Improved Neural Aided Statistical Attack for Cryptanalysis

Yi Chen¹ and Hongbo Yu¹ *

¹ Department of Computer Science and Technology, Tsinghua University, P.R. China, chenyi19@mails.tsinghua.edu.cn
² Department of Computer Science and Technology, Tsinghua University, P.R. China, yuhongbo@mail.tsinghua.edu.cn

Abstract. At CRYPTO 2019, Gohr improved attacks on Speck32/64 using deep learning. In 2020, Chen et al. proposed a neural aided statistical attack that is more generic. Chen et’s attack is based on a statistical distinguisher that covers a prepended differential transition and a neural distinguisher. When the probability of the differential transition is $pq$, its impact on the data complexity is $O(p^{-2}q^{-2})$. In this paper, we propose an improved neural aided statistical attack based on a new concept named Homogeneous Set. Since partial random ciphertext pairs are filtered with the help of homogeneous sets, the differential transition’s impact on the data complexity is reduced to $O(p^{-1}q^{-2})$. As a demonstration, the improved neural aided statistical attack is applied to round-reduced Speck. And several better attacks are obtained.

Keywords: Cryptanalysis · Deep learning · Homogeneous set · Statistical attack · Speck families

1 Introduction

Background. Neural aided cryptanalysis is a refreshing cryptanalysis technique that exploits deep learning. It has attracted much attention from the last century. Rivest in [Ron91] reviewed various connections between machine learning and cryptography. Some possible directions of research in cryptanalytic applications of machine learning were also suggested. Greydanus proved that a simplified version of Enigma can be simulated by recurrent neural networks [Gre17]. At Crypto 2019, Gohr proposed a neural distinguisher and improved key recovery attack on 11-round Speck32/64 [Goh19]. This is the first work that successfully shows the capability of deep learning in conventional cryptanalysis. However, there are several constraints in the application of such an attack.

Inspired by Gohr’s work, Chen et al. [CY20] proposed a neural aided statistical attack (NASA) that is almost as generic as the differential cryptanalysis[BS91]. Its potential has been proven with applications to round-reduced Speck and DES. NASA is based on a new statistical distinguisher that covers a prepended differential transition and a neural distinguisher. Based on the analysis in [CY20], the data complexity of the NASA is mainly affected by the differential transition. When the probability of the differential transition is $p_0 = pq$, its impact on the data complexity is $O(p_0^{-2})$. This is a serious bottleneck when $p_0$ is very low.

Our work. In this paper, our core target is to reduce the data complexity of the NASA. To achieve the target, we introduce a new concept named Homogeneous Set which
summarizes a common phenomenon that exists in differential cryptanalysis. If one plaintext pair passes a given differential transition, we can generate multiple plaintext pairs based on this plaintext pair. These new plaintext pairs can also pass the differential transition simultaneously. In [ER04] and [BLT20], researchers both exploited such a phenomenon, although the concrete techniques for generating new plaintext pairs are different.

Based on the new concept, we propose an improved NASA. The prepended differential transition $\Delta P \rightarrow \Delta S_2$ with a probability of $p_0 = p \times q$ is divided into two parts: the first part $\Delta P \rightarrow \Delta S_1$ with a probability of $p$, and the second part $\Delta S_1 \rightarrow \Delta S_2$ with a probability of $q$. Then the impact of the probability of the prepended differential transition on the data complexity can be reduced from $O(p^{-2}q^{-2})$ to $O(p^{-1}q^{-2})$. In order to verify the superiority of the improved NASA, we apply it to round-reduced Speck[^BSS+15].

The summary of our attacks together with the previous best ones is shown in Table 1.

Table 1: Summary of key recovery attacks on round reduced Speck. DD: differential distinguisher. ND: neural distinguisher. SD: neural aided statistical distinguisher. CP: Chosen-Plaintext.

<table>
<thead>
<tr>
<th>cipher</th>
<th>Distinguisher</th>
<th>Rounds</th>
<th>Complexity</th>
<th>Data</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speck32/64</td>
<td>DD</td>
<td>11</td>
<td>$2^{46.7}$</td>
<td>$2^{40.1}$CP</td>
<td>[ALLW14]</td>
</tr>
<tr>
<td></td>
<td>DD</td>
<td>11</td>
<td>$2^{46}$</td>
<td>$2^{14}$CP</td>
<td>[Din14]</td>
</tr>
<tr>
<td></td>
<td>ND</td>
<td>11</td>
<td>$2^{18}$</td>
<td>$2^{14.5}$CP</td>
<td>[Goh19]</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>11</td>
<td>$2^{12.29}$</td>
<td>$2^{23.44}$CP</td>
<td>[CY20]</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>11</td>
<td>$2^{26.71}$</td>
<td>$2^{21.39}$CP</td>
<td>Section 5.1</td>
</tr>
<tr>
<td></td>
<td>DD</td>
<td>12</td>
<td>$2^{51}$</td>
<td>$2^{19}$CP</td>
<td>[Din14]</td>
</tr>
<tr>
<td></td>
<td>ND</td>
<td>12</td>
<td>-</td>
<td>-</td>
<td>[Goh19]</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>12</td>
<td>$2^{40.35}$</td>
<td>$2^{27.93}$CP</td>
<td>[CY20]</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>12</td>
<td>$2^{37.39}$</td>
<td>$2^{24.92}$CP</td>
<td>Section 5.2</td>
</tr>
<tr>
<td></td>
<td>DD</td>
<td>13</td>
<td>$2^{37}$</td>
<td>$2^{20}$CP</td>
<td>[Din14]</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>13</td>
<td>$2^{28}$</td>
<td>$2^{28.7}$CP</td>
<td>[CY20]</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>13</td>
<td>$2^{55.17}$</td>
<td>$2^{25.87}$CP</td>
<td>Section 5.3</td>
</tr>
<tr>
<td>Speck48/72</td>
<td>DD</td>
<td>12</td>
<td>$2^{43}$</td>
<td>$2^{41}$CP</td>
<td>[BRV14]</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>12</td>
<td>$2^{44.86}$</td>
<td>$2^{34.37}$CP</td>
<td>Section 6.1</td>
</tr>
<tr>
<td>Speck48/96</td>
<td>DD</td>
<td>12</td>
<td>$2^{43}$</td>
<td>$2^{43}$CP</td>
<td>[BRV14]</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>12</td>
<td>$2^{44.86}$</td>
<td>$2^{32.37}$CP</td>
<td>Section 6.2</td>
</tr>
</tbody>
</table>

1. Complexity is given in terms of the full decryption of the attacked cipher.
2. Gohr also provided an attack on 12-round Speck32/64, but the data, computation complexity were not presented in [Goh19].

Compared with that of Chen et's attacks on Round reduced Speck32/64, the data complexity of our improved NASA is lower. For Speck32/64 reduced to 11, 12, 13 rounds, our attacks require lower computation complexity than previous attacks. For Speck48/X reduced to 12 rounds, the computation complexity of our attack is similar to that of previous attacks, and the data complexity is significantly reduced.

The neural distinguisher can also affect the data complexity. As we know, the target of the training of neural networks solving classification problems is to obtain an optimal classification accuracy. But Chen et al. didn't explain whether this common training target is useful for reducing the data complexity. In this paper, we also present an analysis of this issue.

Organizations. In Section 2, we will review the NASA. Our improvement to the differential part and the improved NASA are proposed in Section 3. Several practical
experiments for the verification of homogeneous sets are provided in Section 4. Then we apply the improved NASA to round-reduced Speck for security analysis (Section 5, 6). At last, the analysis about the training of neural distinguishers is given in Section 7.

2 Review of Neural Aided Statistical Attack [CY20]

2.1 Overview

Consider a cipher reduced to \( h \) rounds, we first review the NASA by taking an attack with 1-round decryption as an example. Figure 1 summarizes the general scheme of the NASA.

\[
\begin{align*}
Y(C_0, C_1) &= \begin{cases} 
1, \text{ if } S_0 \oplus S_1 = \Delta S \\
0, \text{ if } S_0 \oplus S_1 \neq \Delta S 
\end{cases} \\
\text{where } Y \text{ is the sample label, and } \Delta S \text{ is the target difference of the intermediate state.}
\end{align*}
\]

\[
Pr(Y = 1 | (C_0, C_1)) = F(C_0, C_1), \quad Pr(Y = 1 | (C_0, C_1)) \in [0, 1]
\]

where \( F(\cdot) \) is the posterior probability estimation function learned by \( ND \). When \( F(C_0, C_1) > 0.5 \), the predicted label of the input sample is 1.

In the NASA setting, the adversary tests a subkey guess \( kg \) by

1. Randomly generate \( N \) plaintext pairs \((P_0^i, P_1^i)\), \( P_0^i \oplus P_1^i = \Delta P, i \in \{1, \cdots, N\} \).
2. Collect corresponding \( N \) ciphertext pairs.
3. Decrypt \( N \) ciphertext pairs with \( kg \).
4. Feed partially decrypted ciphertext pairs \((C_0^i, C_1^i), i \in \{1, \cdots, N\} \) into \( ND \).
5. Collect corresponding posterior probabilities
\[ Z_i = Pr(Y = 1 \mid (C_0^i, C_1^i)) \]

6. Calculate the following statistic \( T \)
\[ T = \sum_{i=1}^{N} \phi(Z_i), \quad \phi(Z_i) = \begin{cases} 1, & \text{if } Z_i > c \\ 0, & \text{if } Z_i \leq c \end{cases} \tag{3} \]

where \( c \) is a posterior probability threshold that is set in advance.

When \( T \) exceeds a decision threshold \( t \), \( kg \) is returned as a subkey candidate.

### 2.2 Distribution of the Statistic under Right and Wrong keys

In [CY20], Chen et al. presented a theoretical analysis about the data complexity \( N \) and the decision threshold \( t \). We mainly review conclusions related to our work.

When the posterior probability threshold \( c \) is set, the classification over a ciphertext pair is modeled as a Bernoulli experiment. The probability that \( Z > c \) holds is stable and can be estimated experimentally. According to the key recovery process, there are three situations:

1. Decrypting a positive sample with the right subkey. \( S_0 \oplus S_1 = \Delta S, kg = sk \).
2. Decrypting a positive sample with wrong subkeys. \( S_0 \oplus S_1 = \Delta S, kg \neq sk \).
3. Decrypting negative samples. \( S_0 \oplus S_1 \neq \Delta S \).

where \( sk \) is the right subkey. The probabilities of \( Z > c \) in these three situations are denoted as \( p_1, p_2, p_3 \) respectively.

Then the distributions of the statistic (formula 3) in these three situations are
\[
\begin{align*}
T_1 & \sim \mathcal{N}(\mu_1, \sigma_1), \mu_1 = N_1 \times p_1, \sigma_1 = \sqrt{N_1 \times p_1 \times (1 - p_1)} \\
T_2 & \sim \mathcal{N}(\mu_2, \sigma_2), \mu_2 = N_2 \times p_2, \sigma_2 = \sqrt{N_2 \times p_2 \times (1 - p_2)} \\
T_3 & \sim \mathcal{N}(\mu_3, \sigma_3), \mu_3 = N_3 \times p_3, \sigma_3 = \sqrt{N_3 \times p_3 \times (1 - p_3)} \tag{4}
\end{align*}
\]

if \( N_1, N_2, N_3 \) are high enough. \( \mathcal{N}(\mu, \sigma) \) is a normal distribution with mean \( \mu \) and standard deviation \( \sigma \), \( i \in \{1, 2, 3\} \). When the probability of the differential is \( p_0 \) and \( N \) samples are collected randomly, thus \( N_1 = N_2 = N \times p_0, N_3 = N(1 - p_0) \).

When the subkey guess is right, the distribution of the statistic is
\[ T_r = T_1 + T_3 \sim \mathcal{N}(\mu_r, \sigma_r) \tag{5} \]
\[ \mu_r = N(p_0 \times p_1 + (1 - p_0)p_3), \quad \sigma_r = \sqrt{N \times p_0 \times p_1 (1 - p_1) + N(1 - p_0)p_3 (1 - p_3)} \tag{6} \]

When the subkey guess is wrong, the distribution of the statistic is
\[ T_w = T_2 + T_3 \sim \mathcal{N}(\mu_w, \sigma_w) \tag{7} \]
\[ \mu_w = N(p_0 \times p_2 + (1 - p_0)p_3), \quad \sigma_w = \sqrt{N \times p_0 \times p_2 (1 - p_2) + N(1 - p_0)p_3 (1 - p_3)} \tag{8} \]

Since \( p_1 \neq p_2 \), these two distributions \( T_r, T_w \) are different. Then the distinguishing between two normal distributions provides a theoretical framework for calculating \( N \) and \( t \) (see Section 2.3).

**Remark 1.** The choice of \( p_2 \) is related to the target of the key recovery attack. For a cipher with a subkey size of \( L \), all the \( 2^L \) subkey guesses are divided into \( L + 1 \) subspaces according to the Hamming distance \( d_1 \) between \( sk \) and \( kg \). For each specific \( d_1 \), the probability that \( Z > c \) is denoted as \( p_{2d_1} \). If the adversary hopes that the Hamming distance between returned subkey guesses and \( sk \) doesn’t exceed \( d \), the final value of \( p_2 \) should be
\[ p_2 = \max \{p_{2d_1} \mid d_1 > d\} \tag{9} \]

More details about the estimation of \( p_2 \) can refer to [CY20].
2.3 Distinguishing between Two Normal Distributions

Consider two normal distributions: \( N(\mu_1, \sigma_1) \), and \( N(\mu_0, \sigma_0) \). A sample \( s \) is sampled from either \( N(\mu_1, \sigma_1) \) or \( N(\mu_0, \sigma_0) \). We have to decide if this sample is from \( N(\mu_1, \sigma_1) \) or \( N(\mu_0, \sigma_0) \). The decision is made by comparing the value \( s \) to some threshold \( t \). Without loss of generality, assume that \( \mu_1 > \mu_0 \). If \( s \geq t \), the decision is \( s \in N(\mu_1, \sigma_1) \). If \( s < t \), the decision is \( s \in N(\mu_0, \sigma_0) \). Then there are error probabilities of two types:

\[
\beta_1 = Pr \{ s \in N(\mu_0, \sigma_0) | s \in N(\mu_1, \sigma_1) \} \\
\beta_0 = Pr \{ s \in N(\mu_1, \sigma_1) | s \in N(\mu_0, \sigma_0) \} 
\]

(10)

Here a condition is given on \( \mu_1, \mu_0, \sigma_1, \sigma_0 \) such that the error probabilities are \( \beta_1 \) and \( \beta_0 \). The proof can refer to related research [Fel68, GHPS74].

**Proposition 1.** For the test to have error probabilities of at most \( \beta_1 \) and \( \beta_0 \), the parameters of the normal distribution \( N(\mu_1, \sigma_1) \) and \( N(\mu_0, \sigma_0) \) with \( \mu_1 > \mu_0 \) have to be such that

\[
\frac{z_{1-\beta_1} \times \sigma_1 + z_{1-\beta_0} \times \sigma_0}{\mu_1 - \mu_0} = 1
\]

(11)

where \( z_{1-\beta_1} \) and \( z_{1-\beta_0} \) are the quantiles of the standard normal distribution.

2.4 Data Complexity

According to Proposition 1, Chen et al. presented the final conclusions about the data complexity \( N \) and the decision threshold \( t \):

\[
a_1 = p_1(1 - p_1), \quad a_2 = p_2(1 - p_2), \quad a_3 = p_3(1 - p_3)
\]

(12)

\[
\sqrt{N} = \frac{z_{1-\beta_1} \times \sqrt{p_0 \times a_1 + (1 - p_0)a_3} + z_{1-\beta_w} \times \sqrt{p_0 \times a_2 + (1 - p_0)a_3}}{(1 - p_0) \times p_0}
\]

(13)

\[
t = \mu_r - z_{1-\beta} \times \sigma_r = \mu_w + z_{1-\beta_w} \times \sigma_w
\]

(14)

The impact of \( p_0 \) on the data complexity is \( O(p_0^{-2}) \). The impact of \( p_3 \) on the data complexity is \( O(p_3) \). The impact of \( p_1, p_2 \) on the data complexity is \( O((p_1 - p_2)^{-2}) \).

3 Improved Neural Aided Statistical Attack

The impact of \( p_0 \) on the data complexity is \( O(p_0^{-2}) \), which is one of the main bottlenecks for the raw NASA [CY20]. We will improve it by reducing the impact of the differential.

3.1 Overview

Similarly, we also take the key recovery attack with 1-round decryption as an example. Figure 2 summarizes the scheme of the improved NASA.

The core idea is dividing the differential transition into two parts: \( \Delta P \rightarrow \Delta S_1 \) with a probability of \( p \), \( \Delta S_1 \rightarrow \Delta S_2 \) with a probability of \( q \). The statistical distinguisher only covers the second part.

When we perform a key recovery attack using the statistical distinguisher, the data complexity is calculated based on \( q \) and the neural distinguisher. Let’s denote the data complexity as \( N_1 \). Now, the core task is collecting \( N_1 \) samples that satisfy the first differential transition \( \Delta P \rightarrow \Delta S_1 \). To solve this task, we propose a concept named Homogeneous Set that can be implemented via various techniques.
Definition 1. Consider a differential transition $\Delta P \rightarrow \Delta S_1$. A homogeneous set is a special set $S$ consisting of $M$ plaintext pairs

$$S = \{(P_0^0, P_0^1), \ldots, (P_M^0, P_M^1)\}, \quad P_i^0 \oplus P_i^1 = \Delta P, \quad i \in \{1, \ldots, M\} \quad (15)$$

Without loss of generality, we can assume that the $M$ plaintext pairs are derived from $(P_0^0, P_0^1)$ using a certain rule. After encryption, these $M$ plaintext pairs can pass the differential transition together in a high probability if $(P_0^0, P_0^1)$ passes the differential transition.

In [ER04], neutral bits were used to generate multiple messages that can pass a differential transition simultaneously. In [BLT20], authors divided a round reduced cipher into two independent sub-ciphers for generating multiple homogeneous plaintext pairs. Maybe there are more techniques that could be used to generate homogeneous sets. Any applicable techniques for generating homogeneous sets can be applied to our improved NASA. Thus we propose such a unified concept for introducing our work.

If the homogeneous set passes the first differential transition, we denote it as a valid homogeneous set. Or we denote it as an invalid homogeneous set. Then the target of gathering $N_1$ samples becomes gathering enough valid homogeneous sets. Now, the key recovery contains two following steps:

1. Data Collection:
   
   (a) Randomly generate a plaintext pair $(P_0^0, P_1^0), P_0^0 \oplus P_1^0 = \Delta P$.
   
   (b) Generate a homogeneous set $S$ based on $(P_0^0, P_1^0)$.
   
   (c) Test whether $S$ is a valid homogeneous set.
   
   (d) Repeat three steps above until enough valid homogeneous sets are saved.

2. Perform the key recovery with gathered $N_1$ samples.

Since the probability that $(P_0^0, P_1^0)$ passes the first differential $\Delta P \rightarrow \Delta S_1$ is $p$, intuition tells us that such a strategy can reduce the differential’s impact on the total data complexity from $O(p^{-2}q^{-2})$ to $O(p^{-1}q^{-2})$. We will present a formal theoretical analysis about the total data complexity later. Now, we wonder how to identify valid homogeneous sets.
3.2 Identify the Valid Homogeneous Set

The valid homogeneous set can be identified by performing the NASA. Let’s take the identification with 1-round decryption as an example, the process is

1. Generate a homogeneous set $S$ randomly.
2. Encrypt $S$ with the attacked cipher.
3. Collect corresponding ciphertext pairs $(C_0^i, C_1^i), i \in \{1, \cdots, M\}$.
4. Decrypt $M$ ciphertext pairs with a subkey guess $kg$.
5. Feed partially decrypted ciphertext pairs into the neural distinguisher $ND$.
6. Collect the neural distinguisher’s outputs and calculate the statistic $T$ (formula (3)).

3.2.1 Distribution of the statistic under valid homogeneous sets

The probability of the differential $\Delta S_1 \rightarrow \Delta S_2$ is $q$. Then there are about $M \times q$ positive samples and $M \times (1-q)$ negative samples.

When $kg$ is the right subkey, the distribution of the statistic (formula (3)) is

$$T_{V_i} = T_1 + T_3 \sim \mathcal{N}(\mu_{V_1}, \sigma_{V_1})$$

$$\mu_{V_1} = M(qp_1 + (1-q)p_3), \quad \sigma_{V_1} = \sqrt{Mqp_1(1-p_1) + M(1-q)p_3(1-p_3)}$$

If $kg$ is a wrong subkey guess, the distribution of the statistic (formula (3)) is

$$T_{V_0} = T_2 + T_3 \sim \mathcal{N}(\mu_{V_0}, \sigma_{V_0})$$

$$\mu_{V_0} = M(qp_2 + (1-q)p_3), \quad \sigma_{V_0} = \sqrt{Mqp_2(1-p_2) + M(1-q)p_3(1-p_3)}$$

3.2.2 Distribution of the statistic under invalid homogeneous sets

When $S$ is an invalid homogeneous set, the distribution of the statistic (formula (3)) is

$$T_I = T_3 \sim \mathcal{N}(\mu_I, \sigma_I)$$

$$\mu_I = Mp_3, \quad \sigma_I = \sqrt{Mp_3(1-p_3)}$$

3.2.3 Identify valid homogeneous sets by counting surviving subkey guesses

No matter whether the key guess $kg$ is right or wrong, distributions of the statistic under valid and invalid homogeneous sets are different. This finding can be used to identify valid homogeneous sets.

For convenience, we can denote $p_1, p_2$ as the same parameter $p_V$. Then $T_{V_1}/T_{V_0}$ can be represented in the same form

$$T_V \sim \mathcal{N}(\mu_V, \sigma_V)$$

$$\mu_V = M(qp_V + (1-q)p_3), \quad \sigma_V = \sqrt{Mqp_V(1-p_V) + M(1-q)p_3(1-p_3)}$$

According to Proposition 1 in Section 2.3, the condition for distinguishing $T_V$ and $T_I$ is

$$\frac{z_{1-\beta_V} \times \sigma_V + z_{1-\beta_I} \times \sigma_I}{\mu_V - \mu_I} = 1$$

(22)
where

\[
\beta_V = Pr\{s \in N(\mu_I, \sigma_I) \mid s \in N(\mu_V, \sigma_V)\} \\
\beta_I = Pr\{s \in N(\mu_V, \sigma_V) \mid s \in N(\mu_I, \sigma_I)\}
\] (23)

For invalid homogeneous sets, the maximum probability that subkey guesses survive the NASA is \(\beta_I\). Divide the whole subkey space into several subspaces according to the Hamming distance \(d_1\) between the right subkey \(sk\) and key guess \(kg\). For valid homogeneous sets, the minimum probability that subkey guesses in the target subspace (related to \(p_V\)) survive the NASA is \(1 - \beta_V\).

Let \(K\) denote the set of all possible subkey guesses

\[K = K_R + K_W\]

where \(K_R\) is the set of subkey guesses in the target subspace, and \(K_W\) is the set of subkey guesses in other subspaces. Then the lower bound of the number of surviving subkeys is \(|K_R| \times (1 - \beta_V)\) when \(S\) is a valid homogeneous set. The upper bound of the number of surviving subkeys is \(|K| \times \beta_I\) when \(S\) is an invalid homogeneous set.

By setting two proper error probabilities \(\beta_V, \beta_I\), we can ensure the following condition always holds

\[|K_R| \times (1 - \beta_V) >> |K| \times \beta_I\] (24)

Then we can set another decision threshold \(t_S\) that satisfies the condition

\[|K_R| \times (1 - \beta_V) \geq t_S >> |K| \times \beta_I\] (25)

It’s expected that we can always identify valid homogeneous sets by comparing the number of surviving subkey guesses with \(t_S\). Algorithm 1 summarizes the concrete process of identifying valid homogeneous sets.

**Algorithm 1 Identify valid homogeneous sets**

**Input:** a homogeneous set \(S\) with a size of \(M\) (formula 23);
- a statistical distinguisher that covers a differential and a neural distinguisher \(ND\);
- the posterior probability threshold \(c\);
- a decision threshold \(t_M\) for filtering subkey guesses;
- a decision threshold \(t_S\) for identifying valid homogeneous sets.

**Output:** the classification of the homogeneous set \(S\).

1: Encrypt \(M\) plaintext pairs of the homogeneous set \(S\) and collect the ciphertexts;
2: Initialize a counter \(cp \leftarrow 0\);
3: for each possible subkey guess \(kg\) do
4: Decrypt \(M\) ciphertext pairs with \(kg\);
5: Feed partially decrypted ciphertext pairs into the neural distinguisher \(ND\);
6: Save the outputs of the neural distinguisher, \(Z_i, i \in \{1, \sim, M\}\);
7: Count the number of \(Z_i > c\), and denote it as \(T_M\);
8: if \(T_M > t_M\) then
9: \(cp \leftarrow cp + 1\);
10: end if
11: end for
12: if \(cp \geq t_S\) then
13: Return 1 (\(S\) is a valid homogeneous set).
14: else
15: Return 0 (\(S\) is an invalid homogeneous set).
16: end if
3.2.4 Further analysis about $p_V$

According to formula (22), the lower bound of the size of the homogeneous set $M$ and the decision threshold $t_M$ are

$$a_V = p_V(1 - p_V), \quad a_3 = p_3(1 - p_3)$$ (26)

$$\sqrt{M} = \frac{z_{1-\beta_V} \times \sqrt{q a_V + (1 - q)a_3} + z_{1-\beta_I} \times \sqrt{a_3}}{(p_V - p_3) \times q}$$ (27)

$$t_M = \mu_V - z_{1-\beta_V} \sigma_V = \mu_I + z_{1-\beta_I} \sigma_I$$ (28)

When $p_V$ increases, $M$ will decrease (see the following proof).

Proof.

$$\sqrt{M} = \frac{z_{1-\beta_V} \times \sqrt{q p_V(1 - p_V) + (1 - q)a_3} + z_{1-\beta_I} \times \sqrt{a_3}}{(p_V - p_3) \times q}$$

$$= \frac{z_{1-\beta_V} \times \sqrt{q(1 - p_V) + \frac{(1-q)a_3}{p_V} + \frac{z_{1-\beta_I} \times \sqrt{a_3}}{p_V}}}{(\sqrt{p_V} - \frac{a_3}{\sqrt{p_V}}) \times q}$$

If $p_V$ increases, the numerator will decrease and the denominator will increase. Then $M$ will decrease.

When the Hamming distance $d_1$ between the right subkey $sk$ and subkey guess $kg$ increases, Chen et al.[CY20] has proved that $p_{2|d_1}$ will decrease in general. But the number of subkey guesses in the subspace may increase sharply when $d$ increases ($d \leq \frac{L}{2}$, $L$ is the subkey size). Thus the choice of $p_V$ is a trade-off. As long as the condition (formula (24)) holds, we advise $p_V = p_{2|d_1}$ where $d_1$ should be as small as possible.

3.3 Key Recovery

Now we can summarize the general model (Algorithm 2) of our improved NASA.

**Algorithm 2** Model of the improved neural aided statistical attack

**Input:** the first differential $\Delta P \rightarrow \Delta S_1$, the second differential $\Delta S_1 \rightarrow \Delta S_2$;
- a neural distinguisher $ND$ built on $\Delta S_2$, the posterior probability threshold $c$;
- the size of homogeneous sets $M$, the number of samples for key recovery $N_1$;
- the decision threshold $t$ for filtering wrong subkey guesses.

**Output:** all the possible subkey guesses.

1. Collect $\left\lceil \frac{N_1}{M} \right\rceil$ valid homogeneous sets using Algorithm 1(Section 3.2).
2. Pick $N_1$ samples from valid homogeneous sets randomly
3. for each subkey guess $kg$ do
4. Decrypt $N_1$ samples with $kg$.
5. Feed partially decrypted ciphertext pairs into $ND$.
6. Collect corresponding posterior probabilities $Z_i, i \in \{1, \cdots, N_1\}$.
7. Count the number of $Z_i > c$ and denote it as $tp$.
8. if $tp \geq t$ then
9. save $kg$ as a subkey candidate.
10. end if
11. end for
12. Test all the subkey candidates against a necessary number of plaintext-ciphertext pairs according to the unicity distance for the attacked cipher.
Algorithm 2 is the general model. If the key recovery only involves 1-round decryption, the second step can be further simplified. Since the process of finding valid homogeneous sets has already filtered wrong subkey guesses.

Before we analyze the total data complexity, there is one remaining problem. In the next, we will introduce one method for generating homogeneous sets.

### 3.4 Generate Homogeneous Sets Using Probabilistic Neutral Bits

Probabilistic neutral bit (PNB) is a typical technique for generating homogeneous sets. PNB is a generalized concept of the neutral bit [ER04], which is proposed for dividing the key space of stream ciphers [AFK+08].

The initial definition of the PNB is proposed for a specific scenario, which can’t be directly applied to generate homogeneous sets. Thus we modify its definition so that it can be used to generate homogeneous sets. For convenience, we also make a little change about the initial form of the neutrality measure.

**Definition 2.** Consider a differential $P_0 \oplus P_1 = \Delta P \rightarrow S_{1,0} \oplus S_{1,1} = \Delta S_1$. Encrypt $P_0 \oplus (1 << j)$, $P_1 \oplus (1 << j)$ and denote the difference between the resulting ciphertext pair as $\Delta S_{\text{new}}$. The probability that $\Delta S_{\text{new}}$ conforms to $\Delta S_1$ is the neutrality measure of the $j$-th bit of $P_0, P_1$.

**Algorithm 3**

**Computing probabilistic neutral bits**

**Input:** Number of samples $X$, the differential $\Delta P \rightarrow \Delta S_1$, the index $j$  
**Output:** the neutrality $\gamma_j$ of the $j$-th bit.

1. Initialize two counters $cp1 \leftarrow 0$, $cp2 \leftarrow 0$
2. for $i = 1$ to $X$ do
3. Generate a random pair, $(P_0, P_1)\|P_0 \oplus P_1 = \Delta P$
4. if $Enc(P_0) \oplus Enc(P_1) = \Delta S_1$ then
5. $cp1 \leftarrow cp1 + 1$
6. if $Enc(P_0 \oplus (1 << j)) \oplus Enc(P_1 \oplus (1 << j)) = \Delta S_1$ then
7. $cp2 \leftarrow cp2 + 1$
8. end if
9. end if
10. end for
11. $\gamma_j = cp2/cp1$

Let $\gamma_j$ denote the neutrality of the $j$-th bit, which can be experimentally estimated using Algorithm 3. A homogeneous set $S$ consisting of $2^m$ plaintext pairs can be created from a set $B$ that contains $m$ PNBs. Then the probability that $S$ is a valid homogeneous set is $p \prod_{j \in B} \gamma_j$ (29), where $p$ is the probability of the differential transition $\Delta P \rightarrow \Delta S_1$.

### 3.5 Data Complexity and the Impact of the Differential

According to the model of the improved NASA (Algorithm 2), the data complexity is related to three factors: the number of samples for recovering key $N_1$, the size of homogeneous sets $M$ (formula (27)), and the probability (formula (29)) that a homogeneous set is a valid homogeneous set. The value of $N_1$ can be calculated according to formula (13).
The simplest method for generating $N_1$ samples is gathering $\lceil \frac{N_1}{M} \rceil$ valid homogeneous sets by running Algorithm 1 many times. Thus the total data complexity $N$ of the improved NASA is

$$N = \left\lceil \frac{N_1}{M} \right\rceil \times M \times \left( p \times \prod_{j \in B} \gamma_j \right)^{-1} \quad (30)$$

where $M$ is present in formula (27). The impact of $q$ on $N_1$ is $O(q^{-2})$. As for $M$, we have

$$\sqrt{M} = \frac{z_{1-\beta_V} \times \sqrt{q a_V + (1-q) a_3} + z_{1-\beta_I} \times \sqrt{a_3}}{(p_V - p_1) \times q} \times q$$

$$\Rightarrow M \propto q^{-2} |a_3 + a_2^2 a_3 + q(a_V - a_3) + 2a_6 \sqrt{a_3} | q a_V + (1-q)a_3|$$

where $a_4 = \frac{z_{1-\beta_V}}{z_{1-\beta_I}}$. Thus the impact of $q$ on $M$ is also $O(q^{-2})$. Then the impact of $q$ on $\lceil \frac{N_1}{M} \rceil$ is $O(1)$. Finally, we have

$$N \propto M \times p^{-1}$$

Compared with the initial NASA [CY20], the differential’s impact on the total data complexity $N$ is reduced from $O(p^{-2} q^{-2})$ to $O(p^{-1} q^{-2})$.

**A little trick for reducing the data complexity.** If there are $m$ high PNBs (eg. $\gamma \approx 1$) in the differential $\Delta P \rightarrow \Delta S_1$, we can reduce the required number of valid homogeneous sets when $2^m \gg M$. Once a valid homogeneous set is found, we can generate another $2^m - M$ samples directly. If $2^m \geq N_1$, we just need to find 1 valid homogeneous set instead of $\lceil \frac{N_1}{M} \rceil$ valid homogeneous sets.

### 4 Verification about Homogeneous Sets

There are two important factors that will affect our improved NASA. The first is the success rate of identifying valid homogeneous sets. The second is the success rate of filtering invalid homogeneous sets. To verify our analysis about the condition (formula (24)) for identifying valid homogeneous sets, two experiments are performed with the help of a 9-round Speck32/64.

A 1-round differential transition $\Delta S_1 = (0x2800, 0x10) \rightarrow \Delta S_2 = (0x40, 0)$ with a probability of $q = 2^{-2}$ is followed by a 7-round student distinguisher $ND^2_7$ that is also provided in [CY20]. A student distinguisher is built on partial ciphertext bits, and the subscript set of selected ciphertext bits is denoted as $\Gamma$. Table 2 shows the estimation of $p_{2|d_1}$ of $ND^2_7$, which is also provided in [CY20].

<table>
<thead>
<tr>
<th>$d_1$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7 $\sim$ 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.3576</td>
<td>0.3230</td>
<td>0.3036</td>
<td>0.2940</td>
<td>0.2893</td>
<td>0.2873</td>
<td>0.2866</td>
</tr>
</tbody>
</table>

When $p_V$ decreases, the value of $M$ will increase. In these two experiments, we adopt the following settings

$$c = 0.55, \quad d = 1, \quad p_V = p_{2|d_1} = d, \quad p_3 = 0.2863, \quad \beta_V = 0.1, \quad \beta_I = 2^{-8} \quad (31)$$

It means that the Hamming distance between subkey guesses of the target subspace $K_R$ and the right subkey is $d = 1$. Besides, we have

$$M = 37941 \approx 2^{15.21}, \quad t_M = 11096 \quad (32)$$
Since only 8 subkey bits $sk_i[7 \sim 0]$ are considered, the lower bound of the number of surviving subkey guesses for valid homogeneous sets is:

$$|K_R| \times (1 - \beta_V) = 8 \times (1 - 0.1) = 7.2 \approx 8$$

And the upper bound of the number of surviving subkey guesses for invalid homogeneous sets is:

$$|K| \times \beta_I = 2^8 \times 2^{-8} = 1$$

Let the decision threshold for distinguishing valid homogeneous sets be $t_S = 8$.

Identification of simulated valid homogeneous sets. We have performed this experiment 100 times with $M = 37941$. For each experiment, we randomly generate $M$ plaintext pairs $(P_i^0, P_i^1)|P_i^0 \oplus P_i^1 = \Delta S_i, i \in \{1, \cdots, M}\}$ to form a simulated valid homogeneous set $S$, and then run Algorithm 2.

The final results are:

1. We have successfully identified the valid homogeneous set in 100 experiments.
2. In 5 experiments, the number of surviving subkey guesses is 8 that is the theoretic lower bound.

Identification of simulated invalid homogeneous sets. We have performed this experiment 100 times with $M = 37941$. For each experiment, we randomly generate $M$ plaintext pairs $(P_i^0, P_i^1)|P_i^0 \oplus P_i^1 \neq \Delta S_i, i \in \{1, \cdots, M\}$ to form a simulated invalid homogeneous set $S$, and then run Algorithm 2.

The final results are:

1. The invalid homogeneous set is successfully identified in 98 experiments.
2. The average number of surviving subkey guesses in 100 experiments is 0.95 that is smaller than the theoretic upper bound.
3. In 72 experiments, the number of surviving subkey guesses is 0. In 9 experiments, the number of surviving subkey guesses is 1.

Analysis about two experiments. The reason why we perform experiments with simulated homogeneous sets is that the basic NASA requires that all the plaintext pairs are randomly sampled. Besides, the condition for distinguishing between valid homogeneous sets and invalid ones is also based on this assumption. The two experiments above can prove our analysis about homogeneous sets. And the derived condition is also applicable.

When the concrete implementation of homogeneous sets decreases the sampling randomness, it will result in a negative influence on the key recovery. But based on our practical experiments (Section 5) of attacking 11-round Speck32/64, we know the negative influence doesn’t break the correctness of our improved NASA.

5 Key Recovery Attacks on Round Reduced Speck32/64

To prove the superiority of our improved NASA, we perform key recovery attacks on Speck32/64 reduced to 11, 12, 13 rounds. In these three attacks, we adopt the same two 1-round differential transitions

$$\Delta P = (0x211, 0x0a04) \xrightarrow{p} \Delta S_1 = (0x2800, 0x10) \xrightarrow{q} \Delta S_2 = (0x40, 0)$$

where $p = 2^{-4}$ and $q = 2^{-2}$.

Let $X = 2^{24}$ and run Algorithm 3 for each plaintext bit, we have estimated the neutrality of each plaintext bit for Speck32/64. Table 3 shows the estimation of each plaintext bit’s neutrality.
Table 3: Neutrality $\gamma$ of each plaintext bit for (0x211, 0xa04) $\rightarrow$ (0x2800, 0x10)

<table>
<thead>
<tr>
<th>Index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma$</td>
<td>0.998</td>
<td>0.996</td>
<td>0.992</td>
<td>0.984</td>
<td>0.969</td>
<td>0.937</td>
<td>0.875</td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.75</td>
<td>0.5</td>
<td>1.5</td>
<td>1.5</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Index</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.998</td>
</tr>
<tr>
<td>Index</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.996</td>
<td>0</td>
<td>0.992</td>
<td>0.984</td>
<td>0.969</td>
<td>0.937</td>
<td>0.875</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Based on $\Delta S_2 = (0x40, 0)$, we have trained five teacher distinguishers $ND^1_5, ND^5_5, ND^6_5, ND^7_5, ND^8_5$ using the same training pipeline presented in [Goh19]. A teacher distinguisher is built on the complete ciphertext pair instead of partial bits. Table 4 shows the estimation of $p_{2|d_1}$ of these five teacher distinguishers.

Table 4: The estimation of $p_{2|d_1}$ of five teacher distinguishers against round reduced Speck32/64. For $ND^1_5, ND^5_5, ND^6_5, ND^7_5, c = 0.55$. For $ND^8_5, c = 0.5$.

<table>
<thead>
<tr>
<th>$ND^1_5$</th>
<th>$d_1$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.995</td>
<td>0.5065</td>
<td>0.2815</td>
<td>0.1686</td>
<td>0.1088</td>
<td>0.0735</td>
</tr>
<tr>
<td>$d_1$</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.0521</td>
<td>0.039</td>
<td>0.0301</td>
<td>0.0239</td>
<td>0.0198</td>
<td>0.0169</td>
</tr>
<tr>
<td>$d_1$</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.0146</td>
<td>0.0129</td>
<td>0.0117</td>
<td>0.0107</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>$ND^5_5$</td>
<td>$d_1$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.8889</td>
<td>0.5151</td>
<td>0.3213</td>
<td>0.2168</td>
<td>0.1556</td>
<td>0.1189</td>
</tr>
<tr>
<td>$d_1$</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.0956</td>
<td>0.08</td>
<td>0.0694</td>
<td>0.0617</td>
<td>0.056</td>
<td>0.0516</td>
</tr>
<tr>
<td>$d_1$</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.0483</td>
<td>0.0456</td>
<td>0.0436</td>
<td>0.0419</td>
<td>0.0407</td>
<td></td>
</tr>
<tr>
<td>$ND^6_5$</td>
<td>$d_1$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.6785</td>
<td>0.4429</td>
<td>0.3135</td>
<td>0.2384</td>
<td>0.1947</td>
<td>0.1684</td>
</tr>
<tr>
<td>$d_1$</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.1518</td>
<td>0.1408</td>
<td>0.1334</td>
<td>0.1283</td>
<td>0.1247</td>
<td>0.1219</td>
</tr>
<tr>
<td>$d_1$</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.1201</td>
<td>0.1183</td>
<td>0.117</td>
<td>0.1171</td>
<td>0.1188</td>
<td></td>
</tr>
<tr>
<td>$ND^7_5$</td>
<td>$d_1$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.4183</td>
<td>0.3369</td>
<td>0.2884</td>
<td>0.2607</td>
<td>0.2442</td>
<td>0.234</td>
</tr>
<tr>
<td>$d_1$</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.2276</td>
<td>0.2236</td>
<td>0.2211</td>
<td>0.2193</td>
<td>0.2183</td>
<td>0.2175</td>
</tr>
<tr>
<td>$d_1$</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.2172</td>
<td>0.2167</td>
<td>0.2159</td>
<td>0.2161</td>
<td>0.209</td>
<td></td>
</tr>
<tr>
<td>$ND^8_5$</td>
<td>$d_1$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.5184</td>
<td>0.5056</td>
<td>0.4993</td>
<td>0.4957</td>
<td>0.4939</td>
<td>0.4927</td>
</tr>
<tr>
<td>$d_1$</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.4925</td>
<td>0.4918</td>
<td>0.4917</td>
<td>0.4914</td>
<td>0.4913</td>
<td>0.4913</td>
</tr>
<tr>
<td>$d_1$</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_{2</td>
<td>d_1}$</td>
<td>0.4911</td>
<td>0.4913</td>
<td>0.4914</td>
<td>0.491</td>
<td>0.4914</td>
<td></td>
</tr>
</tbody>
</table>

According to the two-stage training algorithm proposed in [CY20], we have also trained three student distinguishers $ND^5_5, ND^6_5, ND^7_5$. It’s worth noticing that the subscript set for training student distinguishers is $\Gamma = \{30 \sim 23, 14 \sim 7\}$. Let $c = 0.55$, Table 5 summarizes the estimation of $p_{2|d_1}$ of these three student distinguishers.
Table 5: The estimation of \(p_{2|d_1}\) of \(ND^c\) when \(\Gamma = \{30 \sim 23, 14 \sim 7\}, c = 0.55\)

<table>
<thead>
<tr>
<th>(ND^c)</th>
<th>(d_1)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6 (\sim) 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>(p_{2</td>
<td>d_1})</td>
<td>0.7192</td>
<td>0.5498</td>
<td>0.4342</td>
<td>0.3566</td>
<td>0.3048</td>
<td>0.2708</td>
<td>(\leq 0.2486)</td>
</tr>
<tr>
<td>(ND^c_6)</td>
<td>(d_1)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6 (\sim) 8</td>
</tr>
<tr>
<td>(p_{2</td>
<td>d_1})</td>
<td>0.5132</td>
<td>0.4057</td>
<td>0.3402</td>
<td>0.3025</td>
<td>0.2817</td>
<td>0.2706</td>
<td>(\leq 0.265)</td>
</tr>
<tr>
<td>(ND^c_7)</td>
<td>(d_1)</td>
<td>0.3576</td>
<td>0.3230</td>
<td>0.3036</td>
<td>0.2940</td>
<td>0.2893</td>
<td>0.2873</td>
<td>(\leq 0.2866)</td>
</tr>
</tbody>
</table>

According to the definition of \(p_1\), we know \(p_1 = p_{2|d_1=0}\). We have also estimated \(p_3\) for each neural distinguisher. Table 6 summarizes the estimation of \(p_3\) of these eight neural distinguishers.

Table 6: The estimation of \(p_3\) of eight neural distinguishers. For \(ND^c_8\), \(c = 0.5\). For other \(7\) neural distinguishers, \(c = 0.55\).

<table>
<thead>
<tr>
<th>(ND)</th>
<th>(ND^c_1)</th>
<th>(ND^c_4)</th>
<th>(ND^c_5)</th>
<th>(ND^c_6)</th>
<th>(ND^c_7)</th>
<th>(p_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0069</td>
<td>0.0384</td>
<td>0.1161</td>
<td>0.2162</td>
<td>0.4914</td>
<td>0.1291</td>
<td>0.2603</td>
</tr>
<tr>
<td>0.2866</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To attack round-reduced Speck32/64, the following 8 neural aided statistical distinguishers are built as shown in Table 7.

Table 7: Eight neural aided statistical distinguishers for attacking Speck32/64

<table>
<thead>
<tr>
<th>SD</th>
<th>(\Delta S_1 \rightarrow \Delta S_2)</th>
<th>(q)</th>
<th>(ND)</th>
<th>(c)</th>
<th>(p_1)</th>
<th>(d)</th>
<th>(p_2)</th>
<th>(p_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(SD^t_8)</td>
<td>((0x2800, 0x10) \rightarrow (0x40, 0))</td>
<td>(2^{-2})</td>
<td>(ND^c_1)</td>
<td>0.5</td>
<td>0.5184</td>
<td>1</td>
<td>1.0493</td>
<td>0.4914</td>
</tr>
<tr>
<td>(SD^t_7)</td>
<td>((0x2800, 0x10) \rightarrow (0x40, 0))</td>
<td>(2^{-2})</td>
<td>(ND^c_2)</td>
<td>0.55</td>
<td>0.4057</td>
<td>1</td>
<td>0.3036</td>
<td>0.2863</td>
</tr>
<tr>
<td>(SD^t_6)</td>
<td>((0x2800, 0x10) \rightarrow (0x40, 0))</td>
<td>(2^{-2})</td>
<td>(ND^c_3)</td>
<td>0.55</td>
<td>0.4183</td>
<td>1</td>
<td>0.2884</td>
<td>0.2162</td>
</tr>
<tr>
<td>(SD^t_5)</td>
<td>((0x2800, 0x10) \rightarrow (0x40, 0))</td>
<td>(2^{-2})</td>
<td>(ND^c_4)</td>
<td>0.55</td>
<td>0.5132</td>
<td>1</td>
<td>0.3402</td>
<td>0.2603</td>
</tr>
<tr>
<td>(SD^t_4)</td>
<td>((0x2800, 0x10) \rightarrow (0x40, 0))</td>
<td>(2^{-2})</td>
<td>(ND^c_5)</td>
<td>0.55</td>
<td>0.6785</td>
<td>1</td>
<td>0.3135</td>
<td>0.1161</td>
</tr>
<tr>
<td>(SD^t_3)</td>
<td>((0x2800, 0x10) \rightarrow (0x40, 0))</td>
<td>(2^{-2})</td>
<td>(ND^c_6)</td>
<td>0.55</td>
<td>0.7192</td>
<td>0</td>
<td>0.5498</td>
<td>0.1291</td>
</tr>
<tr>
<td>(SD^t_2)</td>
<td>((0x2800, 0x10) \rightarrow (0x40, 0))</td>
<td>(2^{-2})</td>
<td>(ND^c_7)</td>
<td>0.55</td>
<td>0.8889</td>
<td>0</td>
<td>0.5151</td>
<td>0.0384</td>
</tr>
<tr>
<td>(SD^t_1)</td>
<td>((0x2800, 0x10) \rightarrow (0x40, 0))</td>
<td>(2^{-2})</td>
<td>(ND^c_8)</td>
<td>0.55</td>
<td>0.995</td>
<td>0</td>
<td>0.5065</td>
<td>0.0069</td>
</tr>
</tbody>
</table>

5.1 Key Recovery Attack on 11-round Speck32/64

Attack setting. To recover the last 4 subkeys, the whole attack is divided into seven stages. Table 8 summarizes the attack settings.

Theoretical complexities. Based on Table 8, we have \(N_1 = 2^{14.94}\). To gather \(N_1\) samples that pass the first differential \(\Delta P \rightarrow \Delta S_1\), the neural aided statistical distinguisher \(SD^t_2\) is adopted. According to the setting presented in Section 4, we have \(M = 2^{15.21}\).

Since \(M > N_1\), we only need to find a valid homogeneous set by selecting 16 high PNBs. Let the subscript set of neutral bits be

\[ B = \{11, 14, 15, 20, 21, 22, 0, 1, 3, 23, 24, 26, 4, 27, 5, 28\} \]

Then the probability (formula 29) that a homogeneous set is a valid homogeneous set is

\[ p \prod_{j \in B} \gamma_j = 2^{-4} \times \prod_{j \in B} \gamma_j \approx 2^{-4} \times 0.884 \approx 2^{-4.18} \]
Table 8: Settings of the key recovery attack on 11-round Speck32/64. SKS: surviving key space. EUB: expected upper bound.

<table>
<thead>
<tr>
<th>stage</th>
<th>SD</th>
<th>$\beta_r$</th>
<th>$\beta_w$</th>
<th>$N$</th>
<th>key space</th>
<th>related keys</th>
<th>SKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SD$^2_2$</td>
<td>0.005</td>
<td>$2^{-8}$</td>
<td>$2^{14.94}$</td>
<td>$2^8$</td>
<td>sk$_{11}$[$7 \sim 0$]</td>
<td>$2^4$(EUB)</td>
</tr>
<tr>
<td>2</td>
<td>SD$^2_1$</td>
<td>0.005</td>
<td>$2^{-16}$</td>
<td>$2^{12.94}$</td>
<td>$2^{4+8}$</td>
<td>sk$_{11}$</td>
<td>$2^5$(EUB)</td>
</tr>
<tr>
<td>3</td>
<td>SD$^2_0$</td>
<td>0.001</td>
<td>$2^{-13}$</td>
<td>$2^{12.2}$</td>
<td>$2^{1+8}$</td>
<td>sk$<em>{11}, sk</em>{10}[$7 $\sim 0$]</td>
<td>$2^4$(EUB)</td>
</tr>
<tr>
<td>4</td>
<td>SD$^2_0$</td>
<td>0.001</td>
<td>$2^{-16}$</td>
<td>$2^{2.7}$</td>
<td>$2^{4+8}$</td>
<td>sk$<em>{11}, sk</em>{10}$</td>
<td>$2^5$(EUB)</td>
</tr>
<tr>
<td>5</td>
<td>SD$^5_5$</td>
<td>0.001</td>
<td>$2^{-13}$</td>
<td>$2^{11.73}$</td>
<td>$2^{2+8}$</td>
<td>sk$<em>{11}, sk</em>{10}, sk_9[$7 $\sim 0$]</td>
<td>$2$(EUB)</td>
</tr>
<tr>
<td>6</td>
<td>SD$^5_6$</td>
<td>0.001</td>
<td>$2^{-16}$</td>
<td>$2^{8.78}$</td>
<td>$2^{1+8}$</td>
<td>sk$<em>{11}, sk</em>{10}, sk_9$</td>
<td>$2$(EUB)</td>
</tr>
<tr>
<td>7</td>
<td>SD$^4_3$</td>
<td>0.001</td>
<td>$2^{-17}$</td>
<td>$2^{6.98}$</td>
<td>$2^{1+16}$</td>
<td>sk$<em>{11}, sk</em>{10}, sk_9, sk_8$</td>
<td>$2$(EUB)</td>
</tr>
</tbody>
</table>

Thus the theoretical data complexity is

$$N = 2^{24.18} \times M = 2^{24.18+15.21+1} = 2^{20.39}$$

(35)

As for the computation complexity, it contains two parts. The first is the complexity of finding a valid homogeneous set, which is about $2^{20.39} \times 2^8 \times \frac{1}{10} \approx 2^{24.93}$. The second is the complexity of recovering the right key. It’s worth noticing that stage 1 has been finished when we try to find the valid homogeneous set. Thus the complexity of the key recovery is about $2^{24.45}$. Then the total computation complexity is $2^{24.93} + 2^{24.45} \approx 2^{25.71}$.

Remark 2. In our improved NASA, we also need to perform the verification of neural distinguishers one time after each ciphertext pair is decrypted. Thus we can also measure the complexity as advised in [CY20]. The computation complexity is in terms of full decryption of the attacked cipher. And the time consumption of neural distinguishers is put aside.

Practical experiments. Before performing the key recovery attacks, we first conduct the same two experiments as introduced in Section 4. Now, the homogeneous set is generated using the 16 PNBs as shown in formula (34). After performing the two experiments 100 times respectively, we find

1. To generate 100 valid homogeneous sets, we have generated 1821 homogeneous sets. Thus the probability that a homogeneous set is a valid homogeneous set is about $2^{-4.18}$, which is consistent with the estimated value above.

2. The valid homogeneous set is successfully identified in 100 experiments.

3. The invalid homogeneous set is successfully identified in 94 experiments.

To verify the theoretical analysis of the key recovery attack on 11-round Speck32/64, we have performed practical experiments according to the attack setting above. The target is to recover $(sk_{10}, sk_{11})$, which contains the first 4 stages. In four stages, the number of samples and the decision threshold are $(N, t) = (31341, 9322), (7853, 1995), (4707, 1426), (830, 181)$ respectively. For each experiment, we first find a valid homogeneous set using Algorithm 1. After performing this experiment 100 times, we find

1. In 48 experiments, there are no surviving subkey guess pairs $(kg_{10}, kg_{11})$.

2. In the remaining 52 experiments, the right subkeys $(sk_{10}, sk_{11})$ survives.

3. If we consider all the 100 experiments, the average numbers of surviving subkey guesses are 6.38 (stage 1), 24.31 (stage 2), 13.93 (stage 3), 9.8 (stage 4). If we focus on the 52 experiments, the average numbers of surviving subkey guesses are 11.01 (stage 1), 46.03 (stage 2), 26.46 (stage 3), 18.84 (stage 4).
As we have presented in Section 4, the sampling randomness is a very important point. When we generate homogeneous sets with PNBs, the sampling randomness is likely to be decreased when the number of encryption rounds covered by the first differential transition is very small. In our attack, $\Delta P \rightarrow \Delta S_1$ only covers 1 round. Thus, for a valid homogeneous set, resulting $N$ intermediate state pairs $(S_{i,0}, S_{i,1})$, $i \in \{1, \cdots, N\}$ will be very close to each other in the high-dimensional space.

This will cause a chain reaction. First, corresponding ciphertext pairs may also be close to each other in the input space of neural distinguishers. When we decrypt these ciphertext pairs using several subkey guesses, partially decrypted ciphertext pairs may also possess the similarity. The resulting negative influence is that these subkey guesses will simultaneously survive the attack or be filtered in a high probability.

As we find in the 100 experiments above, there are no surviving subkey guess pairs in some experiments. In the remaining ones, the number of surviving subkey guess pair is a little higher than the expected upper bound. These two phenomenons both occur due to the same cause. Although this negative influence exists, it doesn’t break the correctness of our improved NASA. This is based on following clues

1. The estimation of data complexity for recovering the right subkey is accurate.
2. The number of surviving keys basically meets expectations when all the 100 experiments are considered.

The above analysis can be proved by performing the key recovery attack with simulated valid homogeneous sets. This is completely equivalent to a key recovery attack on 10-round Speck32/64 based on the original NASA. Chen et al. has performed this attack in [CY20].

To ensure the success rate is 1, we can also make a correction to the theoretical complexities of the attack on 11-round Speck32/64 based on our practical experiments. Then the data complexity is about $2^{20.39} \times 2 = 2^{21.39}$, and the computation complexity is about $2^{25.71} \times 2 = 2^{26.71}$.

We have performed more experiments about the negative influence presented above. Generally speaking, the negative influence will be alleviated (even tackled) when the first differential transition or the statistical distinguisher covers more rounds. Thus when we estimate theoretical complexities of attacks based on our improved NASA, it’s acceptable to neglect the negative influence. Chen et al. took about 15 days for performing the above experiment 30 times [CY20]. If we adopt the same hardware, although the time consumption of neural distinguishers is high, we can still perform the attack 100 times in one day.

### 5.2 Key Recovery Attack on 12-round Speck32/64

**Attack setting.** The whole attack is divided into six stages. Table 9 summarizes the attack settings.

<table>
<thead>
<tr>
<th>stage</th>
<th>SD</th>
<th>$\beta_r$</th>
<th>$\beta_w$</th>
<th>$N$</th>
<th>key space</th>
<th>related keys</th>
<th>SKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$SD_8'$</td>
<td>0.005</td>
<td>$2^{-16}$</td>
<td>$2^{18.93}$</td>
<td>$2^{16}$</td>
<td>$sk_{12}$</td>
<td>$2^6$(EUB)</td>
</tr>
<tr>
<td>2</td>
<td>$SD_7'$</td>
<td>0.005</td>
<td>$2^{-14}$</td>
<td>$2^{15.44}$</td>
<td>$2^{4+8}$</td>
<td>$sk_{12}, sk_{11}[7 \sim 0]$</td>
<td>$2^4$(EUB)</td>
</tr>
<tr>
<td>3</td>
<td>$SD_2$</td>
<td>0.005</td>
<td>$2^{-16}$</td>
<td>$2^{12.94}$</td>
<td>$2^{4+8}$</td>
<td>$sk_{12, sk_{11}}$</td>
<td>$2^3$(EUB)</td>
</tr>
<tr>
<td>4</td>
<td>$SD_6'$</td>
<td>0.001</td>
<td>$2^{-13}$</td>
<td>$2^{12.2}$</td>
<td>$2^{5+8}$</td>
<td>$sk_{12, sk_{11, sk_{10}}[7 \sim 0]}$</td>
<td>$2^4$(EUB)</td>
</tr>
<tr>
<td>5</td>
<td>$SD_6'$</td>
<td>0.001</td>
<td>$2^{-16}$</td>
<td>$2^{9.7}$</td>
<td>$2^{4+8}$</td>
<td>$sk_{12, sk_{11, sk_{10}}}$</td>
<td>$2^5$(EUB)</td>
</tr>
<tr>
<td>6</td>
<td>$SD_6'$</td>
<td>0.001</td>
<td>$2^{-21}$</td>
<td>$2^{9.63}$</td>
<td>$2^{5+16}$</td>
<td>$sk_{12, sk_{11, sk_{10}, sk_{9}}}$</td>
<td>$2$(EUB)</td>
</tr>
</tbody>
</table>
Theoretical complexities. Based on Table 9, we have $N_1 = 2^{18.93}$. To gather $N_1$ samples that pass the first differential $\Delta P \rightarrow \Delta S_1$, the neural aided statistical distinguisher $SD_k^1$ is adopted.

The following setting is adopted for finding valid homogeneous sets

$$c = 0.5, \quad d = 1, \quad p_V = p_{2|d_1=d} = 0.5056, \quad p_3 = 0.4914, \quad \beta_V = 0.1, \quad \beta_I = 2^{-16}$$

According to formula (27), we can know $M = 589314 \approx 2^{19.17}$.

Since $M > N_1$, we only need to find a valid homogeneous set by selecting 20 high PNBs. Let the subscript set of neutral bits be

$$B = \{11, 14, 15, 19, 21, 22, 0, 1, 3, 23, 24, 26, 4, 27, 5, 28, 6, 29, 7, 30\}$$

Then the probability (formula (29)) that a homogeneous set is a valid homogeneous set is

$$p \prod_{j \in B} \gamma_j = 2^{-4} \times \prod_{j \in B} \gamma_j \approx 2^{-4} \times 0.594 \approx 2^{-4.75}$$

Thus the theoretical data complexity is

$$N = 2^{4.75} \times M \times 2 = 2^{4.75+19.17+1} = 2^{24.92}$$

The complexity of finding a valid homogeneous set is about $2^{24.92} \times 2^{16} \times \frac{1}{\rho} \approx 2^{37.34}$. The complexity of the key recovery is about $2^{32.39}$. Thus the total computation complexity is $2^{37.34} + 2^{32.39} \approx 2^{37.39}$.

5.3 Key Recovery Attack on 13-round Speck32/64

Attack setting. The attack is divided into three stages. Table 10 summarizes the attack settings.

<table>
<thead>
<tr>
<th>stage</th>
<th>$SD$</th>
<th>$\beta_r$</th>
<th>$\beta_w$</th>
<th>$N$</th>
<th>key space</th>
<th>related keys</th>
<th>SKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$SD_k^1$</td>
<td>0.005</td>
<td>$2^{-32}$</td>
<td>$2^{19.7}$</td>
<td>$2^{32}$</td>
<td>$sk_{12}, sk_{11}$</td>
<td>$2^6$(EUB)</td>
</tr>
<tr>
<td>2</td>
<td>$SD_k^2$</td>
<td>0.005</td>
<td>$2^{-22}$</td>
<td>$2^{13.28}$</td>
<td>$2^{6+16}$</td>
<td>$sk_{12}, sk_{11}, sk_{10}$</td>
<td>$2^6$(EUB)</td>
</tr>
<tr>
<td>3</td>
<td>$SD_k^3$</td>
<td>0.001</td>
<td>$2^{-21}$</td>
<td>$2^{9.97}$</td>
<td>$2^{5+16}$</td>
<td>$sk_{12}, sk_{11}, sk_{10}, sk_9$</td>
<td>$2^6$(EUB)</td>
</tr>
</tbody>
</table>

Theoretical complexities. Based on Table 10, we have $N_1 = 2^{19.7}$. To gather $N_1$ samples that pass the first differential $\Delta P \rightarrow \Delta S_1$, the neural aided statistical distinguisher $SD_k^1$ is adopted.

The following setting is adopted for finding valid homogeneous sets

$$c = 0.5, \quad d = 1, \quad p_V = p_{2|d_1=d} = 0.5056, \quad p_3 = 0.4914, \quad \beta_V = 0.1, \quad \beta_I = 2^{-32}$$

According to formula (27), we can know $M = 1119074 \approx 2^{20.09}$.

Since $M > N_1$, we only need to find a valid homogeneous set by selecting 21 high PNBs. Let the subscript set of neutral bits be

$$B = \{11, 14, 15, 19, 21, 22, 0, 1, 3, 23, 24, 26, 4, 27, 5, 28, 6, 29, 7, 30, 8\}$$

Then the probability (formula (29)) that a homogeneous set is a valid homogeneous set is

$$p \prod_{j \in B} \gamma_j = 2^{-4} \times \prod_{j \in B} \gamma_j \approx 2^{-4} \times 0.446 \approx 2^{-5.17}$$

(38)
Thus the theoretical data complexity is
\[ N = 2^{5.17} \times M \times 2 = 2^{5.17+19.7+1} = 2^{25.87} \quad (39) \]
The complexity of finding a valid homogeneous set is about \(2^{25.87} \times 2^{32} \times 2^{\frac{2}{13}} \approx 2^{55.17}\). The complexity of the key recovery is about \(2^{42.05}\). Thus the total computation complexity is \(2^{55.17} + 2^{42.05} \approx 2^{55.17}\).

6 Key Recovery Attacks on Round Reduced Speck48/X

To prove the superiority of our improved NASA, we have performed key recovery attacks on 12-round Speck48/72 and Speck49/96. In these two attacks, we adopt the same 1-round differential transition and 4-round differential transition
\[
\Delta P = (0x400052, 0x504200) \xrightarrow{d} \Delta S_1 = (0x820200, 0x1202) \\
\Delta S_1 = (0x820200, 0x1202) \xrightarrow{d} \Delta S_2 = (0x808000, 0x808004) \quad (40)
\]
where \(p = 2^{-5}\) and \(q = 2^{-7}\).

Let \(X = 2^{24}\) and run Algorithm 3 for each plaintext bit, we have estimated the neutrality of each plaintext bit for Speck48/X. Table 11 shows the estimation of each plaintext bit's neutrality.

Table 11: Neutrality \(\gamma\) of each plaintext bit for \((0x400052, 0x504200) \rightarrow (0x820200, 0x1202)\)

<table>
<thead>
<tr>
<th>Index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\gamma)</td>
<td>0.999</td>
<td>0.993</td>
<td>0.993</td>
<td>0.991</td>
<td>0.969</td>
<td>0.932</td>
<td>0.872</td>
<td>0.76</td>
<td>0.51</td>
<td>1</td>
</tr>
<tr>
<td>Index</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>0.981</td>
<td>0.969</td>
<td>0.938</td>
<td>0.865</td>
<td>0</td>
<td>0.75</td>
<td>0.504</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Index</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.483</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Index</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>0</td>
<td>1</td>
<td>0.998</td>
<td>0.995</td>
<td>0.994</td>
<td>0.985</td>
<td>0.971</td>
<td>0.936</td>
<td>0.865</td>
<td>0.758</td>
</tr>
<tr>
<td>Index</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\gamma)</td>
<td>0.503</td>
<td>0</td>
<td>0.985</td>
<td>0.972</td>
<td>0.947</td>
<td>0.881</td>
<td>0</td>
<td>0.74</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Based on \(\Delta S_2 = (0x808000, 0x808004)\), we have trained three teacher distinguishers \(ND_3, ND_4, ND_5\) using the same training pipeline presented in [Goh19]. Let \(T = \{47 \sim 32, 23 \sim 8\}\), we have trained two student distinguishers \(ND_3, ND_4\). Let \(c_3 = 0.55\), Table 12 shows the estimation of \(p_{\Delta d_i}\) of the five neural distinguishers.

Table 12: The estimation of \(p_{\Delta d_i}\) of five neural distinguishers against Speck48/X

<table>
<thead>
<tr>
<th>(ND_3)</th>
<th>(d_1)</th>
<th>(\Delta d_i)</th>
<th>(p_{\Delta d_i})</th>
<th>(\Delta d_i)</th>
<th>(p_{\Delta d_i})</th>
<th>(\Delta d_i)</th>
<th>(p_{\Delta d_i})</th>
<th>(\Delta d_i)</th>
<th>(p_{\Delta d_i})</th>
<th>(\Delta d_i)</th>
<th>(p_{\Delta d_i})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d_1)</td>
<td>0.9871</td>
<td>0.5885</td>
<td>0.3508</td>
<td>0.2158</td>
<td>0.1397</td>
<td>0.095</td>
<td>≤ 0.0678</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\Delta d_i)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>(p_{\Delta d_i})</td>
<td>0.7494</td>
<td>0.5531</td>
<td>0.4189</td>
<td>0.329</td>
<td>0.2688</td>
<td>0.2299</td>
<td>≤ 0.2035</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\Delta d_i)</td>
<td>0.6555</td>
<td>0.5097</td>
<td>0.4152</td>
<td>0.3588</td>
<td>0.323</td>
<td>0.3</td>
<td>≤ 0.2849</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(p_{\Delta d_i})</td>
<td>0.3304</td>
<td>0.2906</td>
<td>0.2629</td>
<td>0.2448</td>
<td>0.2312</td>
<td>0.2226</td>
<td>≤ 0.2162</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\Delta d_i)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>(p_{\Delta d_i})</td>
<td>0.2942</td>
<td>0.2696</td>
<td>0.2545</td>
<td>0.2454</td>
<td>0.2407</td>
<td>0.2369</td>
<td>≤ 0.2354</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

According to the definition of \(p_3\), we have also estimated \(p_3\) for each neural distinguisher. Table 13 summarizes the estimation of \(p_3\) of these five neural distinguishers.

Based on these five neural distinguishers above, the following 5 neural aided statistical distinguishers are built as shown in Table 14.
Table 13: The estimation of \( p_3 \) of five neural distinguishers, \( c = 0.55 \).

<table>
<thead>
<tr>
<th>ND</th>
<th>( ND_{13}^1 )</th>
<th>( ND_{14}^1 )</th>
<th>( ND_{26}^1 )</th>
<th>( ND_{31}^1 )</th>
<th>( ND_{32}^1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_3 )</td>
<td>0.0102</td>
<td>0.2352</td>
<td>0.1349</td>
<td>0.2304</td>
<td>0.1999</td>
</tr>
</tbody>
</table>

Table 14: Five neural aided statistical distinguishers for attacking Speck48/X. \( \Delta S_1 \rightarrow \Delta S_2 = 0x820200/0x1202 \rightarrow 0x808000/0x808004 \).

<table>
<thead>
<tr>
<th>SD</th>
<th>( q )</th>
<th>( ND )</th>
<th>( c )</th>
<th>( p_l )</th>
<th>( p_r )</th>
<th>( p_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( SD_{26}^1 )</td>
<td>( 2^{-12} )</td>
<td>( ND_{13}^1 )</td>
<td>0.55</td>
<td>0.2942</td>
<td>1</td>
<td>0.2545</td>
</tr>
<tr>
<td>( SD_{31}^1 )</td>
<td>( 2^{-12} )</td>
<td>( ND_{26}^1 )</td>
<td>0.55</td>
<td>0.3304</td>
<td>1</td>
<td>0.2629</td>
</tr>
<tr>
<td>( SD_{32}^1 )</td>
<td>( 2^{-12} )</td>
<td>( ND_{31}^1 )</td>
<td>0.55</td>
<td>0.6555</td>
<td>1</td>
<td>0.4152</td>
</tr>
<tr>
<td>( SD_{3}^1 )</td>
<td>( 2^{-12} )</td>
<td>( ND_{32}^1 )</td>
<td>0.55</td>
<td>0.7494</td>
<td>0</td>
<td>0.5531</td>
</tr>
<tr>
<td>( SD_{3}^1 )</td>
<td>( 2^{-12} )</td>
<td>( ND_{3}^1 )</td>
<td>0.55</td>
<td>0.9871</td>
<td>0</td>
<td>0.5885</td>
</tr>
</tbody>
</table>

6.1 Key Recovery Attack on 12-round Speck48/72

Attack setting. To attack 12-round Speck48/72, we need to recover the last 3 subkeys. Table 15 summarizes the attack settings.

Table 15: Settings of the key recovery attack on 12-round Speck48/72 with \( p_0 = 2^{-12} \).

<table>
<thead>
<tr>
<th>stage</th>
<th>( SD )</th>
<th>( \beta_r )</th>
<th>( \beta_w )</th>
<th>( N )</th>
<th>key space</th>
<th>related keys</th>
<th>SKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( SD_{26}^1 )</td>
<td>0.005</td>
<td>( 2^{-16} )</td>
<td>2^{26.32}</td>
<td>( 2^{16} )</td>
<td>( sk_{12}[15 \sim 0] )</td>
<td>( 2^4(EUB) )</td>
</tr>
<tr>
<td>2</td>
<td>( SD_{31}^1 )</td>
<td>0.001</td>
<td>( 2^{-24} )</td>
<td>2^{25.27}</td>
<td>( 2^{4+8} )</td>
<td>( sk_{12} )</td>
<td>( 2^6(EUB) )</td>
</tr>
<tr>
<td>3</td>
<td>( SD_{32}^1 )</td>
<td>0.001</td>
<td>( 2^{-21} )</td>
<td>2^{21.64}</td>
<td>( 2^{5+16} )</td>
<td>( sk_{12}, sk_{11}[15 \sim 0] )</td>
<td>( 2^6(EUB) )</td>
</tr>
<tr>
<td>4</td>
<td>( SD_{3}^1 )</td>
<td>0.001</td>
<td>( 2^{-24} )</td>
<td>2^{21.74}</td>
<td>( 2^{5+8} )</td>
<td>( sk_{12}, sk_{11} )</td>
<td>( 2(EUB) )</td>
</tr>
<tr>
<td>5</td>
<td>( SD_{3}^1 )</td>
<td>0.001</td>
<td>( 2^{-25} )</td>
<td>2^{16.36}</td>
<td>( 2^{4+24} )</td>
<td>( sk_{12}, sk_{11}, sk_{10} )</td>
<td>( 2(EUB) )</td>
</tr>
</tbody>
</table>

Theoretical complexities. Based on Table 15, we have \( N_1 = 2^{26.32} \). To gather \( N_1 \) samples that pass the first differential \( \Delta P \rightarrow \Delta S_1 \), the neural aided statistical distinguisher \( SD_{26}^1 \) is adopted.

The following setting is adopted for finding valid homogeneous sets

\[
c = 0.55, \quad d = 1, \quad p_V = p_{2d|d=0} = 0.2696, \quad p_3 = 0.2304, \quad \beta_V = 0.1, \quad \beta_I = 2^{-16}
\]

According to formula (27), we can know \( M = 56194311 \approx 2^{25.74} \).

Since \( M > 2^{20} \), we need to select 26 high \(\text{PNB}^s\) for finding a valid homogeneous set. Since \( M < N_1 < 2 \times M \), we need to gather 2 valid homogeneous sets. However, by exploiting a plaintext bit which has \( \gamma = 1 \), we can generate \( N_1 \) samples with 1 valid homogeneous set. Let the subscript set of neutral bits be

\[
\mathcal{B} = \{0 \sim 4, 10 \sim 12, 18, 19, 21, 23, 25 \sim 27, 29, 31 \sim 37, 42 \sim 44\}
\]

Then the probability (formula (29)) that a homogeneous set is a valid homogeneous set is

\[
\mathcal{P} \prod_{j \in \mathcal{B}} \gamma_j = 2^{-5} \times \prod_{j \in \mathcal{B}} \gamma_j \approx 2^{-5} \times 0.65 \approx 2^{-5.62} \quad (41)
\]

It’s worth noticing that \( \gamma_9 = 1 \). After a valid homogeneous set is found, we can generate the left \( N_1 - M \) samples by flipping the ninth plaintext bit of \( M \) plaintext pairs. Thus the total data complexity is

\[
N = 2^{5.62} \times M \times 2 + (N_1 - M) \times 2 \approx 2^{42.37} \quad (42)
\]

The complexity of finding a valid homogeneous set is about \( 2^{5.62} \times M \times 2 \approx 2^{44.78} \). The complexity of the key recovery is about \( 2^{40.58} \). Thus the total computational complexity is \( 2^{44.78} + 2^{40.58} \approx 2^{44.86} \).
6.2 Key Recovery Attack on 12-round Speck48/96

**Attack setting.** To attack 12-round Speck48/96, we need to recover the last 4 subkeys. The attack setting for 12-round Speck48/72 can be adopted for recovering the last 3 subkeys of Speck48/96. We can recover \( k_9 \) by adopting a teacher distinguisher \( ND_2 \). And the extra computation complexity is negligible.

**Theoretical complexity.** Based on the attack setting, we know the data complexity is \( N \approx 2^{32.37} \). The computation complexity is about \( 2^{44.86} \).

7 Analysis of the Neural Distinguisher’s Training

During the training of neural distinguishers, the optimization goal is distinguishing accuracy. We wonder whether this goal is useful for performing a better NASA.

To answer this question, we can set the posterior probability threshold to \( c = 0.5 \). Denote the distinguishing accuracy of the neural distinguisher as \( acc \), then the relation between \( acc \) and \( p_1, p_3 \) is

\[
acc = 0.5 \times p_1 + 0.5 \times (1 - p_3)
\]

The target of a neural distinguisher’s training is increasing \( acc \). This is equivalent to increasing \( p_1 \) or decreasing \( p_3 \). The impacts of \( p_1, p_3 \) on the data complexity are \( \mathcal{O}((p_1 - p_2)^{-2}) \) and \( \mathcal{O}(p_3) \) respectively, thus the goal of optimizing the distinguishing accuracy is useful for NASA. When we change the value of \( c \), the difference only has a slight influence on the data complexity. Conclusion above still holds.

8 Conclusions

In this paper, we have proposed an improved neural aided statistical attack for cryptanalysis. By dividing the prepended differential transition into two parts, the impact of the probability \( p \) of the first part on the data complexity can be reduced to \( \mathcal{O}(p^{-1}) \) with the help of the newly proposed homogeneous set. Applications to round reduced Speck32/64, Speck48/72, Speck48/96 have proved the superiority of the improved neural aided statistical attack. Actually, our work in this paper is equivalent to filtering wrong pairs in differential cryptanalysis.

Next, our research will focus on the following issues. The first is exploring the properties of the neural distinguisher. If we know how to build neural distinguishers against more rounds, the statistical attack model can be greatly improved. The second is understanding the knowledge learned by the neural distinguisher. It can help us get rid of the dependence on neural distinguishers and improve the theoretical framework. The third is developing a more efficient and generic key search strategy. Applying techniques of the artificial intelligence community to cryptanalysis is an exciting research topic. We believe more aspects of cryptography will benefit from the development of this promising direction.

Acknowledgement

This work is supported by the National Key Research and Development Program of China (2018YFB0803405, 2017YFA0303903).

References

[AFK+08] Jean-Philippe Aumasson, Simon Fischer, Shahram Khazaei, Willi Meier, and Christian Rechberger. New features of latin dances: analysis of salsa, chacha,


### A Description of Speck and Key Schedule Inversion

A short description of Speck is provided in this appendix. More details can be found in [BSS+15].
A.1 The Key Schedule and Round Function

Speck is a family of block ciphers containing 10 variants. The variants are identified with a $2n/mn$ label, and defined with rotation constants $\alpha$ and $\beta$ and a number of rounds $T$. The block size is $2n$ bits and the internal word size is $n$ bits. The key size is $mn$ bits.

The key schedule of Speck family expands the initial $m$-word master key $(\ell_{m-2}, \ldots, \ell_0, k_0)$ into $h$ round key words $k_0, k_1, \ldots, k_{h-1}$ according to the algorithm 4. Figure 3 shows the round function of Speck32/64.

Algorithm 4 Key Schedule of Speck

\begin{verbatim}
1: for $i = 0$ to $h - 2$ do
2: $\ell_{i+m-1} \leftarrow (sk_i \oplus (\ell_i \gg \alpha)) \oplus i$;
3: $sk_{i+1} \leftarrow (sk_i \ll \beta) \oplus \ell_{i+m-1}$;
4: end for
\end{verbatim}

Figure 3: The round function of Speck. For Speck32/64, $\alpha = 7, \beta = 2$. For Speck48/X, $\alpha = 8, \beta = 3$.

A.2 Key Schedule Inversion

The master key of $h$-round Speck$mn/2n$ can be directly recovered once the last $m$ subkeys are recovered. This is based on the key schedule inversion algorithm [Din14].

Given a sequence of $m$ round key words $sk_{j-m}, sk_{j-m+1}, sk_{j-m+2}, sk_{j-m+3}$ for any $j \in \{m, \ldots, h\}$, we can efficiently invert the key schedule and calculate the master key.

First, we can determine $sk_{j-m-1}$ using the following key schedule equalities

\begin{align}
\ell_{j+m-3} & = sk_{j-1} \oplus (sk_{j-2} \ll \beta) \\
\ell_{j-2} & = ((\ell_{j+m-3} \oplus (j-2)) \Box sk_{j-2}) \ll \alpha \\
sk_{j-m-1} & = (sk_{j-m} \oplus \ell_{j-2}) \gg \beta
\end{align}

Next, given $sk_{j-m-1}, \ldots, sk_{j-2}$, we iteratively continue the inversion of the key schedule and derive the master key.