU is equivalent to

$$2^8+\frac{1}{15}\times 2^{16}\times 2^8\approx 2^{20.1} \tag{25}$$

times of 15-round SIMON32 encryption at most. As we have explained above, the time of running a neural network on GPU is negligible compared with operations on CPU, so it is not calculated here. We make a successful attack criterion, that is, if the correct key ranks among the top five in *CK*, we consider the attack to be successful. We repeat 100 key recovery attacks against different keys, and finally successfully recovered 23 times, so the experimental attack success rate is about 23%. The success rate obtained by different success criteria will of course be different, so this success rate is only used as a reference for the effectiveness of the key-recovery algorithm.

4. Performance of Different Network Structures

Using deep residual neural networks, Gohr [12] achieves better results than the best classical cryptanalysis on 11-round SPECK32. However, the network adopted by Gohr is not considered the best performing network in [17]. In order to study which structure choice is helpful to result in better distinguishers, we experiment with five residual network structures (showed in **Fig. 2**) on SIMON and SPECK respectively. We visualize part of the experimental data as **Fig. 3** and **Fig. 4**. On an RTX 2080 Ti graphics card, an epoch of training according to the basic training schedule for one of the ten-block networks takes about 110s at batch size 5000.



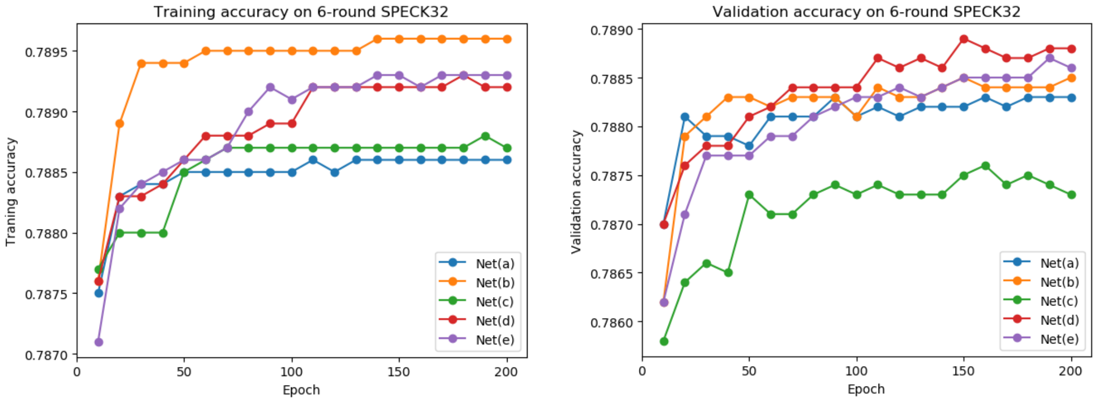


Fig. 3. Training five residual networks to distinguish 5-, 6-round SPECK32 output for the input difference 0x0040/0000 from random data

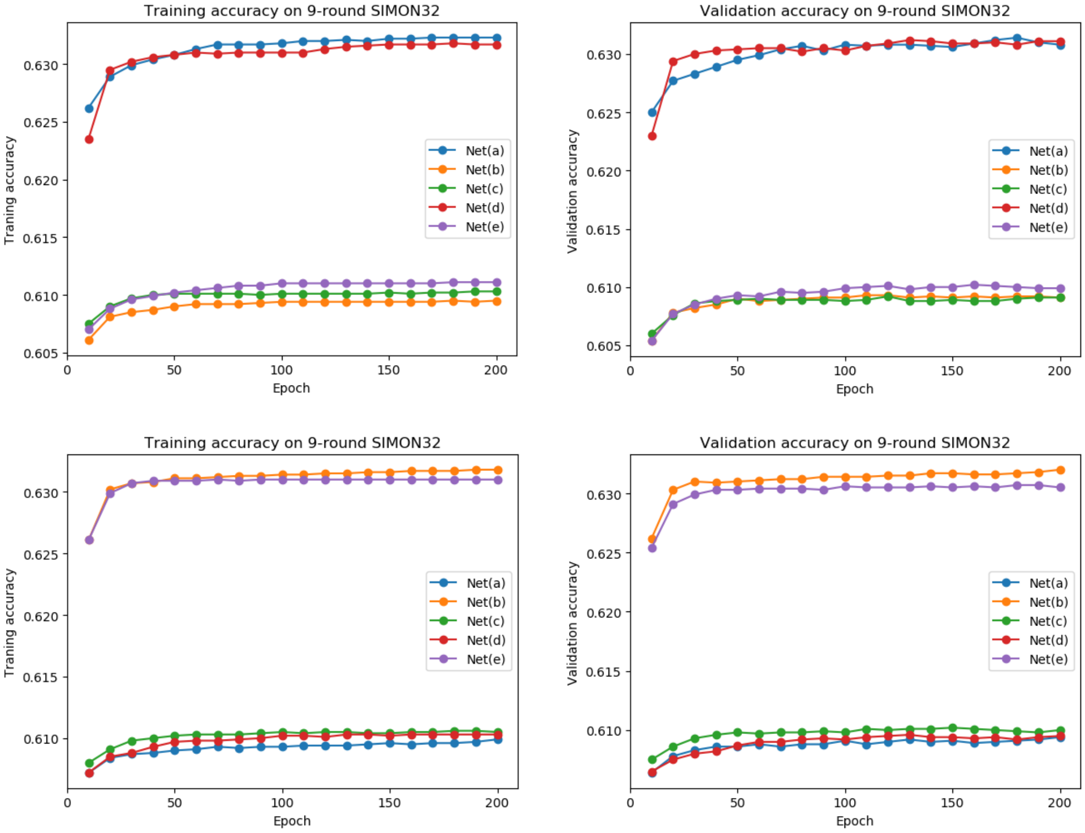


Fig. 4. Training five residual networks to distinguish 9-round SIMON32 output for the input difference δ from random data. In the top two figures, $\delta = 0x0000/0400$, and $\delta = 0x0000/0008$ in the bottom two

By analyzing the experimental data, we have several interesting findings, which may be of some reference value for the choice of neural network models in future studies. Of course, another discovery might be made by considering more disparate network structures such as multilayer perceptron (MLP) and recurrent neural network (RNN), but this would be a fairly complex discussion that could lead to another paper. Here we only focus on the influence of different residual network structures on the distinguishers. We summarize the following four rules.

1. Training accuracy may not always reflect the validation accuracy. As can be seen from the figures above, the model with the highest training accuracy is usually not the one with the highest validation accuracy, while the model with the lowest training accuracy does not necessarily have the worst performance on the validation data set. Training accuracy only reflects the performance of network on the training data, and only the validation accuracy can reflect the performance of network on new data. Therefore, the model with the highest validation accuracy should be selected. We speculate that paper [12] chose Net(c) because of its high training accuracy on the 5-round distinguisher for SPECK32/64. Based on our experimental results, it is reasonable to believe that other networks can achieve better results on the same task.
2. Different networks have significant effects on training results. As shown in **Fig. 4**, for the distinguishers of 9-round SIMON32, the highest network accuracy is nearly 0.022 higher than the lowest network accuracy, resulting in a reduction of about 8% chosen plaintexts in an attack. Besides, the convergence rates of training are obviously different.
3. For different ciphers and even for different rounds or different input differences of the same cipher, the optimal network may be different. Intuitively, we tend to think that a model will always fit a cipher and will not change greatly with the number of rounds and other minor changes. In fact, as we can see, for 5-round SPECK32, Net(e) achieves the highest accuracy in 140 epochs, while for 6-round SPECK32, Net(d) ends up with the best results. For 9-round SIMON32, Net(a) and Net(d) perform better with input difference 0x0000/0400, while Net(b) is the best with input difference 0*x*0000/0008. This inspires us not to hastily apply a certain model to other rounds or even other ciphers just because it performs well on individual *n*-round ciphers.
4. Different random training data have little effect on training results. For each distinguisher, we randomly generate different chosen plaintexts for 10 experiments, and there are only negligible differences in final accuracy (on the order of magnitude 10^{-4}).

Frankly, it might make more sense to discuss the differences in the results of taking different network structures, such as fully connected networks, but it is not pointless to focus on the details of residual networks. As mentioned above, small changes in the structure of the residual block can lead to a nonnegligible improvement in the accuracy of the model, thereby reducing the number of chosen plaintexts that are required to attack. This is a revelation that the improvement of the model does not necessarily require a large change in structure, and some small changes may also have considerable effects.

5. Conclusion

In this paper, we focus on evaluating the strength of the SIMON32/64 under the neural differential cryptanalysis and get better results than conventional differential cryptanalysis. We firstly prove theoretically that SIMON is a non-Markov cipher, which means that the analysis results based on the conventional differential cryptanalysis under the Markov assumption may be inaccurate. The neural differential cryptanalysis is independent of the Markov assumption, so we use ciphertext pairs generated by plaintext pairs with a fixed input difference to train 7-, 8-, 9-round neural differential distinguishers, and extend them to 11, 12, 13 rounds with a known differential transition. In order to prove the effectiveness of our neural differential distinguishers, we use them to perform the distinguishing attack and key-recovery attack on the 15-round SIMON32. Our results may not be an attack with the most rounds against SIMON, but they are the least costly at the same level of data and computation. Besides, our work demonstrates the effectiveness of deep learning assisted differential cryptanalysis on SIMON, and by further improving the neural network, better results are bound to be produced.

Furthermore, the effect of different residual network structures on the training results of neural distinguishers are studied and some observations are presented. It is hoped that our findings will provide some reference for future research.

To be sure, much further work remains to be done. For example, we only consider limited variations of the techniques used in [12], and do not discover significant effects. In the future, we plan to study which components of the network structure used are necessary to obtain good results and how to quickly and accurately determine the neural network for a certain cipher. These allow cryptanalysts to be more targeted when constructing neural networks. Whether neural differential analysis can still achieve better results on Markov cipher than conventional differential cryptanalysis is also worth studying, since there is currently no deep learning assisted differential cryptanalysis for Markov ciphers.

References

[1] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, pp. 3-72, 1991. Article (CrossRef Link)

[2] C. Blondeau and B. Gérard, "Multiple differential cryptanalysis: theory and practice," in *Proc. of the International Workshop on Fast Software Encryptio*, vol. 6733, pp. 35-54, 2011. Article (CrossRef Link)

[3] L. R. Knudsen, "Truncated and higher order differentials," in *Proc. of the Internaional Workshop on Fast Software Encryption*, vol. 1008, pp. 196-211, 1995. Article (CrossRef Link)

[4] L. R. Knudsen, "DEAL - a 128-bit block cipher," Department of Informatics, University of Bergen, Bergen, Norway, Feb. 1998. Article (CrossRef Link)

[5] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials," *Journal of Cryptology*, pp. 291-311, 2005. Article (CrossRef Link)

[6] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *Proc. of the International Conference on Security, Privacy, and Applied Cryptography Engineering*, vol. 10076, pp. 3-26, 2016. Article (CrossRef Link)

[7] S. Picek, I. P. Samiotis, J. Kim, A. Heuser, S. Bhasin, and A. Legay, "On the performance of convolutional neural networks for side-channel analysis," in *Proc. of the International Conference on Security, Privacy, and Applied Cryptography Engineering*, vol. 11348, pp. 157-176, 2018. Article (CrossRef Link)

[8] B. Timon, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 2, pp. 107-131, 2019. Article (CrossRef Link)

[9] M. Baryalai, J. Jang-Jaccard, and D. Liu, "Towards privacy-preserving classification in neural networks," in *Proc. of the 14th Annual Conference on Privacy, Security and Trust*, pp. 392-399, 2016. Article (CrossRef Link)

[10] M. Carbone, V. Conin, M. A. Cornélie, F. Dassance, G. Dufresne, C. Dumas, E. Prouff, and A. Venelli, "Deep learning to evaluate secure RSA implementations," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2, pp. 132-161, 2019. Article (CrossRef Link)

[11] X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, B. Li, and T. Wang, "DEEPSEC: a uniform platform for security analysis of deep learning model," in *Proc. of the 40th IEEE Symposium on Security and Privacy*, pp. 673-690, 2019. Article (CrossRef Link)

[12] A. Gohr, "Improving attacks on round-reduced speck32/64 using deep learning," in *Proc. of the Annual International Cryptology Conference*, vol. 11693, pp. 150-179, 2019. Article (CrossRef Link)

[13] A. Baksi, J. Breier, Y. Chen, and X. Dong, "Machine learning assisted differential distinguishers for lightweight ciphers," *The Cryptology ePrint Archive*, pp. 1-17, 2020. Article (CrossRef Link)

[14] J. So, "Deep learning-based cryptanalysis of lightweight block ciphers," *Security and Communication Networks*, vol. 2020, pp. 1-11, 2020. Article (CrossRef Link)

[15] B. Hou, Y. Li, H. Zhao, and B. Wu, "Linear attack on round-reduced DES using deep learning," in *Proc. of the 25th European Symposium on Research in Computer Security*, pp. 131-145, Sep. 2020. Article (CrossRef Link)

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016. Article (CrossRef Link)

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. of European Conference on Computer Vision*, vol. 9908, pp. 630-645, 2016. Article (CrossRef Link)

[18] S. Kölbl, G. Leander, and T. Tiessen, "Observations on the SIMON block cipher family," in *Proc. of the Annual Cryptology Conference*, vol. 9215, pp. 161-185, 2015. Article (CrossRef Link)

[19] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK families of lightweight block ciphers," *The Cryptology ePrint Archive*, vol. 404, 2013. Article (CrossRef Link)

[20] J. Alizadeh, H. AlKhzaimi, M. R. Aref, N. Bagheri, P. Gauravaram, A. Jumar, M. M. Lauridsen, S. K. Sanadhya, "Cryptanalysis of SIMON variants with connections," in *Proc. of the International Workshop on Radio Frequency Identification: Security and Privacy Issues*, vol. 8651, pp. 90-107, 2015. Article (CrossRef Link)

[21] A. Biryukov, A. Roy, and V. Velichkov, "Differential analysis of block ciphers SIMON and SPECK," in *Proc. of the International Workshop on Fast Software Encryption*, vol. 8540, pp. 546-570, 2015. Article (CrossRef Link)

[22] S. Sun, L. Hu, P. Wang, K. Qiao, X. Ma, and L. Song, "Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers," in *Proc. of the Advances in Cryptology - ASIACRYPT 2014*, vol. 8873, pp. 158-178, 2014. Article (CrossRef Link)

[23] Q. Wang, Z. Liu, K. Varici, Y. Sasaki, V. Rijmen, and Y. Todo, "Cryptanalysis of reduced-round SIMON32 and SIMON48," in *Proc. of the International Conference in India*, vol. 8885, pp. 143-160, 2014. Article (CrossRef Link)

[24] K. Qiao, L. Hu, and S. Sun, "Differential analysis on Simeck and SIMON with dynamic key-guessing techniques," in *Proc. of the International Conference on Information Systems Security and Privacy 2016*, vol. 691, pp. 64-85, 2017. Article (CrossRef Link)

[25] Z. Liu, Y. Li, and M. Wang, "Optimal differential trails in SIMON-like ciphers," *IACR Transactions on Symmetric Cryptology*, vol. 2017, no. 1, pp. 358-379, 2017. Article (CrossRef Link)

[26] F. Abed, E. List, S. Lucks, and J. Wenzel, "Differential cryptanalysis of round-reduced Simon and speck," in *Proc. of the International Conference on Fast Software Encryption*, vol. 8540, pp. 525-545, 2014. Article (CrossRef Link)

[27] X. Lai, J. L. Massey, and S. Murphy, "Markov ciphers and differential cryptanalysis," in *Proc. of the Workshop on Theory and Application of Cryptographic Techniques*, vol. 547, pp. 17-38, 1991. Article (CrossRef Link)

[28] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Cambridge, MA, USA: MIT Press, 2016.

[29] C. Francois, Deep learning with python, Greenwich, USA: Manning Publications, 2017.

[30] E. Biham and R. Chen, "Near-collisions of SHA-0," in *Proc. of the Annual International Cryptology Conference*, vol. 3152, pp. 290-305, 2014. Article (CrossRef Link)

Wenqiang Tian received his B.E. degree in cryptology from the PLA SSF Information Engineering University, Zhengzhou, China, in 2018. Now he is a master's candidate at the PLA SSF Information Engineering University. His main research interests include automatic cryptanalysis and intelligent cryptanalysis of symmetric ciphers.

Bin Hu received his Ph.D. degree in cryptology from the PLA SSF Information Engineering University, Zhengzhou, China, in 2008. He is currently a professor at the PLA SSF Information Engineering University. His research interests include the theory of information security, cipher design and cryptanalysis.