Weak Tweak-Keys for the CRAFT Block Cipher

Gregor Leander[1] and Shahram Rasoolzadeh[2]

[1] Ruhr University Bochum, Bochum, Germany, firstname.lastname@rub.de
[2] Radboud University, Nijmegen, The Netherlands, firstname.lastname@ru.nl

Abstract. CRAFT is a lightweight tweakable Substitution-Permutation-Network (SPN) block cipher optimized for efficient protection of its implementations against Differential Fault Analysis (DFA) attacks. In this paper, we present an equivalent description of CRAFT up to a simple mapping on the plaintext, ciphertext and round tweakeys. We show that the new representation, for a sub-class of keys, leads to a new structure which is a Feistel network, with non-linear operation and key addition only on half the state. Consequently, it reveals a class of weak keys for which CRAFT is less resistant against differential and linear cryptanalyses. As a result, we present one weak-key single-tweak differential attack on 23 rounds (with time complexity of 2^{94} encryptions and data complexity of 2^{74} chosen plaintext/tweak/ciphertext tuples and works for 2^{112} weak keys) and one weak-key related-tweak attack on 26 rounds of the cipher (with time complexity of 2^{105} encryptions and data complexity 2^{73} chosen plaintext/tweak/ciphertext tuples and works for 2^{108} weak keys). Note that these attacks do not break the security claim of the CRAFT block cipher.

Keywords: CRAFT · partial key addition · partial non-linear layer

1 Introduction

CRAFT is a tweakable block cipher presented at FSE 2019 and designed by Beierle, Leander, Moradi, and Rasoolzadeh [BLMR19]. The cipher follows the SPN design with 32 rounds and iterates 4 round tweakeys as of the tweakey schedule. The main goal of CRAFT was to efficiently provide protection of its implementations against DFA attacks [BS97] while to provide decryption on top of the encryption with minimum overhead was considered as a side goal in the design criteria. The encryption-only implementation of the cipher needs 949 GE (using the IBM 130nm ASIC library), which is less than that of the any reported round-based implementation of a lightweight block cipher whose key size is 128 bits. Besides, considering the protected against DFA implementation of the cipher, under the same settings with respect to the employed error-detection code, its area overhead (even with decryption and tweak support) is smaller than all block ciphers considered in [BLMR19] with compatible state and key size.

The designers of CRAFT provided a detailed security analysis of the cipher in their proposal paper which covers differential, linear, impossible differential, zero-correlation linear hull, meet-in-the-middle, time-data-memory trade-offs, integral (and division property), and invariant attacks. Overall, they claimed 124 bit security in the related-tweak attacker model. After the publication of the design, some other follow-up cryptanalysis has been published [HSN+19, MA19, EY19, GSS+20] and in the following, we briefly explain results of these analyses.

1.1 Known Results on CRAFT

Hadipour et al. [HSN+19] presented a detailed security analysis of CRAFT. In particular, they presented 14-round related-tweak zero-correlation linear hull distinguishers. Using the same distinguishers and following the connection between zero-correlation and integral distinguisher, they also presented a 14-round related-tweak integral distinguisher. Furthermore, using the automated search model based on an MILP tool, they found the mistake reported on the differential probability and on the maximum number of rounds for single-tweak differential distinguishers.

Moghaddam and Ahmadian [MA19] used an MILP-based tool to find truncated differentials distinguishers for CRAFT, MIDORI, and SKINNY block ciphers. In the case of CRAFT, they reported a 12-round distinguisher. ElSheikh and Youssef [EY19] presented a related-key differential attack that recovers the whole 128-bit key in a full-round CRAFT with querying corresponding ciphertext for about 2^{36} chosen plaintexts and time complexity of about 2^{36} encryptions using a negligible memory. Note that the designers did not claim any security in the related-key model.

More recently, and most relevant for our work, Guo et al. [GSS+20] studied the combination of the involutory S-box and the simple tweakey schedule used in the CRAFT block cipher. They found that some input difference at a particular position can be preserved through any number of rounds if the input pair follows certain truncated differential trails. They used this property to construct weak-key truncated differential distinguishers of round-reduced CRAFT. As a result, they found some 16-round and one 18-round truncated differential distinguishers of CRAFT that can be extended to a 20-round distinguisher with probability 2^{-63}. Moreover, they presented a key recovery attack on the 19-round CRAFT with 2^{61} data, 2^{68} memory, $2^{94.6}$ time complexity and success probability of about 80%.

1.2 Our Contribution

In this paper, we first study the round operations used in CRAFT in detail. Using properties of these operations, we redefine the round function of the cipher which leads to an equivalent description of CRAFT up to a simple mapping on the plaintext, ciphertext and round tweakeys. In a weak tweak-key scenario, mainly thanks to the involutory S-box and the special choice of MixColumns used in CRAFT, the equivalent representation of the cipher leads to a Feistel network where the non-linear operation (S-box layer) only is applied on half of the state. In this weak tweak-key class of the encryption, the 128-bit key must be one of 2^{88} weak keys, and for each weak-key there are exactly 2^{8} 64-bit tweaks those are included in the set of 2^{32} weak tweaks.

We analyze the security of the new weak tweak-key structure of the cipher and show that compared to the original structure of CRAFT, the new structure is less resistant against differential and linear cryptanalyses. This part of our results in particular gives another explanation of the results in [GSS+20] and explains the weak tweak-keys identified there.

As a consequence, we find several 18-round single-tweak differential and several 21-round related-tweak differentials with higher EDP than 2^{-64}. We apply one of 18-round single-tweak differentials to do a differential key recovery attack on 23-round CRAFT under the weak key model which can recover the key (out of 2^{112} weak keys) using 2^{74} chosen plaintext/tweak/ciphertext tuples within about 2^{94} encryptions by using 2^{51} blocks of memory. We also apply one of 21-round related-tweak differentials to do a 26-round differential key recovery attack which recovers the key (out of 2^{108} weak keys) using 2^{73} chosen plaintext/tweak/ciphertext tuples within about 2^{105} encryptions by using 2^{60} blocks of memory. Note that it is possible to reduce the complexity of attack, by reducing number of the appending rounds. We emphasize that these attacks do not overcome the security claim of the CRAFT block cipher and they are only a security evaluation of the cipher.

1.3 Outline

First in Section 2, we recall the design of CRAFT. Then in Section 3, we present an equivalent definition for CRAFT round function and using the new representation, we introduce a weak-key structure for the cipher. In Section 4, we use an MILP tool to find all the activity patterns with the minimum number of active S-boxes in differential and linear trails of the weak tweak-key CRAFT structure. We estimate the expected differential probability (EDP) for the differential effect within these differential activity patterns and we show that the actual weak-key space in the differential activity patterns can be larger than the weak tweak-key space for the weak tweak-key structure. Later, in Section 5, we describe the details of our single-tweak attack 23-round and related-tweak attack on 26-round CRAFT. Finally, we conclude our paper in Section 6.

2 CRAFT Specification

CRAFT is a lightweight tweakable block cipher consisting of a 64-bit block, a 128-bit key, and a 64-bit tweak. The state is viewed as a 4×4 array of nibbles. The notation $X[i, j]$ denotes the nibble located at row i and column j of the state. By concatenating the rows of the state, one can denote the state as a vector of nibbles that $X[i]$ denotes the nibble in i-th position of this vector, i.e., $X[i, j] = X[4i + j]$.

The 128-bit key is split into two 64-bit keys K_0 and K_1. Together with the 64-bit tweak input T, four 64-bit round-tweakeys TK_0, TK_1, TK_2 and TK_3 are derived. The cipher uses 32 rounds that the first 31 one are identical round functions $\mathcal{R}_i$ with $0 \le i \le 30$ and the last one is a linear round $\mathcal{R}_{31}$. CRAFT makes use of the following five operations:

- SB: The 4-bit involutory S-box S is applied to each nibble of the state. This S-box is the same as the S-box used in the block cipher MIDORI [BBI+15].

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	c	a	d	3	e	b	f	7	8	9	1	5	0	2	4	6

- MC: The following involutory binary matrix is multiplied to each column of the state:

$$M = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

- PN: Using an involutory permutation P, the position of the nibbles in the state changes. Particularly, $X[i]$ is replaced with $X[P(i)]$, where

$$P = [15, 12, 13, 14,\ 10, 9, 8, 11,\ 6, 5, 4, 7,\ 1, 2, 3, 0] .$$

- $\mathtt{A}_{RC_i}$: One 4-bit and one 3-bit round-constant value is XORed with the forth and the fifth state nibble, respectively.

- $\mathtt{A}_{TK_i}$: The cipher derives four 64-bit tweakeys TK_0, TK_1, TK_2 and TK_3 from the tweak T and the key $(K_0 \,\|\, K_1)$ as

$$TK_0 = K_0 \oplus T\ ,\ TK_1 = K_1 \oplus T\ ,\ TK_2 = K_0 \oplus \mathtt{QN}(T)\ ,\ TK_3 = K_1 \oplus \mathtt{QN}(T)\ ,$$

where $\mathtt{QN}(T)$ applies the permutation

$$Q = [12, 10, 15, 5,\ 14, 8, 9, 2,\ 11, 3, 7, 4,\ 6, 0, 1, 13]$$

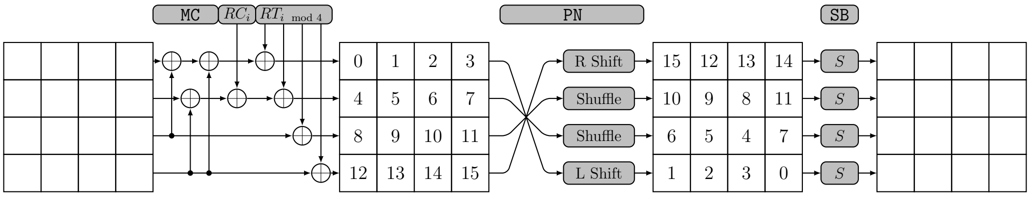


Figure 1: One full round of CRAFT.

on the position of tweak nibbles which $T[i]$ is replaced by $T[Q(i)]$. Then in each round i, without any key update, the tweakey $TK_{i \bmod 4}$ is XORed to the state. For simplicity, we will use TK_i.

The round functions $\mathcal{R}_i$, with $0 \le i < 31$, are defined as

$$\mathcal{R}_i = \texttt{SB} \circ \texttt{PN} \circ \texttt{A}_{TK_i} \circ \texttt{A}_{RC_i} \circ \texttt{MC}$$

and the last round $\mathcal{R}_{31}$ is defined as

$$\mathcal{R}_{31} = \texttt{A}_{TK_3} \circ \texttt{A}_{RC_{31}} \circ \texttt{MC}\,.$$

The full one-round function of CRAFT is depicted in Figure 1.

3 CRAFT Weak Tweak-Key Structure

In this section, based on the given properties in the following for the linear round operations of the cipher, we present an equivalent definition for the CRAFT round function. Based on the new representation, we introduce a weak tweak-key class for the cipher. In this weak tweak-key class of the encryption, the 128-bit key must be one of 2^{88} weak keys, and for each weak-key there are exactly 2^8 64-bit tweaks those are included in the set of 2^{32} weak tweaks. In the second part of the section, we show how to minimize the size of the weak-key set by slightly modifying QN operation with respect to the criteria applied in the design of CRAFT cipher. However, it is necessary to analyze the security of the modified cipher concerning the other cryptanalysis.

We use X' and X'' to denote left and right halves of the state X, i.e., $X' = (X[0], \ldots, X[7])$ and $X'' = (X[8], \ldots, X[15])$. We use the same notation to denote each halves of the key, tweak and tweakey, e.g., we use TK'_i and TK''_i for the latter case.

Property 1. In MC operation, for each column index $j \in \{0, \ldots, 3\}$, we have

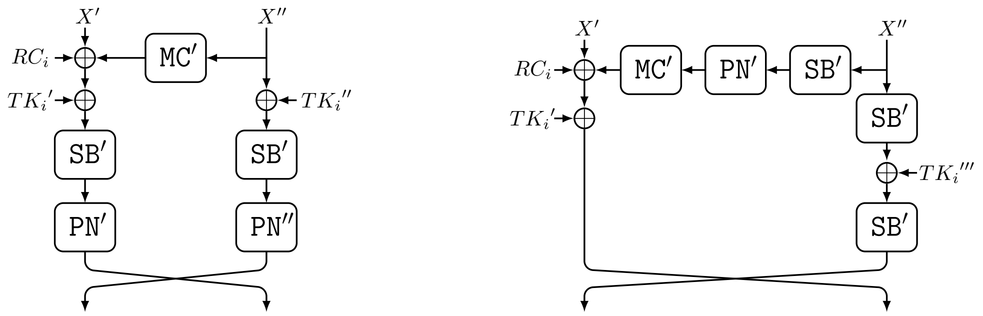
$$M\left(\begin{bmatrix} X[0,j] \\ X[1,j] \\ X[2,j] \\ X[3,j] \end{bmatrix}\right) = \begin{bmatrix} X[0,j] \\ X[1,j] \\ X[2,j] \\ X[3,j] \end{bmatrix} \oplus \begin{bmatrix} X[2,j] \oplus X[3,j] \\ X[3,j] \\ 0 \\ 0 \end{bmatrix}.$$

That is, a linear combination of the right half is XORed with the left half, i.e.,

$$\texttt{MC}(X' \,\|\, X'') = \left(X' \oplus \texttt{MC}'(X'') \,\|\, X''\right),$$

where $\texttt{MC}'$ is the corresponding linear operation with multiplying the binary matrix M' to each column of the right half:

$$M' = \begin{bmatrix} \texttt{1} & \texttt{1} \\ \texttt{0} & \texttt{1} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} X[2,j] \\ X[3,j] \end{bmatrix} \mapsto \begin{bmatrix} X[2,j] \oplus X[3,j] \\ X[3,j] \end{bmatrix}.$$



(a)

(b)

Figure 2: The round function for (a) CRAFT representation, and (b) equivalent CRAFT.

Property 2. PN operation replaces the left half of the state with a nibble permutation of the right half and vice versa, i.e.,

$$\mathtt{PN}(X' \,\|\, X'') = \big(\mathtt{PN}''(X'') \,\|\, \mathtt{PN}'(X')\big)\,,$$

with $\mathtt{PN}'$ using the following P' permutation to replace $X[i]$ by $X[P'(i)]$:

$$P' = [6, 5, 4, 7, 1, 2, 3, 0]\ .$$

Moreover, since PN is an involutive operation, we have $\mathtt{PN}'' = \mathtt{PN}'^{-1}$.

3.1 CRAFT Equivalent and Weak Tweak-Key Representation

Using Property 1 and Property 2, it is possible to represent the round function of CRAFT as the function shown in Figure 2(a) where we use $\mathtt{SB}'$ operation to denote the application of the S-box S to each of eight nibbles. Besides, we use $\mathtt{A}'$ to denote the tweakey or round constant addition in each half of the state.

Proposition 1. *CRAFT encryption is equivalent (up-to a nibble-permutation and an S-box operation on the right half of the plaintext/ciphertext and a nibble-permutation on the right half of the round tweakeys) to the encryption with the round function*

$$\mathcal{R}'_i(X' \,\|\, X'') = \big(\mathtt{SB}' \circ \mathtt{A}'_{TK_i'''} \circ \mathtt{SB}'(X'') \,\|\, \mathtt{A}'_{TK_i'} \circ \mathtt{A}'_{RC_i} \circ \mathtt{MC}' \circ \mathtt{PN}' \circ \mathtt{SB}'(X'') \oplus X'\big)\,,$$

that $TK_i''' = \mathtt{PN}''(TK_i'')$, except in the last round that similar to the Feistel network, the final swapping between the right and left halves is omitted. The equivalent round function is shown in Figure 2(b).

Proof. For CRAFT round function, we have:

$$\begin{aligned}
\mathcal{R}_i(X' \,\|\, X'') &= \mathtt{SB} \circ \mathtt{PN} \circ \mathtt{A}_{TK_i} \circ \mathtt{A}_{RC_i} \circ \mathtt{MC}(X' \,\|\, X'') \\
&= \mathtt{PN} \circ \mathtt{SB} \circ \mathtt{A}_{TK_i} \circ \mathtt{A}_{RC_i} \circ \mathtt{MC}(X' \,\|\, X'') \\
&= \mathtt{PN} \circ \mathtt{SB} \circ \mathtt{A}_{TK_i} \circ \mathtt{A}_{RC_i}(X' \oplus \mathtt{MC}'(X'') \,\|\, X'') \\
&= \mathtt{PN} \circ \mathtt{SB} \circ \mathtt{A}_{TK_i}\Big(\mathtt{A}'_{RC_i}\big(X' \oplus \mathtt{MC}'(X'')\big) \,\|\, X''\Big) \\
&= \mathtt{PN} \circ \mathtt{SB}\Big(\mathtt{A}'_{TK_i'} \circ \mathtt{A}'_{RC_i}\big(X' \oplus \mathtt{MC}'(X'')\big) \,\|\, \mathtt{A}'_{TK_i''}(X'')\Big) \\
&= \mathtt{PN}\Big(\mathtt{SB}' \circ \mathtt{A}'_{TK_i'} \circ \mathtt{A}'_{RC_i}\big(X' \oplus \mathtt{MC}'(X'')\big) \,\|\, \mathtt{SB}' \circ \mathtt{A}'_{TK_i''}(X'')\Big) \\
&= \Big(\mathtt{PN}'' \circ \mathtt{SB}' \circ \mathtt{A}'_{TK_i''}(X'') \,\|\, \mathtt{PN}' \circ \mathtt{SB}' \circ \mathtt{A}'_{TK_i'} \circ \mathtt{A}'_{RC_i}\big(X' \oplus \mathtt{MC}'(X'')\big)\Big)\,.
\end{aligned}$$

This is the same representation of $\texttt{CRAFT}$ round function in Figure 2(a). Similarly for the last linear round, we have:

$$\mathcal{R}_{31}(X' \,\|\, X'') = \Big(\texttt{A}'_{TK'_3}\big(X' \oplus \texttt{MC}'(X'') \oplus RC'_{31}\big) \,\|\, \texttt{A}'_{TK''_3}(X'')\Big)\,.$$

Consider now a bijective function $\mathcal{G}$. By iterating $\mathcal{R}'_i = \mathcal{G} \circ \mathcal{R}_i \circ \mathcal{G}^{-1}$ instead of $\mathcal{R}_i$ round functions, we reach to an encryption equivalent to the $\texttt{CRAFT}$ encryption.

$$\begin{aligned}\mathcal{R}'_{31} \circ \ldots \circ \mathcal{R}'_1 \circ \mathcal{R}'_0 &= \mathcal{G} \circ \mathcal{R}_{31} \circ \mathcal{G}^{-1} \circ \ldots \circ \mathcal{G} \circ \mathcal{R}_1 \circ \mathcal{G}^{-1} \circ \mathcal{G} \circ \mathcal{R}_0 \circ \mathcal{G}^{-1}\\ &= \mathcal{G} \circ \mathcal{R}_{31} \circ \ldots \circ \mathcal{R}_1 \circ \mathcal{R}_0 \circ \mathcal{G}^{-1}\,.\end{aligned}$$

Precisely, for the plaintext X and the corresponding ciphertext Y in the $\texttt{CRAFT}$ encryption, the plaintext $\mathcal{G}(X)$ will be encrypted to the ciphertext $\mathcal{G}(Y)$ in the equivalent cipher with $\mathcal{R}'_i$ round functions. By choosing $\mathcal{G}$ as

$$\mathcal{G}(X' \,\|\, X'') = \big(X' \,\|\, \texttt{SB}' \circ \texttt{PN}''(X'')\big) \;\Rightarrow\; \mathcal{G}^{-1}(X' \,\|\, X'') = \big(X' \,\|\, \texttt{PN}' \circ \texttt{SB}'(X'')\big),$$

it is possible to simplify the equivalent round functions:

$$\begin{aligned}&\mathcal{R}'_i(X' \,\|\, X'') = \mathcal{G} \circ \mathcal{R}_i \circ \mathcal{G}^{-1}(X' \,\|\, X'') = \mathcal{G} \circ \mathcal{R}_i\big(X' \,\|\, \texttt{PN}' \circ \texttt{SB}'(X'')\big)\\
&= \mathcal{G}\Big(\texttt{PN}'' \circ \texttt{SB}' \circ \texttt{A}'_{TK''_i} \circ \texttt{PN}' \circ \texttt{SB}'(X'') \,\|\, \texttt{PN}' \circ \texttt{SB}' \circ \texttt{A}'_{TK'_i} \circ \texttt{A}'_{RC_i}\big(X' \oplus \texttt{MC}' \circ \texttt{PN}' \circ \texttt{SB}'(X'')\big)\Big)\\
&= \mathcal{G}\Big(\texttt{SB}' \circ \texttt{A}'_{TK'''_i} \circ \texttt{PN}'' \circ \texttt{PN}' \circ \texttt{SB}'(X'') \,\|\, \texttt{PN}' \circ \texttt{SB}' \circ \texttt{A}'_{TK'_i} \circ \texttt{A}'_{RC_i}\big(X' \oplus \texttt{MC}' \circ \texttt{PN}' \circ \texttt{SB}'(X'')\big)\Big)\\
&= \mathcal{G}\Big(\texttt{SB}' \circ \texttt{A}'_{TK'''_i} \circ \texttt{SB}'(X'') \,\|\, \texttt{PN}' \circ \texttt{SB}' \circ \texttt{A}'_{TK'_i} \circ \texttt{A}'_{RC_i}\big(X' \oplus \texttt{MC}' \circ \texttt{PN}' \circ \texttt{SB}'(X'')\big)\Big)\\
&= \Big(\texttt{SB}' \circ \texttt{A}'_{TK'''_i} \circ \texttt{SB}'(X'') \,\|\, \texttt{SB}' \circ \texttt{PN}'' \circ \texttt{PN}' \circ \texttt{SB}' \circ \texttt{A}'_{TK'_i} \circ \texttt{A}'_{RC_i}\big(X' \oplus \texttt{MC}' \circ \texttt{PN}' \circ \texttt{SB}'(X'')\big)\Big)\\
&= \Big(\texttt{SB}' \circ \texttt{A}'_{TK'''_i} \circ \texttt{SB}'(X'') \,\|\, \texttt{SB}' \circ \texttt{SB}' \circ \texttt{A}'_{TK'_i} \circ \texttt{A}'_{RC_i}\big(X' \oplus \texttt{MC}' \circ \texttt{PN}' \circ \texttt{SB}'(X'')\big)\Big)\\
&= \Big(\texttt{SB}' \circ \texttt{A}'_{TK'''_i} \circ \texttt{SB}'(X'') \,\|\, \texttt{A}'_{TK'_i} \circ \texttt{A}'_{RC_i}\big(X' \oplus \texttt{MC}' \circ \texttt{PN}' \circ \texttt{SB}'(X'')\big)\Big)\,,\end{aligned}$$

where $TK'''_i = \texttt{PN}''(TK''_i)$. Note that this is the same round function as in Figure 2(b). Similarly for the last round, we have:

$$\mathcal{R}'_{31}(X' \,\|\, X'') = \Big(\texttt{A}'_{TK'_3} \circ \texttt{A}'_{RC_{31}}\big(X' \oplus \texttt{MC}' \circ \texttt{PN}' \circ \texttt{SB}'(X'')\big) \,\|\, \texttt{SB}' \circ \texttt{A}'_{TK'''_3} \circ \texttt{SB}'(X'')\Big)\,,$$

which is same as the other round functions $\mathcal{R}'_i$ without the final swapping between the left and the right halves of the state.

We provided an step by step approach of this proof with illustrations in Figure 6. □

Hereafter, we will simply call *equivalent* ***CRAFT*** to this equivalent representation of the cipher (the one depicted in Figure 2(b)). Note that the equivalent $\texttt{CRAFT}$ encryption is quite similar to the Feistel network. The only difference is in the transition of the right half of the state which is through $\texttt{SB}' \circ \texttt{A}_{TK'''_i} \circ \texttt{SB}'$ function while in the case of a Feistel network, this function is the identity function.

On the other hand, the equivalent $\texttt{CRAFT}$ provides a weak tweak-key structure for $\texttt{CRAFT}$. If the function $\texttt{SB}' \circ \texttt{A}_{TK'''_i} \circ \texttt{SB}'$ is equal to the identity function, the equivalent $\texttt{CRAFT}$ will be a Feistel network which includes partial nonlinear round and partial key-addition. Therefore, for such weak tweak-key classes, the equivalent $\texttt{CRAFT}$ might shows weaker resistance against some analysis, e.g., differential and linear attacks. Hereafter, we will simply call *weak tweak-key* ***CRAFT*** to this weak tweak-key encryption of the equivalent $\texttt{CRAFT}$ (the ones depicted in Figure 3). In the following, we discuss the necessary requirements for the weak tweak-key $\texttt{CRAFT}$.

Lemma 1. *$\mathtt{SB}' \circ \mathtt{A}'_{TK'''} \circ \mathtt{SB}'$ is equal to the identity function if and only if $TK''' = 0$.*

Proof. $\mathtt{SB}' \circ \mathtt{A}'_{TK'''} \circ \mathtt{SB}'$ is the identity function if and only if for any $X' \in \mathbb{F}_2^{32}$,

$$\mathtt{SB}' \circ \mathtt{A}'_{TK_i'''} \circ \mathtt{SB}'(X') = X' \,.$$

Since $\mathtt{SB}'$ is involution, this equally means for any X', we must have

$$\mathtt{A}'_{TK_i'''} \circ \mathtt{SB}'(X') = \mathtt{SB}'(X') \ \Leftrightarrow\ TK_i''' \oplus \mathtt{SB}'(X') = \mathtt{SB}'(X') \ \Leftrightarrow\ TK_i''' = 0 \,.$$

□

Lemma 2. *In the weak tweak-key CRAFT, with the given permutation Q in the cipher's specification, the 128-bit key must be one of 2^{88} weak keys, and for each weak-key there are exactly 2^8 64-bit tweaks those are included in the set of 2^{32} weak tweaks.*

Proof. As a corollary from 1, to achieve the weak tweak-key CRAFT, it is necessary to have $TK_i''' = 0$ for all rounds. This equally means $TK_0'' = TK_1'' = TK_2'' = TK_3'' = 0$. Simplifying these equations to the key and tweak variables, we reach to

$$K_0'' = K_1'' = T'' = \mathtt{QN}(T)[8, \ldots, 15] \,.$$

For the given Q permutation in the CRAFT specification, the above equation is same as

$$\begin{array}{llll}
K_0[8] = K_1[8] = T[8] = T[11] \,, & & K_0[9] = K_1[9] = T[9] = T[3] & , \\
K_0[10] = K_1[10] = T[10] = T[7] & , & K_0[11] = K_1[11] = T[11] = T[4] & , \\
K_0[12] = K_1[12] = T[12] = T[6] & , & K_0[13] = K_1[13] = T[13] = T[0] & , \\
K_0[14] = K_1[14] = T[14] = T[1] & , & K_0[15] = K_1[15] = T[15] = T[13] \,. &
\end{array}$$

This means that in the weak tweak-key CRAFT, the 128-bit key must be one of 2^{88} weak keys satisfying above equations, i.e., both $K_0'' = K_1''$ must be in the following form:

$$K_0'' = K_1'' = \big(x_0, x_1, x_2, x_0, x_3, x_4, x_5, x_4\big) \,.$$

Besides, the 64-bit tweak must be one of 2^{32} weak tweaks in the form of

$$T = \big(t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_4, t_3, t_7, t_4, t_6, t_0, t_1, t_0\big) \,.$$

Moreover, the weak key and the weak tweak must satisfy the followings:

$$x_0 = t_4 \,, \quad x_1 = t_3 \,, \quad x_2 = t_7 \,, \quad x_3 = t_6 \,, \quad x_4 = t_0 \,, \quad x_5 = t_1 \,.$$

Therefore, for each weak key, there is exactly 2^8 weak tweaks with freedom on the t_2 and t_5 nibbles, and in total there are $2^{88+32-24} = 2^{96}$ weak tweak-key pairs. □

The round function of the weak tweak-key CRAFT is shown in Figure 3(a) which by using an equivalent tweakey schedule changes to the Feistel round function shown in Figure 3(b).

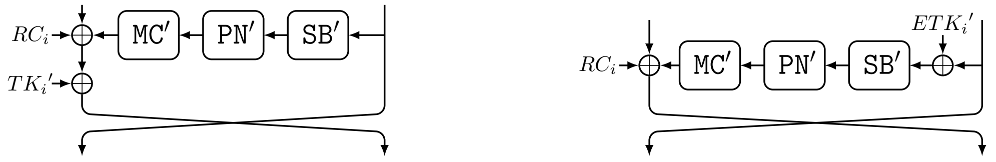
Lemma 3. *If all the right halves of the tweakeys in CRAFT encryption are equal to zero, i.e., all $TK_i'' = 0$ with $0 \leq i < 4$, then the encryption is equivalent to the Feistel network with the following round function*

$$\mathcal{R}_i(X', X'') = \big(X'', X' \oplus \mathcal{F}_i(X'' \oplus ETK_i)\big) \text{ with } \mathcal{F}_i := \mathtt{A}'_{RC_i} \circ \mathtt{MC}' \circ \mathtt{PN}' \circ \mathtt{SB}' \,,$$

where ETK_i is an equivalent tweakey. Moreover, the equivalent round tweakeys are

$$\begin{array}{llll}
ETK_0 = 0 \quad , & ETK_1 = K_0' \oplus T' \,, & ETK_2 = K_1' \oplus T' \,, & ETK_3 = T' \oplus T''', \\
ETK_4 = T' \oplus T''', & ETK_5 = K_0' \oplus T''', & ETK_6 = K_1' \oplus T''', & ETK_7 = 0
\end{array}$$

while for $i \geq 8$, we have $ETK_i = ETK_{i \bmod 8}$. Besides, T''' denotes the left half of $\mathtt{QN}(T)$, i.e., $T''' = \big(\mathtt{QN}(T)[0], \ldots, \mathtt{QN}(T)[7]\big)$.



(a)

(b)

Figure 3: The round function for weak tweak-key CRAFT.

Proof. The behavior of the key addition in the Feistel network is already studied well and it is known in the literature that the Feistel cipher with round functions defined by $\mathcal{R}_i(X', X'') = \big(X'', RK_i \oplus X' \oplus \mathcal{F}_i(X'')\big)$ is equal to the Feistel cipher with round functions of $\mathcal{R}'_i(X', X'') = \big(X'', X' \oplus \mathcal{F}_i(X'' \oplus ERK_i)\big)$ where for each i with $i > 1$, we have

$$ERK_i = RK_{i-1} \oplus ERK_{i-2}\,,$$

while $ERK_0 = 0, ERK_1 = RK_0$. Also, there is a whitening key in the ciphertext with value of ERK_{r-1} and ERK_r in the right and left halves, respectively.

The tweakey schedule in the weak tweak-key CRAFT uses four 32-bit round tweakeys, TK'_0, TK'_1, TK'_2 and TK'_3 repeatedly. For the equivalent round tweakeys, we would have the following eight round tweakeys:

$$\begin{array}{lllllll}
ETK_0 = & & = & & = & 0 & ,\\
ETK_1 = & & = & TK'_0 & = & K'_0 \oplus T' & ,\\
ETK_2 = & ETK_0 \oplus TK'_1 & = & TK'_1 & = & K'_1 \oplus T' & ,\\
ETK_3 = & ETK_1 \oplus TK'_2 & = & TK'_2 \oplus TK'_0 & = & T' \oplus T''' & ,\\
ETK_4 = & ETK_2 \oplus TK'_3 & = & TK'_3 \oplus TK'_1 & = & T' \oplus T''' & ,\\
ETK_5 = & ETK_3 \oplus TK'_0 & = & TK'_2 & = & K'_0 \oplus T''' & ,\\
ETK_6 = & ETK_4 \oplus TK'_1 & = & TK'_3 & = & K'_1 \oplus T''' & ,\\
ETK_7 = & ETK_5 \oplus TK'_2 & = & & = & 0 & ,
\end{array}$$

that are used repeatedly. For the whitening keys we have $ERK_{31} = ERK_{32} = 0$. □

As you see, half of the round tweakeys in the equivalent tweakey schedule are independent of the key value. More important, $ETK_0 = ETK_7 = 0$ makes the first and the last rounds of the weak tweak-key structure of CRAFT to be key-less rounds. This means that the security of 32-round weak tweak-key CRAFT cipher with Feistel network structure is based on the middle 30 rounds and the other two rounds are actually useless.

3.2 Effect of Q Permutation on the Size of Weak Key Set

The permutation Q plays an important role in the size of weak tweak or weak key sets. For an arbitrary choice for permutation Q, from Lemma 2, we know that the essential condition to achieve the weak tweak-key CRAFT is to have

$$\begin{array}{ll}
K_0[8] = K_1[8] = T[8] = T[Q[8]]\,, & K_0[9] = K_1[9] = T[9] = T[Q[9]]\,,\\
K_0[10] = K_1[10] = T[10] = T[Q[10]]\,, & K_0[11] = K_1[11] = T[11] = T[Q[11]]\,,\\
K_0[12] = K_1[12] = T[12] = T[Q[12]]\,, & K_0[13] = K_1[13] = T[13] = T[Q[13]]\,,\\
K_0[14] = K_1[14] = T[14] = T[Q[14]]\,, & K_0[15] = K_1[15] = T[15] = T[Q[15]]\,.
\end{array}$$

Note that in the design rationale for CRAFT, the only criterion for the permutation Q was to be a circulant one that the cipher can resist against time-data-memory trade-off attack. Based on this criterion and depending on the choice for permutation Q, the size for weak key set can be in the form of $2^{64+4\cdot\ell}$ (out of 2^{128} keys) with $1 \leq \ell \leq 8$.

The maximum size for the weak key set with $\ell = 8$ happens if the circulant permutation Q be in the form of that for any $i \in \{8, \ldots, 15\}$ then $Q[i] \in \{0, \ldots, 7\}$. We denote such a permutation by Q_{max} that there $8! \cdot 7!$ ones out of 15! circulant permutations. For a Q_{max} permutation, the size for weak key and weak tweak sets are 2^{96} and 2^{32}, respectively, that must satisfy eight conditions on the key and tweak nibbles (i.e., there are 2^{96} weak tweak-key values).

On the other hand, the minimum size for the weak key set with $\ell = 1$ happens if in the corresponding circle in graph representation of the permutation, the eight values in $\{8, \ldots, 15\}$ are consecutive. In other meaning, there exist an $i \in \{0, \ldots, 7\}$ such that for any j with $1 \leq j \leq 8$, we have $Q^j[i] \in \{8, \ldots, 15\}$. We show such a permutation by Q_{min} that there are $(8!)^2$ ones out of 15! circulant permutations. For a Q_{min} permutation, we have $T[8] = \ldots = T[15]$ which results in

$$T = (t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_i, t_i, \ldots, t_i, t_i) , \quad \text{and} \quad K_0'' = K_1'' = (t_i, t_i, \ldots, t_i, t_i) .$$

This means that the size for weak key and weak tweak sets are 2^{68} and 2^{32}, respectively, that must satisfy one condition on the key and tweak nibbles (i.e., there are 2^{96} weak tweak-key values).

Among all the 15! circulant permutations, the chosen Q permutation for CRAFT was taken from a set of one thousand randomly generated permutations that it is the one with most resistance against related-tweak differential attack. As a designer one can add another criterion for Q permutation that it is one of $(8!)^2$ Q_{min} permutations to minimize the size for weak-key set in the weak tweak-key CRAFT while (s)he must consider resistance of the cipher against related-tweak attacks.

4 Differential and Linear Analysis

In this section, first we use an MILP tool to find all the activity patterns with the minimum number of active S-boxes in reduced-round differential and linear trails of the weak tweak-key CRAFT structure. We apply the methods introduced in [ELR20] to estimate the EDP for the differential effect within these differential activity patterns. Then, we discuss the conditions for applying the same differential trails of the weak tweak-key CRAFT in the equivalent CRAFT structure. There, we show that the weak tweak-key space in the differential activity patterns of the equivalent CRAFT can be larger than the one for the weak tweak-key CRAFT. Later, we investigate possibility of applying related-tweak differentials in the weak tweak-key or in the equivalent CRAFT structures.

As a result, we present 18-round single-tweak differential activity patterns which provide multiple differentials with EDP of $2^{-58.11}$, and several 21-round related-tweak differential activity patterns that include multiple differentials with EDP of higher than 2^{-64}.

4.1 Minimum Number of Active S-boxes

To compare the resistance of original CRAFT and the weak tweak-key CRAFT against the differential and linear attacks, we compute the minimum number of active S-boxes in the single-tweak model. In order to compute these bounds, similar to the one in the CRAFT proposal paper [BLMR19], we use the MILP as explained in [MWGP11]. It is noteworthy to mention that this approach is independent of the specification of the S-box and it takes only the properties of the linear layer into account.

While for computing the minimum number of active S-boxes in the original CRAFT, in the MILP codes, for each round, we need to consider all the 16 corresponding variables to the S-box layer as objective variables, in the case for weak tweak-key CRAFT, we must consider only the 8 corresponding variables to the right half variables in the S-box layer.

Table 1: The minimum number of active S-boxes in differential and linear activity patterns in up to 32 rounds of CRAFT. n_1 and n_2 denote the numbers for original and weak tweak-key CRAFT, respectively.

r	1	2	3	4	5	6	7	8	$9 \leq r < 32$
n_1	1	2	4	6	10	14	20	26	$4 \cdot (r-1)$
n_2	0	1	2	3	4	7	10	13	$2 \cdot (r-1)$

Beyond this difference in two structures, by taking benefit of being a Feistel design, we slightly improved the MILP codes used in [BLMR19] to reduce the number of variables.

As it is already mentioned in [MWGP11], to find the the minimum number of active S-boxes for linear activity patterns, it is enough to replace the matrix of MC layer, M, with the corresponding inverse of transpose matrix, $(M^{-1})^T$ (which equals to M^T). The current choice for M causes that solving the equation to find the minimum number of active S-boxes with matrix M, to be the same as solving with M^T. This means that for a given number of rounds, the minimum number of active S-boxes in differential activity patterns is the same as the minimum number of active S-boxes in linear activity patterns.

Table 1 shows the minimum number of active S-boxes in the single-tweak model for up to 32 rounds in both differential and linear activity patterns. We use n_1 and n_2 to denote the numbers for original and weak tweak-key CRAFT, respectively. One interesting observation from Table 1 is that for most of number of rounds, n_2 is exactly half of n_1. Intuitively, this makes sense because in the weak tweak-key structure, each left half of the state is considered once, while in the original structure, each left half-state is considered twice: first in the left half-state of the current round and second in the right half-state of the next round.

While for the original CRAFT, after 9 rounds, all the numbers of active S-boxes are higher than or equal to 32, for weak tweak-key CRAFT, this happens after 17 rounds. Note that having at least 32 active S-boxes is important because the maximum differential probability (resp. absolute linear correlation) for an active S-box is 2^{-2} (resp. 2^{-1}) and this makes the probability (resp. absolute correlation) of a differential (resp. linear) characteristic to be less than or equal to 2^{-64} (resp. 2^{-32}). Therefore, such a characteristic cannot distinguish the (reduced-round) cipher from a random permutation.

4.2 Differential Effects

Finding the minimum number of active S-boxes considers only a single characteristic in the analysis. Therefore, the differential or linear distinguisher might actually be stronger due to the differential or linear hull effects, respectively. To have a better estimation about the strength of the differential distinguishers, we compute the EDP of the differentials. To this point, we use the MILP technique introduced in [SHW$^+$14] to find all the differential activity patterns with the minimum possible active S-boxes. Then, for each given differential activity pattern, we use the methods in [ELR20] to compute the EDP of the differentials within the activity pattern. That is by fixing the input and output differences in the differential, we consider all different single characteristics which follow the same activity pattern with the minimum number of active S-boxes. Then by summing all these probabilities of each single characteristics, we find a lower bound for the probability of corresponding differential. We repeat this computation for all different values for the input and the output differences in the differential to find the ones with the maximum EDP.

It is noteworthy to mention that the computed values for the EDPs are lower bounds, because for a fixed input and output difference, there might be some other single characteristics that are not following the S-box activity pattern. However, as for such characteristics

Table 2: The maximum EDP for the differentials within the activity patterns with minimum number of active S-boxes. Note we use $p = -\log_2 \text{EDP}$ instead of showing values for the EDP.

r	9	10	11	12	13	14	15	16	17	18	19
p	25.79	29.79	35.54	41.42	41.19	45.19	50.42	56.42	54.00	58.11	63.25

Table 3: 18-round differentials with the highest EDP ($= 2^{-58.11}$) within the differentials of the activity patterns with the minimum number of active S-boxes ($= 34$).

	trail 1	*trail 2*	*trail 3*	*trail 4*
	01010101 10100000	10101010 01010000	01010101 10100000	10101010 01010000
	10100000 00010000	01010000 10000000	10100000 01000000	01010000 00100000
	00010000 10000010	10000000 01000001	01000000 00101000	00100000 00010100
	10000010 10000001	01000001 00011000	00101000 00100100	00010100 01000010
	10000001 10000011	00011000 01000011	00100100 00101100	01000010 00011100
	10000011 00000000	01000011 00000000	00101100 00000000	00011100 00000000
	00000000 10000011	00000000 01000011	00000000 00101100	00000000 00011100
	10000011 10000001	01000011 00011000	00101100 00100100	00011100 01000010
	10000001 10000011	00011000 01000011	00100100 00101100	01000010 00011100
	10000011 00000000	01000011 00000000	00101100 00000000	00011100 00000000
	00000000 10000011	00000000 01000011	00000000 00101100	00000000 00011100
	10000011 10000001	01000011 00011000	00101100 00100100	00011100 01000010
	10000001 10000011	00011000 01000011	00100100 00101100	01000010 00011100
	10000011 00000000	01000011 00000000	00101100 00000000	00011100 00000000
	00000000 10000011	00000000 01000011	00000000 00101100	00000000 00011100
	10000011 10000001	01000011 00011000	00101100 00100100	00011100 01000010
	10000001 10000010	00011000 01000001	00100100 00101000	01000010 00010100
	10000010 00010000	01000001 10000000	00101000 01000000	00010100 00100000
	00010000 10100010	10000000 01010001	01000000 10101000	00100000 01010100
ΔP	0x0a0x0a a0y00000	a0x0a0x0 0a0y0000	0a0x0a0x a0y00000	x0a0x0a0 0a0y0000
ΔC	000y0000 a0z000t0	y0000000 0a0z000t	0a000000 z0y0t000	00a00000 0z0y0t00
	$t = y \oplus z$ and $(xyz) \in \{$(5,a,0), (7,5,f), (7,d,7), (a,5,f), (a,a,0), (a,d,7), (a,f,5), (d,a,0), (f,a,0), (f,f,5)$\}$		$t = z \oplus$ a , $x \in \{$5,a,d,f$\}$ and $(y, z) \in \{$(5,f), (a,0), (d,7), (f,5)$\}$	

the number of active S-boxes will be higher, we assume their affect on the probability of differential to be negligible.

Table 2 summarizes the maximum EDP for the differentials within the activity patterns with minimum number of active S-boxes up to 19 rounds. For 19 rounds and more, there is no differential within the activity patterns of minimum number of active S-boxes that has EDP of significantly higher than 2^{-64}. Note that in this table, instead of showing value of EDP, we use $p = -\log_2 \text{EDP}$.

The differentials of 18-round with the highest EDP ($= 2^{-58.11}$) within the differentials of the activity patterns with the minimum number of active S-boxes (34 S-boxes) are listed in Table 3 which are from four different activity patterns. Note that in these activity patterns, the active and the inactive nibbles of states are denoted by 1 and 0, respectively. Besides, the values of the input difference (ΔP) and the output difference (ΔC) are shown in the hexadecimal.

4.3 Enlarging Weak Tweak-Key Set in a Differential Activity Pattern

To achieve the Feistel round function of CRAFT (shown in Figure 3(b)), it is necessary to have $TK_i''' = 0$ for all i values which leads to 2^{88} weak keys (out of 2^{128}) and 2^{32} weak tweaks (out of 2^{64}) together with 24 bit conditions between them which leaves 2^{96} weak tweak-keys (out of 2^{196}). But for the differentials within an activity pattern, considering $TK_i''' = 0$ is a non necessary condition. Considering Figure 2(b), to assure that the differential probability of a differential transition over the right branch (over the $\mathtt{SB}' \circ \mathtt{A}_{TK_i'''} \circ \mathtt{SB}'$ operation) is equal 1, it is enough that only the nibbles of TK_i''' to be zero which are the corresponding nibbles to the active nibbles in the difference. This is because of the property of a bijective S-box which a zero difference in the input leads to zero difference in the output and vice versa.

Therefore, it is possible to use the same activity patterns found for the weak tweak-key CRAFT with Feistel round functions shown in Figure 3, also for the equivalent CRAFT with the round functions shown in Figure 2(b). To do this, we only need to consider weak tweak-keys which make the active nibbles of all TK_i'''s to be zero.

Example 1. Consider a differential distinguisher corresponding to the trail 1 from Table 3. This distinguisher works for equivalent CRAFT with the following weak tweak-key set:

$$\begin{array}{rr}
TK_0'''[0] = TK_0'''[2] = \mathtt{0}, & TK_{10}'''[0] = TK_{10}'''[6] = TK_{10}'''[7] = \mathtt{0},\\
TK_1'''[3] = \mathtt{0}, & TK_{11}'''[0] = TK_{11}'''[7] = \mathtt{0},\\
TK_2'''[0] = TK_2'''[6] = \mathtt{0}, & TK_{12}'''[0] = TK_{12}'''[6] = TK_{12}'''[7] = \mathtt{0},\\
TK_3'''[0] = TK_3'''[7] = \mathtt{0}, & TK_{14}'''[0] = TK_{14}'''[6] = TK_{14}'''[7] = \mathtt{0},\\
TK_4'''[0] = TK_4'''[6] = TK_4'''[7] = \mathtt{0}, & TK_{15}'''[0] = TK_{15}'''[7] = \mathtt{0},\\
TK_6'''[0] = TK_6'''[6] = TK_6'''[7] = \mathtt{0}, & TK_{16}'''[0] = TK_{16}'''[6] = \mathtt{0},\\
TK_7'''[0] = TK_7'''[7] = \mathtt{0}, & TK_{17}'''[3] = \mathtt{0},\\
TK_8'''[0] = TK_8'''[6] = TK_8'''[7] = \mathtt{0}, & TK_{18}'''[0] = TK_{18}'''[2] = \mathtt{0}.
\end{array}$$

Moreover, considering that $TK_{i+4} = TK_i$, all the above conditions can be combined in below conditions:

$$\begin{array}{rr}
TK_0'''[0] = TK_0'''[2] = TK_0'''[6] = TK_0'''[7] = \mathtt{0}, & TK_1'''[3] = \mathtt{0},\\
TK_2'''[0] = TK_2'''[2] = TK_2'''[6] = TK_2'''[7] = \mathtt{0}, & TK_3'''[0] = TK_3'''[7] = \mathtt{0}.
\end{array}$$

These conditions lead to 2^{48} weak tweaks (with four conditions of $T[0] = T[13] = T[15]$ and $T[4] = T[8] = T[11]$), and 2^{112} weak-keys (with four conditions of $K_0[8] = K_0[11] = K_1[11]$ and $K_0[13] = K_0[15] = K_1[15]$), with three extra conditions between the tweak and key nibbles. All together, the distinguisher works for a weak tweak-key set of size $2^{112+48-12} = 2^{148}$.

It is important to emphasize that by changing the index of starting round, the size for weak key set or weak tweak set can change while the size for weak tweak-key set stays the same. Besides, the size of the weak tweak-key set for other distinguishers from Table 3 is the same. Moreover, it is noteworthy to mention that there are other distinguishers with a larger size for the weak key or weak tweak-key sets, but with a lower value for the EDP.

By appending some rounds before and after the distinguisher, one can use these distinguishers to do a key recovery attack on the reduced-round CRAFT. For all the distinguishers in Table 3, it is possible to append at most 3 (resp. 5) rounds before (resp. after) the distinguisher. Therefore, it is possible to do a key recovery attack on at most 26 rounds of CRAFT cipher with such a weak key, but notice that the time complexity of the attack (which must not exceed 2^{112} encryptions) is not considered here.

4.4 Related-Tweak Differentials in the Weak Tweak-Key CRAFT

While the differentials discussed in the previous subsections were based on a single-tweaks, in this subsection, we investigate the possibility for existence of related-tweak differentials

Table 4: The minimum number of active S-boxes and the maximum EDP for the related-tweak differentials within the activity patterns with the minimum (or close to the minimum) number of active S-boxes. Note that RT_i refers to the characteristic starting with round round $4 \cdot j + i$, and also, we use $p = -\log_2 \mathrm{EDP}$ instead of showing the value for EDP.

	minimum number of active S-boxes					highest expected differential probability				
r	18	19	20	21	22	18	19	20	21	22
RT_0	31	33	36	37	38	46.64	49.54	57.09	60.83	66.00
RT_1	32	35	36	37	40	47.54	55.54	56.00	62.25	64.96
RT_2	34	35	36	39	42	54.45	54.45	56.45	60.30	64.30
RT_3	32	33	36	39	40	47.54	49.54	55.54	63.54	66.25

in the weak tweak-key CRAFT.

As explained previously in Section 3 and Subsection 4.3, it is possible that any difference α in the input of $S\big(S(\cdot) \oplus t\big)$ transits to the same difference in the output of the function, if α is equal to zero or if the round tweakey nibble t is equal to zero. Hence, it is still possible to consider existence of related-tweak differentials, if we keep these conditions, precisely, if the difference in the right half of the round tweakeys, TK_i'', (consequently, in the right half of the round tweaks T_i'') is equal to zero. Therefore, by considering $\Delta T_0'' = \Delta T_1'' = \Delta T_2'' = \Delta T_3'' = 0$, we can check if there is any possibility for related-tweak differential in the weak tweak-key CRAFT.

$$\begin{cases} \Delta T_0'' = \Delta T_1'' = 0 \ \Rightarrow \ \Delta T[8,9,10,11,12,13,14,15] = 0 \\ \Delta T_2'' = \Delta T_3'' = 0 \ \Rightarrow \ \Delta T[0,1,3,4,6,7,11,13] = 0\,. \end{cases}$$

This means that there is still freedom in choosing the difference in two nibbles of tweak, namely $\Delta T[2]$ and $\Delta T[5]$. Considering that $\Delta T[2] = x$ and $\Delta T[5] = y$, the difference in the left half of the round tweaks would be equal to $\Delta T_0' = \Delta T_1' = (0,0,x,0,0,y,0,0)$ and $\Delta T_2' = \Delta T_3' = (0,0,0,y,0,0,0,x)$.

Similar to the single-tweak differential model, we use an MILP tool to find all the activity patterns with the minimum and close to the minimum number of active S-boxes. We show the minimum number of active S-boxes in the related-tweak model of the weak tweak-key CRAFT for 18 up to 22 rounds in Table 4. Note that in the related-tweak model, the differentials are dependent on the starting round. For this reason, we use the index of RT to show the starting round.

One interesting observation in searching the activity patterns is that for most of them, there is no differential characteristics that can follow the activity pattern. It is important to mention that this is independent of the choice of S-box, and it is only because that a set of linear equations must be satisfied between the variables of difference in the input and output of the active S-boxes and also the active nibble(s) in the tweak difference. Therefore, before using the activity pattern to find the differentials with high EDP value, we use an algorithm to sieve all the activity patterns which lead to an invalid set of linear equations. Briefly explaining, in this algorithm for a given activity pattern, we consider a variable for input difference and one another for output difference of each active S-box, and also one variable for each active nibble in the tweak. Then, based on the relations in the linear layer of the round function, we build a set of linear equations that must be satisfied for the given pattern. Each of these equations includes some variables which their XOR sum must be equal to zero. After applying Gaussian elimination to this set of equations, there must be no equation with a single variable. Otherwise, it means that this single variable is equal to zero and this contradicts the activity of this variable. Hence, this algorithm can efficiently determine if the given activity pattern is a valid one.

After filtering the useful activity patterns, we again use the method of [ELR20] to

Table 5: 21-round related-tweak differentials with the highest EDP within the differentials of the activity patterns with the minimum (or close to the minimum) number of active S-boxes.

	trail 1 with ST_0	*trail 2 with* ST_1	*trail 3 with* ST_2	*trail 4 with* ST_3
	00110001 10000000	10100011 00010010	00010011 10000000	11101101 01100100
	10000000 10000000	00010010 00000001	10000000 00000011	01100100 00100000
	10000000 10000010	00000001 00000011	00000011 00010001	00100000 00000000
	10000010 10000000	00000011 10010000	00010001 00010001	00000000 00000000
	10000000 10010010	10010000 00010001	00010001 00000011	00000000 00000001
	10010010 00010011	00010001 10000010	00000011 10000000	00000001 00010001
	00010011 10000000	10000010 10000000	10000000 00010011	00010001 00010011
	10000000 00000011	10000000 10010010	00010011 10010010	00010011 10000011
	00000011 00010001	10010010 00010011	10010010 10000000	10000011 10000011
	00010001 00010001	00010011 10000000	10000000 10000010	10000011 00010011
	00010001 00000011	10000000 00000011	10000010 00010001	00010011 00010001
	00000011 10000000	00000011 00010001	00010001 10010000	00010001 00000001
	10000000 00010011	00010001 00010001	10010000 00000011	00000001 00000000
	00010011 10010010	00010001 00000011	00000011 10000001	00000000 00000000
	10010010 10000000	00000011 10000000	10000001 00010011	00000000 00100000
	10000000 10000010	10000000 00010011	00010011 00010011	00100000 01100100
	10000010 00010001	00010011 10010010	00010011 10000001	01100100 10101101
	00010001 10010000	10010010 10000000	10000001 00000011	10101101 01000000
	10010000 00000011	10000000 10000010	00000011 00010000	01000000 00000101
	00000011 00000001	10000010 00010000	00010000 00000001	00000101 00110000
	00000001 00010010	00010000 10000000	00000001 00100000	00110000 01100010
	00010010 10100011	10000000 00110001	00100000 01000100	01100010 11111101
ΔT	00x00000 00000000	00x00000 00000000	00a00000 00000000	00a00000 00000000
ΔP	00xy000z a0000000	y0z000tx 000a00x0	000x00ax y0000000	xya0xa0a 0za00z00
ΔC	000w00x0 t0u000vx	v0000000 00xa000u	00a00000 0a000a00	0au000v0 pqastq0a

compute the EDP of differentials within the activity pattern and find the differentials with the highest EDP. We recall that in this method by fixing the input and output differences of the differential together with the tweak difference, we consider all the different single characteristics which follow the same activity pattern with the minimum number of active S-boxes. Table 4 summarizes the maximum EDP for the differentials within the activity patterns with minimum (or close to minimum) number of active S-boxes for 18 up to 22 rounds and for different index of starting round. For 22 rounds and more, we did not find any differentials within the activity patterns with the minimum (or close to the minimum) number of active S-boxes that has EDP of significantly higher than 2^{-64}. Note that in this table, instead of showing value of the EDP, we use $p = -\log_2 \mathrm{EDP}$.

As a result, we find several 21-round differentials with EDP higher than 2^{-64} for different index of RT. Precisely, for each RT_i, we find only one activity pattern that includes several differentials with the highest possible EDP. These highest EDP values are $2^{-60.83}$, $2^{-62.25}$, $2^{-60.30}$ and $2^{-63.54}$ for RT_0, RT_1, RT_2 and RT_3, respectively, that are listed in Table 5. Note that in these activity patterns, the active and the inactive nibbles of states are denoted by 1 and 0, respectively. Besides, the values for input difference (ΔP), output difference (ΔC) and tweak difference (ΔT) are shown in hexadecimal. Moreover, since there are many choices for the differential with the highest possible EDP, we simply denote the variables by $p, q, \ldots, y, z$ to show how the active nibbles in $\Delta P, \Delta C$ and ΔT are related. It might be interesting to mention that the number of differentials with the highest EDP are 2688, 16, 24 and 250000, for trail 1, 2, 3 and 4, respectively.

About the weak tweak-key sets for the given related-tweak differentials in Table 5, we use the same approach as in Example 1 to find the conditions between the key and tweak nibbles. For both trail 1 and trail 2, we need to have

$$\begin{aligned} &TK_0'''[0] = TK_0'''[3] = TK_0'''[6] = TK_0'''[7] = 0, && TK_2'''[0] = TK_2'''[6] = TK_2'''[7] = 0, \\ &TK_1'''[0] = TK_1'''[3] = TK_1'''[6] = TK_1'''[7] = 0, && TK_3'''[0] = TK_3'''[6] = TK_3'''[7] = 0, \end{aligned}$$

which leads to

$$\begin{aligned} K_0[8] = K_1[8] = K_0[11] = K_1[11] = T[4] &= T[8] = T[11]\,, \\ K_0[15] = K_1[15] = T[13] &= T[15]\ , \\ K_0[14] = K_1[14] &= T[14]\ . \end{aligned}$$

For trail 3, we need to have

$$\begin{aligned} &TK_1'''[0] = TK_1'''[3] = TK_1'''[6] = TK_1'''[7] = 0, && TK_0'''[3] = TK_0'''[7] = 0, \\ &TK_2'''[0] = TK_2'''[2] = TK_2'''[6] = TK_2'''[7] = 0, && TK_3'''[0] = TK_3'''[6] = TK_3'''[7] = 0, \end{aligned}$$

that is same as

$$\begin{aligned} K_0[8] = K_1[8] = K_0[11] = K_1[11] = T[4] &= T[8] = T[11]\,, \\ K_0[15] = K_1[15] = T[13] &= T[15]\ , \\ K_0[14] = K_1[14] &= T[14]\ , \\ K_0[13] &= T[0]\,. \end{aligned}$$

And for trail 4, we need to have

$$\begin{aligned} TK_0'''[1] = TK_0'''[2] = TK_0'''[3] = TK_0'''[6] = TK_0'''[7] &= 0, \\ TK_1'''[2] = TK_0'''[3] = TK_0'''[5] = TK_0'''[6] = TK_0'''[7] &= 0, \\ TK_2'''[0] = TK_2'''[1] = TK_2'''[2] = TK_2'''[3] = TK_2'''[5] = TK_2'''[6] = TK_2'''[7] &= 0, \\ TK_3'''[0] = TK_3'''[1] = TK_3'''[2] = TK_3'''[3] = TK_3'''[5] = TK_3'''[6] = TK_3'''[7] &= 0, \end{aligned}$$

which is equal to

$$\begin{aligned} K_0[8] = K_1[8] = K_0[11] = K_1[11] = T[4] = T[8] &= T[11]\,, \\ K_0[13] = K_1[13] = K_0[15] = K_1[15] = T[0] &= T[13]\ , \\ K_0[12] = K_1[12] = T[6] &= T[12]\ , \\ K_0[14] = K_1[14] = T[1] &= T[14]\ , \\ K_0[9] = K_1[9] = T[3] &= T[9]\,, \\ K_1[10] &= T[7]\ . \end{aligned}$$

All together, the size for weak keys, weak tweaks, and weak tweak-keys sets are 2^{108}, 2^{52} and 2^{148} for trail 1, 2 and 3, and 2^{92}, 2^{40} and 2^{108} for trail 4.

In order to do a key recovery attack, the attacker can append several rounds before and after the above 21-round related tweak differentials. But, it should be taken into account that the number of appended rounds before the differential starting with round i, RT_i, is either i or $i + 4$. For the above differentials, appending more than 4 rounds actives all the nibbles at the plaintext state. Hence, for the differentials with RT_i we can append i rounds in before the differential. However, there is no such a restriction on the number of rounds to append after the differentials and it is possible to extend those differentials by adding at most 3, 2, 3 and 3 rounds for key recovery. Therefore, it is possible that an attacker have a successful related-tweak differential attack on at most 28, 25, 27 and 27 rounds, respectively. It is noteworthy that here, we did not consider the time complexity for the key recovery to check the feasibility of the attack. Hence, the number of rounds that can be analyzed by the attacker is an upper bound.

Note that even though, we only analyzed security of the weak tweak-key CRAFT against differential cryptanalysis, but we believe that it can be applied to improve the result against other attacks, such as impossible differential, (zero-correlation) linear hull, meet-in-the-middle, and integral. However, in contrary to [GSS+20], we keep the general weakness as it is and do not specialize it only for differential attack.

4.5 Differential Properties of $S^*_c := S\big(S(\cdot) \oplus c\big)$

We already mentioned that since S is an involution, S^*_0 is the same as identity function. This makes it possible for any input difference of $\alpha \in \mathbb{F}_2^4$ to transit to the same difference in the output of S-box S^*_0 with probability 1. Here, we study the probability for transition of an input difference α to the same difference in the output of S^*_c for non-zero values of $c \in \mathbb{F}_2^4$. Table 6 shows the number of $x \in \mathbb{F}_2^4$ such that for each given c and α, $S^*_c(x \oplus \alpha) \oplus S^*_c(x) = \alpha$.

Interestingly, there are some high values in this table. Specially, there are two non-zero values for c and α pair which the input difference α stays the same in the output with probability 1, namely for $c = \alpha = \mathtt{2}$ and $c = \alpha = \mathtt{a}$. For our application, this means that even if the corresponding nibble of the round tweakey for an active S-box of S^* is not zero (i.e., the tweakey value is not included in the weak tweak-key space), it may lead to a high EDP. But this EDP is always smaller than the one we computed for the weak tweak-keys (which this nibble of the round tweakey is necessarily zero) and this is because of the restriction in the values of the input/output difference of S^* S-box. For instance, in case of $c = \mathtt{a}$, while only the input difference value of a can transit with probability 1, input difference with a value in $\{\mathtt{5}, \mathtt{7}, \mathtt{d}, \mathtt{f}\}$ can also transit to the same difference in the output with probability 2^{-1}. Therefore, the differentials discussed in the previous sections are not only useful for the tweakeys within the weak tweak-key spaces, but it might be possible to be applied for the tweakeys out of the weak tweak-key spaces with a smaller EDP.

It is noteworthy to mention that the transition probability for $c = \alpha = \mathtt{a}$, previously was observed and applied in [GSS+20] to enlarge the weak tweak-key space. While their technique makes it possible to enlarge the weak tweak-key space, it fixes the difference value in some intermediate nibbles. Therefore, the EDP of the differential gets smaller in favor of making the weak tweak-key space larger. This can be used as a trade-off between the EDP and the size of weak tweak-key space, which in case of a key recovery attack, both of these parameters affect number of needed plaintext differential pairs. Hence, the attacker can take advantage of this to reduce the data or time complexity of the attack.

Table 6: Number of entries x for $S^*_c(x \oplus \alpha) \oplus S^*_c(x) = \alpha$.

		α														
		1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
c	1	2	4	0	6	2	6	0	2	0	0	0	0	0	2	0
	2	4	16	4	4	0	4	0	0	4	0	4	4	0	4	0
	3	0	4	0	6	0	4	2	2	2	0	0	0	2	0	2
	4	6	4	6	2	2	0	0	2	0	0	2	0	0	0	0
	5	2	0	0	2	4	0	4	0	2	8	0	2	4	0	4
	6	6	4	4	0	0	0	2	2	0	0	0	2	2	0	2
	7	0	0	2	0	4	2	4	0	0	8	2	0	4	2	4
	8	2	0	2	2	0	2	0	0	2	0	2	2	0	2	0
	9	0	4	2	0	2	0	0	2	6	0	6	2	0	0	0
	a	0	0	0	0	8	0	8	0	0	16	0	0	8	0	8
	b	0	4	0	2	0	0	2	2	6	0	4	0	2	0	2
	c	0	4	0	0	2	2	0	2	2	0	0	6	0	6	0
	d	0	0	2	0	4	2	4	0	0	8	2	0	4	2	4
	e	2	4	0	0	0	0	2	2	0	0	0	6	2	4	2
	f	0	0	2	0	4	2	4	0	0	8	2	0	4	2	4

5 Differential Key Recovery Attack

In this section, we describe one single-tweak weak-key and one related-tweak weak key differential key recovery attack on respectively 23- and 26-round CRAFT block cipher that is based on the weak tweak-key CRAFT. For the single-tweak weak key attack, we apply some of the differentials with the highest EDP in one of the 18-round trails (entitled trail 1 at Table 3). By appending three rounds before (in the plaintext side) and two rounds after (in the ciphertext side) the trail, we extend it to a 23-round differential key recovery attack. In this attack, we recover the weak key (out of 2^{112} weak keys) using 2^{74} chosen plaintext/tweak/ciphertext tuples, in about 2^{94} time of 23-round CRAFT encryptions and with 2^{51} blocks of memory usage.

For the related-tweak weak-key attack, we apply some of the differentials with the highest EDP in the 21-round trail starting with RT_0 (see Table 5). By appending four rounds before and one round after the trail, we extend it for a 26-round differential key recovery attack. In this attack, we recover the weak key (out of 2^{108} weak keys) using 2^{73} chosen plaintext/tweak/ciphertext tuples, in about 2^{105} time of 26-round CRAFT encryptions and with 2^{60} blocks of memory usage.

It is important to mention that it is possible to reduce the complexity of attack, by reducing number of the appending rounds. In the following subsections, we describe the form of weak tweak-key sets, the attack procedures, and complexity of the attacks in detail.

5.1 Single-Tweak Differential Attack on 23-Round CRAFT

The highest EDP for differentials within the trail 1 of Table 3, is $2^{-58.11}$ and it happens for 10 differentials. Considering that the trail is staring at the beginning of third round, the value of these differences are shown below.

$$\Delta X_3 = (\mathtt{0}x\mathtt{0a}\ \ \mathtt{0}x\mathtt{0a}\ \ \mathtt{a0}y\mathtt{0}\ \ \mathtt{0000})\,, \quad \Delta X_{21} = (\mathtt{000}y\ \ \mathtt{0000}\ \ \mathtt{a0}z\mathtt{0}\ \ \mathtt{00}t\mathtt{0})\,,$$

with $t = y \oplus z$ and $(xyz) \in \{(\mathtt{5,a,0}), (\mathtt{7,5,f}), (\mathtt{7,d,7}), (\mathtt{a,5,f}), (\mathtt{a,a,0}), (\mathtt{a,d,7}), (\mathtt{a,f,5}), (\mathtt{d,a,0}), (\mathtt{f,a,0}), (\mathtt{f,f,5})\}$. To achieve this 18-round single-tweak differentials, we need to have following weaknesses in the tweakey schedule of (equivalent) CRAFT:

$$\begin{aligned} TK_0'''[3] &= \mathtt{0}, & TK_1'''[0] &= TK_1'''[2] = TK_1'''[6] = TK_1'''[7] = \mathtt{0},\\ TK_2'''[0] = TK_2'''[7] &= \mathtt{0}, & TK_3'''[0] &= TK_3'''[2] = TK_3'''[6] = TK_3'''[7] = \mathtt{0}. \end{aligned}$$

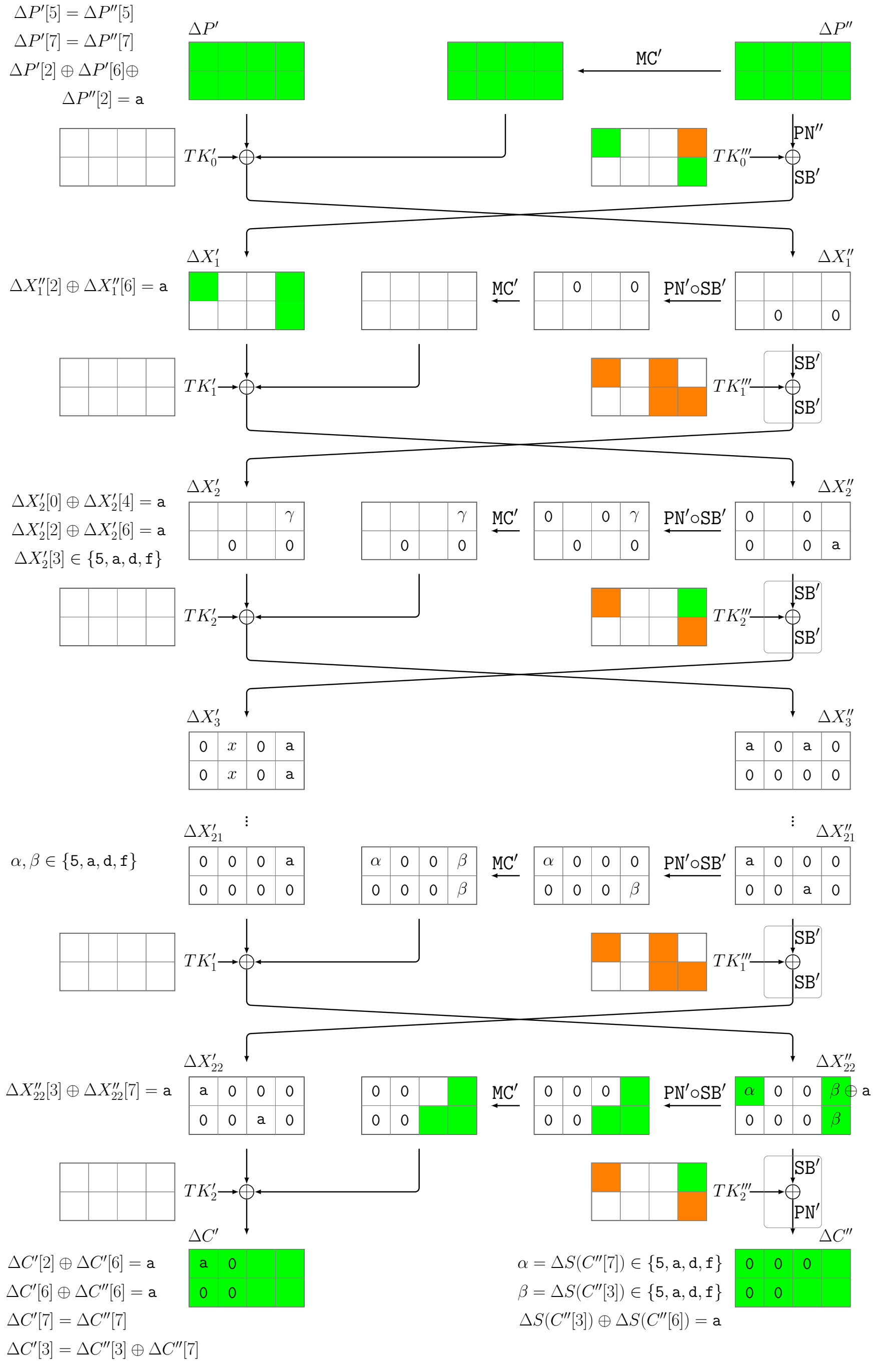
Note that since we start the trail after three rounds we need to shift the indices for round tweakeys from what we had in Example 1. These conditions lead to 2^{48} weak tweaks (with four conditions of $T[0] = T[13] = T[15]$ and $T[4] = T[8] = T[11]$), and 2^{112} weak-keys (with four conditions of $K_0[11] = K_1[8] = K_1[11]$ and $K_0[15] = K_1[13] = K_1[15]$), with three extra conditions between the tweak and key nibbles. All together, the distinguisher works for a weak tweak-key set of size $2^{112+48-12} = 2^{148}$. Therefore, the weak tweaks and weak keys will be in the following forms:

$$\begin{aligned} K_0 &= (k_0,\, k_1,\, k_2,\, k_3,\, k_4,\, k_5,\, k_6,\, k_7,\, k_8,\, k_9,\, k_{10},\, k_8',\, k_{12},\, k_{13},\, k_{14},\, k_{13}')\,,\\ K_1 &= (k_0',\, k_1',\, k_2',\, k_3',\, k_4',\, k_5',\, k_6',\, k_7',\, k_8',\, k_9',\, k_{10}',\, k_8',\, k_{12}',\, k_{13}',\, k_{14}',\, k_{13}')\,,\\ T\ &= (\,t_0,\ t_1,\ t_2,\ t_3,\ t_4,\ t_5,\ t_6,\ t_7,\ t_4,\ t_9,\ t_{10},\ t_4,\ t_{12},\ \ t_0,\ t_{14},\ \ t_0)\,, \end{aligned}$$

together with three conditions between key and tweak: $k_8' = t_4,\ k_{13}' = t_0,\ k_{14} = t_{14}$.

Since, the values for key nibbles k_8', k_{13}' and k_{14} are unknown to the attacker, it is not possible to check their equality with t_4, t_0 and t_{14}, respectively. However, the attacker can repeat the key recovery attack for all 2^{12} different values of t_4, t_0 and t_{14} and observe for which value of these three nibbles, the differential distinguisher occurs as it is expected. In other words, each of these differential will occur with probability of $2^{-58.11}$, if $k_8' = t_4,\ k_{13}' = t_0,\ k_{14} = t_{14}$; otherwise, it occurs with probability of about 2^{-64}. Besides, note that due to the tweakey schedule of CRAFT, in the weak tweak-key set for this trail, we know the values for $TK_0'''[0], TK_0'''[7]$ and $TK_2'''[3]$ round tweakey nibbles.

To reduce the effect of differential expansion in extending the trails, we only use the ones with $y = \mathtt{a}$ and $z = \mathtt{0}$. Extending these differentials by three rounds in the beginning activates all the nibbles in plaintext difference, with three linear conditions in the difference



$\Delta P'[5] = \Delta P''[5]$
$\Delta P'[7] = \Delta P''[7]$
$\Delta P'[2] \oplus \Delta P'[6] \oplus$
$\Delta P''[2] = \mathtt{a}$
$\Delta P'$
$\Delta P''$
MC'
TK_0'
TK_0'''
PN''
SB'
$\Delta X_1'$
$\Delta X_1''$
$\Delta X_1''[2] \oplus \Delta X_1''[6] = \mathtt{a}$
MC'
PN'∘SB'
TK_1'
TK_1'''
SB'
SB'
$\Delta X_2'$
$\Delta X_2''$
$\Delta X_2'[0] \oplus \Delta X_2'[4] = \mathtt{a}$
$\Delta X_2'[2] \oplus \Delta X_2'[6] = \mathtt{a}$
$\Delta X_2'[3] \in \{\mathtt{5}, \mathtt{a}, \mathtt{d}, \mathtt{f}\}$
γ
MC'
PN'∘SB'
TK_2'
TK_2'''
SB'
SB'
$\Delta X_3'$
$\Delta X_3''$
x
⋮
$\Delta X_{21}'$
$\Delta X_{21}''$
$\alpha, \beta \in \{\mathtt{5}, \mathtt{a}, \mathtt{d}, \mathtt{f}\}$
α
β
MC'
PN'∘SB'
TK_1'
TK_1'''
SB'
SB'
$\Delta X_{22}'$
$\Delta X_{22}''$
$\Delta X_{22}''[3] \oplus \Delta X_{22}''[7] = \mathtt{a}$
$\beta \oplus \mathtt{a}$
MC'
PN'∘SB'
TK_2'
TK_2'''
SB'
PN'
$\Delta C'$
$\Delta C''$
$\Delta C'[2] \oplus \Delta C'[6] = \mathtt{a}$
$\Delta C'[6] \oplus \Delta C''[6] = \mathtt{a}$
$\Delta C'[7] = \Delta C''[7]$
$\Delta C'[3] = \Delta C''[3] \oplus \Delta C''[7]$
$\alpha = \Delta S(C''[7]) \in \{\mathtt{5}, \mathtt{a}, \mathtt{d}, \mathtt{f}\}$
$\beta = \Delta S(C''[3]) \in \{\mathtt{5}, \mathtt{a}, \mathtt{d}, \mathtt{f}\}$
$\Delta S(C''[3]) \oplus \Delta S(C''[6]) = \mathtt{a}$

Figure 4: Extending the first 18-round activity pattern from Table 3 by three rounds before and two rounds after the trail.

of plaintext nibbles:

$$\Delta P[5] = \Delta P[13]\,, \quad \Delta P[7] = \Delta P[15] \quad \text{and} \quad \Delta P[2] \oplus \Delta P[6] \oplus \Delta P[10] = \texttt{a}\,.$$

On the other side, extending the differentials by two rounds, activates only eight nibbles in the ciphertext difference, with one constant, four linear and three nonlinear conditions in the difference of ciphertext nibbles:

$$\Delta C[2] = \Delta C[14]\,,\ \Delta C[7] = \Delta C[15]\,,\ \Delta C[2] \oplus \Delta C[6] = \texttt{a}\,,\ \Delta C[3] \oplus \Delta C[11] \oplus \Delta C[15] = \texttt{0}\,,$$
$$\Delta S(C[11]) = \Delta S(C[14]) = \texttt{a}\,,\ \Delta S(C[11]) \in \{\texttt{5},\texttt{a},\texttt{d},\texttt{f}\} \text{ and } \Delta S(C[15]) \in \{\texttt{5},\texttt{a},\texttt{d},\texttt{f}\}\,.$$

Figure 4 depicts details of the extensions in both directions. In this figure, the values written in each cell shows the difference in the corresponding nibble. Besides, the nibbles in the round state or round tweakey whose value are known, are shown in green color, while the round tweakey nibbles with value of zero are shown in orange color.

As mentioned before, to recover the whole key, we need to repeat the attack for each 2^{12} values of t_0, t_4 and t_{14}. Therefore, we first fix eight tweak nibbles, $t_1, t_2, t_3, t_5, t_6, t_7, t_9$, t_{10}, t_{13}, to zero and then choose value for t_0, t_4, t_{14}.

Collecting Data Structures For each values of t_0, t_4, t_{14}, we choose 2^{62} random plaintexts $(p_0, \ldots, p_{15})$ and ask for their encryption $(c_0, \ldots, c_{15})$ using the weak tweak $(t_0, \texttt{0}, \texttt{0}, \texttt{0}, t_4, \texttt{0}, \texttt{0}, \texttt{0}, t_4, \texttt{0}, \texttt{0}, t_4, \texttt{0}, t_0, t_{14}, t_0)$. Considering value of $a_0 = p_5 \oplus p_{13}$, $a_1 = p_7 \oplus p_{15}$ and $a_2 = p_2 \oplus p_6 \oplus p_{10}$ for each plaintext, we separate the collected data to 2^{12} sets which each set contains only the data with the same value for (a_0, a_1, a_2). In average, there are 2^{50} plaintext/ciphertext pairs. Using any plaintext/ciphertext pair of set (a_0, a_1, a_2) together with any plaintext/ciphertext pair of set $(a_0, a_1, a_2 \oplus \texttt{a})$ provides a differential pair with $\Delta p_5 = \Delta p_{13}$, $\Delta p_7 = \Delta p_{15}$ and $\Delta p_2 \oplus \Delta p_6 \oplus \Delta p_{10} = \texttt{a}$. Hence, there are about $2^{12} \cdot 2^{50 \cdot 2 - 1} = 2^{111}$ differential pairs with an appropriate plaintext difference.

The probability that one of these differential pairs lead to one of those four differences in ΔX_3 is $4 \cdot 2^{-52} = 2^{-50}$. In other meaning, in average, there are about 2^{61} differential pairs with the correct difference in ΔX_3 and about $2^{61-58.11} \approx 7$ differential pairs with the correct difference in ΔX_{21}. Therefore, for each t_0, t_4, t_{14} value, the attacker needs to use 2^{62} plaintext/ciphertext pairs (2^{111} differential pairs) to insure that for the case that $k'_8 = t_4,\ k'_{13} = t_0,\ k_{14} = t_{14}$, there are about 7 of right differential pairs with the corresponding ΔX_{21}.

Filtering Wrong Differential Pairs On the other side of the differential pairs, the attacker can use the ciphertext differentials to sieve the potentially right differential pairs. These conditions are listed below.

$$\Delta c_0 = \texttt{a}\,,\ \Delta c_1 = \Delta c_4 = \Delta c_5 = \Delta c_8 = \Delta c_9 = \Delta c_{10} = \Delta c_{12} = \Delta c_{13} = \texttt{0}\,,$$
$$\Delta c_2 = \Delta c_{14}\,,\ \Delta c_7 = \Delta c_{15}\,,\ \Delta c_2 \oplus \Delta c_6 = \texttt{a}\,,\ \Delta c_3 \oplus \Delta c_{11} \oplus \Delta c_{15} = \texttt{0}\,,$$
$$\Delta S(c_{11}) = \Delta S(c_{14}) = \texttt{a}\,,\ \Delta S(C[11]) \in \{\texttt{5},\texttt{a},\texttt{d},\texttt{f}\} \text{ and } \Delta S(C[15]) \in \{\texttt{5},\texttt{a},\texttt{d},\texttt{f}\}\,.$$

All together, for each t_0, t_4, t_{14} value, after this filtering, from all 2^{111} differential pairs, there will be only $2^{111-60} = 2^{51}$ of them left. We keep all these remaining differential pairs in a memory to use them in the key recovery step.

Recovering Key Nibbles Apart from k'_8, k'_{13}, k_{14}, the attacker can recover other 22 nibbles of the key, namely $K'_0[0,1,2,3,4,5,7]$, $K'''_0[1,2,4,5,6]$, $K'_1[0,1,2,3,5,7]$ and $K'''_1[1,3,4]$, by using the following 16 equations which are based on the partial encryption or decryption of the differential pairs. Note that only three nibbles of the weak key remain unknown, $k'_4 = K'_1[4,6], k'_6 = K'_1[6]$ and $k'_9 = K'''_1[5]$.

1. $\Delta X'_2[0] \oplus \Delta X'_2[4] = \mathtt{a}$ using $K'_0[4]$ and $K'''_1[4]$
2. $\Delta X'_2[3] \in \{\mathtt{5}, \mathtt{a}, \mathtt{d}, \mathtt{f}\}$ using $K'_0[3]$ and $K'''_1[3]$

3. $\Delta X''_2[0] = 0$ using $K'_0[1, 6]$
4. $\Delta X''_2[2] = 0$ using $K'_0[3, 4]$ and $K'''_0[2]$
5. $\Delta X''_2[4] = 0$ using $K'_0[1]$ and $K'''_0[4]$
6. $\Delta X''_2[6] = 0$ using $K'_0[3]$ and $K'''_0[6]$
7. $\Delta X''_2[7] = \mathtt{a}$ using $K'_0[0]$

8. $\Delta X'_3[1] = x$ using $K'_0[2, 5]$, $K'''_0[1]$ and $K'_1[1]$
9. $\Delta X'_3[3] = \mathtt{a}$ using $K'_0[0, 7]$ and $K'_1[3]$
10. $\Delta X'_3[5] = x$ using $K'_0[2]$, $K'''_0[5]$ and $K'_1[5]$

11. $\Delta X''_3[1] = \mathtt{0}$ using $K'_0[1, 2, 3, 4]$, $K'''_0[2, 5]$, $K'_1[2, 5]$ and $K'''_1[1]$
12. $\Delta X''_3[3] = \mathtt{0}$ using $K'_0[0, 1, 3, 6]$, $K'_1[0, 7]$ and $K'''_1[3]$
13. $\Delta X''_3[0] = \mathtt{a}$ using $K'_0[0, 2, 5]$, $K'''_0[1]$ and $K'_1[1]$
14. $\Delta X''_3[2] = \mathtt{a}$ using $K'_0[0, 7]$ and $K'_1[3]$

15. $\Delta X'_{21}[0] = \mathtt{0}$ using $K'_0[6]$
16. $\Delta X'_{21}[7] = \mathtt{0}$ using $K'_0[0]$ and $K'''_0[1] \oplus K'''_0[6]$

Note that the average probability for satisfying each of above equations is 2^{-4}, except for number 2, 12, 15, 16 and one of 8 or 10 that in average are 2^{-2}. Hence, for each remaining differential pair, there are in average $2^{88} \cdot 2^{-(4 \cdot 11 + 2 \cdot 5)} = 2^{34}$ key candidates that satisfy all the equations. Moreover, this means that in average, each of these keys will be counted about $2^{51} \cdot 2^{34} \cdot 2^{-88} = 2^{-3}$ times, while for the right value of the key it is expected to be about 7 times. In other meaning, signal to noise ratio of the differential key recovery attack is about 56.

To not use a memory for 2^{88} key counters, instead, the attacker can guess value of these 22 key nibbles and then find the differential pairs which this key value is a candidate to satisfy those 16 equations. Note that if the attacker guesses all the 22 key nibbles at once, then the computation time for key recovery will be higher than exhaustive search for the weak key. To avoid this problem, he can guess the key nibbles one-by-one and at each level he checks for the corresponding equation. For instance, he can start with equations which only need one key nibble to be guessed. In this way, the computation complexity of this step will be reduced significantly.

Beyond these 22 key nibbles, the value of t_0, t_4, t_{14} which the right differential pairs happen as expected determines value of the corresponding three key nibbles. This leaves $28 - 22 - 3 = 3$ key nibbles, $k'_4 = K'_1[4], k'_6 = K'_1[6]$ and $k'_9 = K'''_1[5]$, that can be found by doing an exhaustive search.

Attack Complexity Since the signal to noise ratio of the attack is high, 7 correct differential pairs is enough to recover 100 bits of the weak key. Overall, we need to ask for $2^{12} \cdot 2^{62} = 2^{74}$ data encryptions which determines the data complexity of the attack. Besides, to keep the potentially right differential pairs for each value of t_0, t_4, t_{14}, we need about 2^{51} blocks of memory.

About the time complexity, the order of equations to be checked and the corresponding key nibbles to be guessed is important. If order of the equations to be checked is 7 ($K'_0[7]$), 15 ($K'_0[6]$), 3 ($K'_0[1]$), 5 ($K'''_0[4]$), 16 ($K'''_0[1] \oplus K'''_0[6]$), 9 ($K'_0[7]$ and $K'_1[3]$), 14 (nothing), 6 ($K'_0[3]$ and $K'''_0[6]$), 2 ($K'''_1[3]$), 1 ($K'_0[4]$ and $K'''_1[4]$), 4 ($K'''_0[2]$), 12 ($K'_1[0, 7]$), 13 ($K'_0[2, 5]$ and $K'_1[1]$), 10 ($K'''_0[5]$ and $K'_1[5]$) and 11 ($K'_1[2]$ and $K'''_1[1]$), then we need to do about 2^{34} partial encryption/decryption for each remaining differential pairs. Since the partial encryption is at most for three rounds, this means that the computation cost for key recovery step is about $2^{34} \cdot \frac{3}{23} \approx 2^{31}$ encryptions per pair and each tweak value. Therefore, the time complexity of the attack is about $2^{12} \cdot 2^{51} \cdot 2^{31} = 2^{94}$ encryption.

5.2 Related-Tweak Differential Attack on 26-Round CRAFT

The highest EDP for differentials within the trail RT_0 of Table 5, is $2^{-60.83}$ and it happens for 2688 differentials. Considering that the trail is staring at the beginning of fourth round, the format of these differences are shown below.

$$\Delta X_4 = (\mathtt{00}xy\ \ \mathtt{000}z\ \ \mathtt{a000}\ \ \mathtt{0000})\,, \quad \Delta X_{25} = (\mathtt{000}w\ \ \mathtt{00}x\mathtt{0}\ \ t\mathtt{0}u\mathtt{0}\ \ \mathtt{00}vx)\,,$$

with tweak difference of $(\mathtt{00}x\mathtt{0}\ \ \mathtt{0000}\ \ \mathtt{0000}\ \ \mathtt{0000})$. To achieve this 21-round related-tweak differentials, we need to have following weaknesses in the tweakey schedule of CRAFT:

$$\begin{aligned} TK_0'''[0] &= TK_0'''[3] = TK_0'''[6] = TK_0'''[7] = \mathtt{0}, & TK_2'''[0] &= TK_2'''[6] = TK_2'''[7] = \mathtt{0}, \\ TK_1'''[0] &= TK_1'''[3] = TK_1'''[6] = TK_1'''[7] = \mathtt{0}, & TK_3'''[0] &= TK_3'''[6] = TK_3'''[7] = \mathtt{0}. \end{aligned}$$

These lead to 2^{52} weak tweaks (with four conditions of $T[4] = T[8] = T[11]$ and $T[13] = T[15]$), and 2^{108} weak-keys (with five conditions of $K_0[8] = K_1[8] = K_0[11] = K_1[11]$, $K_0[15] = K_1[15]$ and $K_0[14] = K_1[14]$, with three extra conditions between the tweak and key nibbles. All together, the distinguisher works for a weak tweak-key set of size $2^{108+52-12} = 2^{148}$. Therefore, the weak tweaks and weak keys will be in the forms of

$$\begin{aligned} K_0 &= (k_0,\, k_1,\, k_2,\, k_3,\, k_4,\, k_5,\, k_6,\, k_7,\, k_8,\, k_9,\, k_{10},\, k_8,\, k_{12},\, k_{13},\, k_{14},\, k_{15})\,, \\ K_1 &= (k_0',\, k_1',\, k_2',\, k_3',\, k_4',\, k_5',\, k_6',\, k_7',\, k_8,\, k_9',\, k_{10}',\, k_8,\, k_{12}',\, k_{13}',\, k_{14},\, k_{15})\,, \\ T &= (t_0,\, t_1,\, t_2,\, t_3,\, t_4,\, t_5,\, t_6,\, t_7,\, t_4,\, t_9,\, t_{10},\, t_4,\, t_{12},\, t_{13},\, t_{14},\, t_{13})\,, \end{aligned}$$

together with three conditions between key and tweak: $k_8 = t_4,\ k_{15} = t_{13},\ k_{14} = t_{14}$.

Since, the values for key nibbles k_8, k_{14} and k_{15} are unknown to the attacker, to check their equality with t_4, t_{14} and t_{13}, respectively, the attacker needs to repeat the key recovery attack for all 2^{12} different values of t_4, t_{13} and t_{14}. Moreover, note that in the weak tweak-key set for this trail, we know the values for $TK_2'''[3]$ and $TK_3'''[3]$.

To reduce the effect of differential expansion in extending the trails and to have more differentials for a fixed related-tweak, we only use the ones with $x = z = \mathtt{a}$ and $y = \mathtt{0}$. This way, there are 256 differentials left that $u = v \oplus \mathtt{a}$, $t \in \{\mathtt{5}, \mathtt{a}, \mathtt{d}, \mathtt{f}\}$ and there are 64 choices for (v, w) pair (that we denote the set of these 64 pairs by $\Delta_{v,w}$). Extending these differentials by four rounds in the beginning activates all the nibbles in plaintext difference, with three linear conditions in the difference of plaintext nibbles:

$$\Delta P[5] = \Delta P[13]\,, \quad \Delta P[7] \oplus \Delta P[15] \ \text{ and } \ \Delta P[3] \oplus \Delta P[7] \oplus \Delta P[11] \in \{\mathtt{5}, \mathtt{a}, \mathtt{d}, \mathtt{f}\}\,.$$

On the other side, extending the differentials by one round, activates eleven nibbles in the ciphertext difference, with two constant, four linear and three nonlinear conditions in the difference of ciphertext nibbles:

$$\begin{aligned} &\Delta C[0] = \Delta C[8]\,,\ \ \Delta C[7] = \Delta C[15]\,,\ \ \Delta C[1] = \Delta C[5] = \Delta C[13]\,,\ \ \Delta S(C[11]) = \mathtt{a}\,, \\ &\Delta S(C[15]) \in \{\mathtt{5}, \mathtt{a}, \mathtt{d}, \mathtt{f}\}\,,\ \ \big(\Delta S(C[8])\,, \Delta C[3] \oplus \Delta C[11] \oplus \Delta C[15]\big) \in \Delta_{v,w} \end{aligned}$$

The details of extensions in both direction are depicted in Figure 5.

This attack is similar to the previously mentioned single-tweak differential attack on 23 rounds. To recover the whole key, we need to repeat the attack for each 2^{12} values of t_4, t_{13} and t_{14}. Therefore, we first fix ten tweak nibbles, $t_0, t_1, t_2, t_3, t_5, t_6, t_7, t_9, t_{10}, t_{12}$, to zero and then choose value for t_0, t_4, t_{14}. Note that in the related tweak, we need to fix t_2 to $\mathtt{a}$.

Collecting Data Structures For each values of t_4, t_{13}, t_{14}, we choose 2^{60} random plaintexts $(p_0, \ldots, p_{15})$ and ask for their encryption $(c_0, \ldots, c_{15})$ using the weak tweak $(\mathtt{0}, \mathtt{0}, \mathtt{0}, \mathtt{0}, t_4, \mathtt{0}, \mathtt{0}, \mathtt{0}, t_4, \mathtt{0}, \mathtt{0}, t_4, \mathtt{0}, t_{13}, t_{14}, t_{13})$ and ask also for encryption of another 2^{60} random plaintexts using the related tweak $(\mathtt{0}, \mathtt{0}, \mathtt{a}, \mathtt{0}, t_4, \mathtt{0}, \mathtt{0}, \mathtt{0}, t_4, \mathtt{0},\ \mathtt{0}, t_4, \mathtt{0}, t_{13}, t_{14}, t_{13})$. For both of the tweak

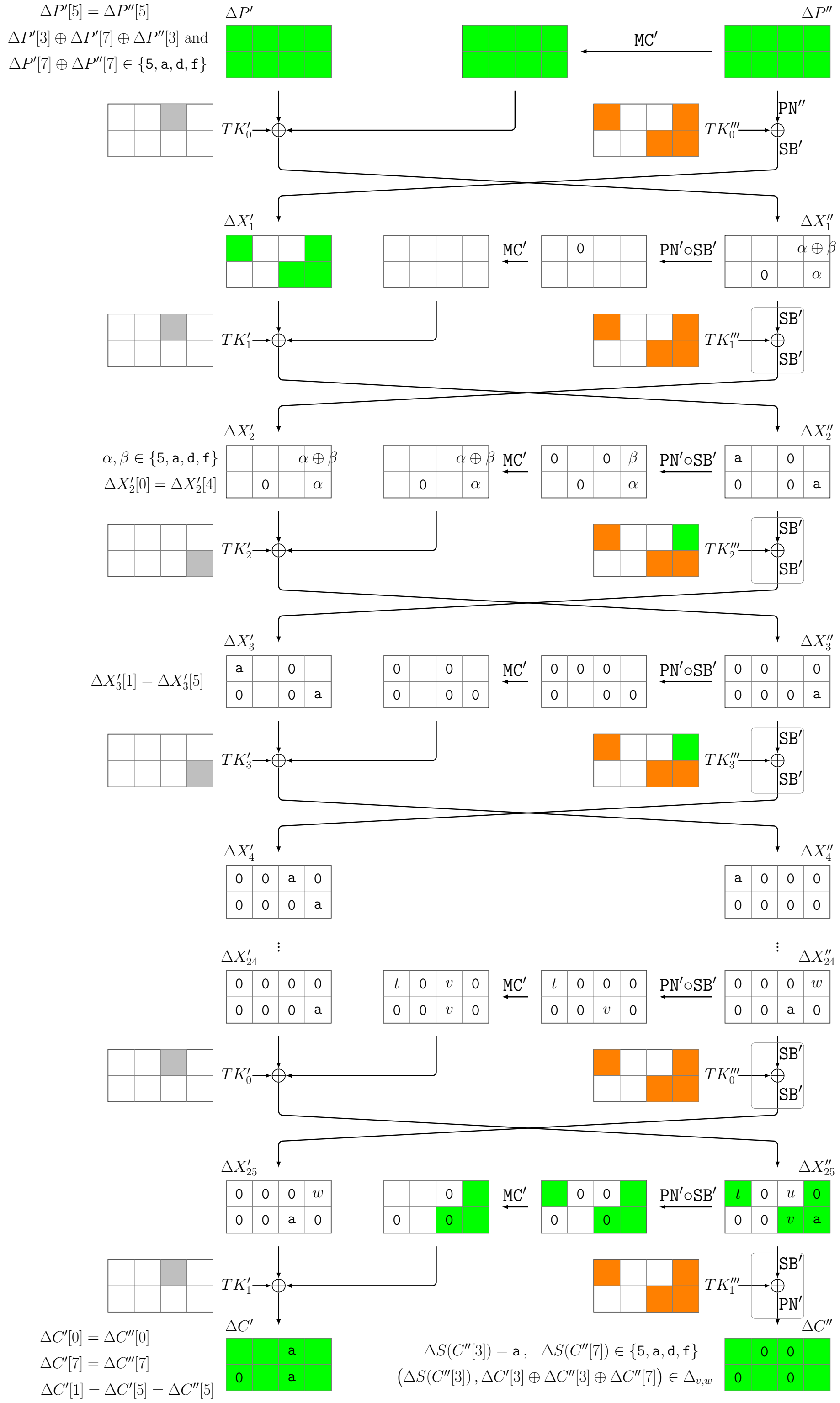

$\Delta P'[5] = \Delta P''[5]$
$\Delta P'[3] \oplus \Delta P'[7] \oplus \Delta P''[3]$ and
$\Delta P'[7] \oplus \Delta P''[7] \in \{5, a, d, f\}$
$\Delta P'$
$\Delta P''$
MC′
TK'_0
TK'''_0
PN″
SB′
$\Delta X'_1$
$\Delta X''_1$
MC′
PN′∘SB′
$\alpha \oplus \beta$
α
TK'_1
TK'''_1
$\Delta X'_2$
$\Delta X''_2$
$\alpha, \beta \in \{5, a, d, f\}$
$\Delta X'_2[0] = \Delta X'_2[4]$
β
TK'_2
TK'''_2
$\Delta X'_3$
$\Delta X''_3$
$\Delta X'_3[1] = \Delta X'_3[5]$
TK'_3
TK'''_3
$\Delta X'_4$
$\Delta X''_4$
$\Delta X'_{24}$
$\Delta X''_{24}$
t
v
w
TK'_0
TK'''_0
$\Delta X'_{25}$
$\Delta X''_{25}$
u
TK'_1
TK'''_1
PN′
$\Delta C'$
$\Delta C''$
$\Delta C'[0] = \Delta C''[0]$
$\Delta C'[7] = \Delta C''[7]$
$\Delta C'[1] = \Delta C'[5] = \Delta C''[5]$
$\Delta S(C''[3]) = a, \quad \Delta S(C''[7]) \in \{5, a, d, f\}$
$\left(\Delta S(C''[3]), \Delta C'[3] \oplus \Delta C''[3] \oplus \Delta C''[7]\right) \in \Delta_{v,w}$

Figure 5: Extending the 21-round activity pattern RT_0 from Table 5 by four rounds before and one round after the trail.

values, considering value of $a_0 = p_5 \oplus p_{13}$, $a_1 = p_7 \oplus p_{15}$ and $a_2 = p_3 \oplus p_7 \oplus p_{11}$ for each plaintext, we separate the collected data to 2^{12} sets which each set contains only the data with the same value for (a_0, a_1, a_2). In average, there are 2^{48} plaintext/ciphertext pairs. Using any plaintext/ciphertext pair of set (a_0, a_1, a_2) from the weak tweak together with any plaintext/ciphertext pair of set (a_0, a'_1, a'_2) from the related tweak with $a'_1 \in a_1 \oplus \{\mathtt{5}, \mathtt{a}, \mathtt{d}, \mathtt{f}\}$ and $a'_2 \in a_2 \oplus \{\mathtt{5}, \mathtt{a}, \mathtt{d}, \mathtt{f}\}$, provides a differential pair with $\Delta p_5 = \Delta p_{13}$, $\Delta p_7 \oplus \Delta p_{15}$ and $\Delta p_3 \oplus \Delta p_7 \oplus \Delta p_{11} \in \{\mathtt{5}, \mathtt{a}, \mathtt{d}, \mathtt{f}\}$. Hence, there are about $2^{12} \cdot 4^2 \cdot 2^{48 \cdot 2} = 2^{112}$ differential pairs with an appropriate plaintext difference.

The probability that one of these differential pairs lead to the difference in ΔX_4 is 2^{-56}. In other meaning, in average, there are about 2^{56} differential pairs with the correct difference in ΔX_4 and about $256 \cdot 2^{56-60.83} \approx 9$ differential pairs with the correct difference in ΔX_{25}. Therefore, for each t_4, t_{13}, t_{14} value, the attacker needs to use $2 \cdot 2^{60}$ plaintext/ciphertext pairs (2^{112} differential pairs) to insure that for the case that $k_8 = t_4$, $k_{15} = t_{13}$, $k_{14} = t_{14}$, there are about 9 of right differential pairs with the corresponding ΔX_{25}.

Filtering Wrong Differential Pairs On the other side of the differential pairs, the attacker can use the ciphertext differentials to sieve the potentially right differential pairs.

$$\begin{gathered}
\Delta c_4 = \Delta c_9 = \Delta c_{10} = \Delta c_{12} = \Delta c_{14} = \mathtt{0}\,, \quad \Delta c_2 = \Delta c_6 = \mathtt{a}\,, \\
\Delta c_0 = \Delta c_8\,, \quad \Delta c_7 = \Delta c_{15}\,, \quad \Delta c_1 = \Delta c_5 = \Delta c_{13}\,, \quad \Delta S(c_{11}) = \mathtt{a}\,, \\
\Delta S(c_{15}) \in \{\mathtt{5}, \mathtt{a}, \mathtt{d}, \mathtt{f}\}\,, \quad \big(\Delta S(c_8)\,, \Delta c_3 \oplus \Delta c_{11} \oplus \Delta c_{15}\big) \in \Delta_{v,w}
\end{gathered}$$

All together, for each t_4, t_{13}, t_{14} value, after this filtering, from all 2^{112} differential pairs, there will be only $2^{112-52} = 2^{60}$ of them left. We keep all these remaining differential pairs in a memory to use them in the key recovery step.

Recovering Key Nibbles Apart from k_8, k_{15}, k_{14}, the attacker can recover other 23 nibbles of the key, namely K'_0, K'_1, $K'''_0[1,2,4,5]$ and $K'''_1[1,2,4]$, by using the following 19 equations which are based on the partial encryption or decryption of the differential pairs. Note that only one nibble of the weak key remains unknown, $K'''_1[5]$.

1. $\Delta X'_2[0] = \Delta X'_2[4]$ using $K'_0[4]$ and $K'''_1[4]$

2. $\Delta X''_2[0] = \mathtt{a}$ using $K'_0[1,6]$
3. $\Delta X''_2[2] = \mathtt{0}$ using $K'_0[3,4]$ and $K'''_0[2]$
4. $\Delta X''_2[4] = \mathtt{0}$ using $K'_0[1]$ and $K'''_0[4]$
5. $\Delta X''_2[6] = \mathtt{0}$ using $K'_0[3]$
6. $\Delta X''_2[7] = \mathtt{a}$ using $K'_0[0]$

7. $\Delta X'_3[1] = \Delta X'_3[5]$ using $K'_0[2,5]$, $K'''_0[1,5]$ and $K'_1[1,5]$

8. $\Delta X''_3[0] = \mathtt{0}$ using $K'_0[2,3,5]$, $K'''_0[1]$ and $K'_1[1,6]$
9. $\Delta X''_3[1] = \mathtt{0}$ using $K'_0[1,2,3,4]$, $K'''_0[2,5]$, $K'_1[2,5]$ and $K'''_1[1]$
10. $\Delta X''_3[6] = \mathtt{0}$ using $K'_0[0,7]$ and $K'_1[3]$
11. $\Delta X''_3[7] = \mathtt{a}$ using $K'_0[1,6]$ and $K'_1[0]$
12. $\Delta X''_3[3] \oplus \Delta X''_3[7] = \mathtt{a}$ using $K'_0[0]$ and $K'_1[7]$

13. $\Delta X'_4[2] = \mathtt{a}$ using $K'_0[0,1,2,7]$, $K'''_0[4]$, $K'_1[3,4]$ and $K'''_1[2]$

14. $\Delta X''_4[3] = \mathtt{0}$ using $K'_0[0,1,2,3,5,6,7]$, $K'''_0[1]$ and $K'_1[0,1,3,6]$
15. $\Delta X''_4[5] = \mathtt{0}$ using $K'_0[0,1,2,7]$, $K'''_0[4,5]$, $K'_1[3,4,5]$ and $K'''_1[2]$

16. $\Delta X''_{25}[2] = v \oplus \mathtt{a}$ using $K'''_1[2]$

17. $\Delta X'_{24}[0] = \mathtt{0}$ using $K'_1[6]$
18. $\Delta X'_{24}[2] = \mathtt{0}$ using $K'_1[3]$, $K'''_1[2]$ and $K'_1[4] \oplus K'''_1[1]$
19. $\Delta X'_{24}[6] = \mathtt{0}$ using $K'_1[3]$

Note that the average probability for satisfying each of above equations is 2^{-4}, except for number 11, 12, 17 and one of 18 or 19 that in average are 2^{-2}. Hence, for each remaining differential pair, there are in average $2^{23\cdot 4} \cdot 2^{-(4\cdot 15+2\cdot 4)} = 2^{24}$ key candidates that satisfy all the equations. Moreover, this means that in average, each of these keys will be counted about $2^{60} \cdot 2^{24} \cdot 2^{-92} = 2^{-8}$ times, while for the right value of the key it is expected to be about 9 times. In other meaning, signal to noise ratio of the differential key recovery attack is about 2300. It is important to mention that to increase the signal to noise ratio, in the ciphertext side, we use two rounds for the key recovery attack. This is possible because the differentials (and their corresponding EDPs) that we are using, all are based on the same activity pattern which requires conditions to be satisfied.

Moreover, the value of t_4, t_{13}, t_{14} which the right differential pairs happen as expected determines value of corresponding three key nibbles and the remaining key nibble, $K_1'''[5]$ can be found by doing an exhaustive search.

Attack Complexity Since the signal to noise ratio of the attack is very high, 9 correct differential pairs is enough to recover 108 bits of the weak key. Overall, we need to ask for $2^{12} \cdot 2 \cdot 2^{60} = 2^{73}$ data encryptions which determines the data complexity of the attack. Besides, to keep the potentially right differential pairs for each value of t_4, t_{13}, t_{14}, we need about 2^{60} blocks of memory.

About the time complexity, if order of the equations to be checked is 5 ($K_0'[3]$), 6 ($K_0'[0]$), 16 ($K_1'''[2]$), 12 ($K_1'[7]$), 17 ($K_1'[6]$), 19 ($K_1'[3]$), 10 ($K_0'[7]$), 18 ($K_1'[4] \oplus K_1'''[1]$), 1 ($K_0'[4]$ and $K_1'''[4]$), 3 ($K_0'''[2]$), 2 ($K_0'[1,6]$), 4 ($K_0'''[4]$), 11 ($K_1'[0]$), 13 ($K_0'[2]$ and $K_1'[4]$), 15 ($K_0'''[5]$ and $K_1'[5]$), 9 ($K_1'[2]$), 7 ($K_0'[5]$, $K_0'''[1]$ and $K_1'[1]$), 8 (-) and 14 (-), then we need to do about 2^{36} partial encryption/decryption for each remaining differential pairs. Since the partial encryption is at most for three rounds, this means that the computation cost for key recovery step is about $2^{36} \cdot \frac{3}{26} \approx 2^{33}$ encryptions per pair and each tweak value. Therefore, the time complexity of the attack is about $2^{12} \cdot 2^{60} \cdot 2^{33} = 2^{105}$ encryptions.

6 Conclusion

Most of the security analysis for new block cipher designs are based on the assumption of independent round keys. While this is not an issue in most cases, it may result in overestimating the resistant of the design, in particular for in the weak-key scenario.

In this work, we showed how the SPN structure of CRAFT block cipher (with full-state non-linear layer) changes to a Feistel-network structure (with half-state non-linear layer) in the weak tweak-key scenario. Consequently, in the same number of rounds, the weak tweak-key structure of the cipher is less resistant against differential and linear cryptanalyses. As an application of this observation, we present one weak-key single-tweak differential attack on 23 rounds of the cipher with time complexity of 2^{94} encryptions and data complexity of 2^{74} chosen plaintext/tweak/ciphertext tuples that works for 2^{112} weak keys. We also present one weak-key related-tweak attack on 26 rounds of the cipher time complexity of 2^{105} encryptions and data complexity 2^{73} chosen plaintext/tweak/ciphertext tuples that works for 2^{108} weak keys. It is important to mention that these attacks do not overcome the security claim of the CRAFT block cipher.

Acknowledgments

We want to thank Christof Beierle for his constructive comments while preparing this work. Besides, the work described in this paper has been partially supported by the German Research Foundation (DFG) within the project LE 3372/5-1 and also by the Netherlands Organisation for Scientific Research (NWO) under TOP grant TOP1.18.002 SCALAR.

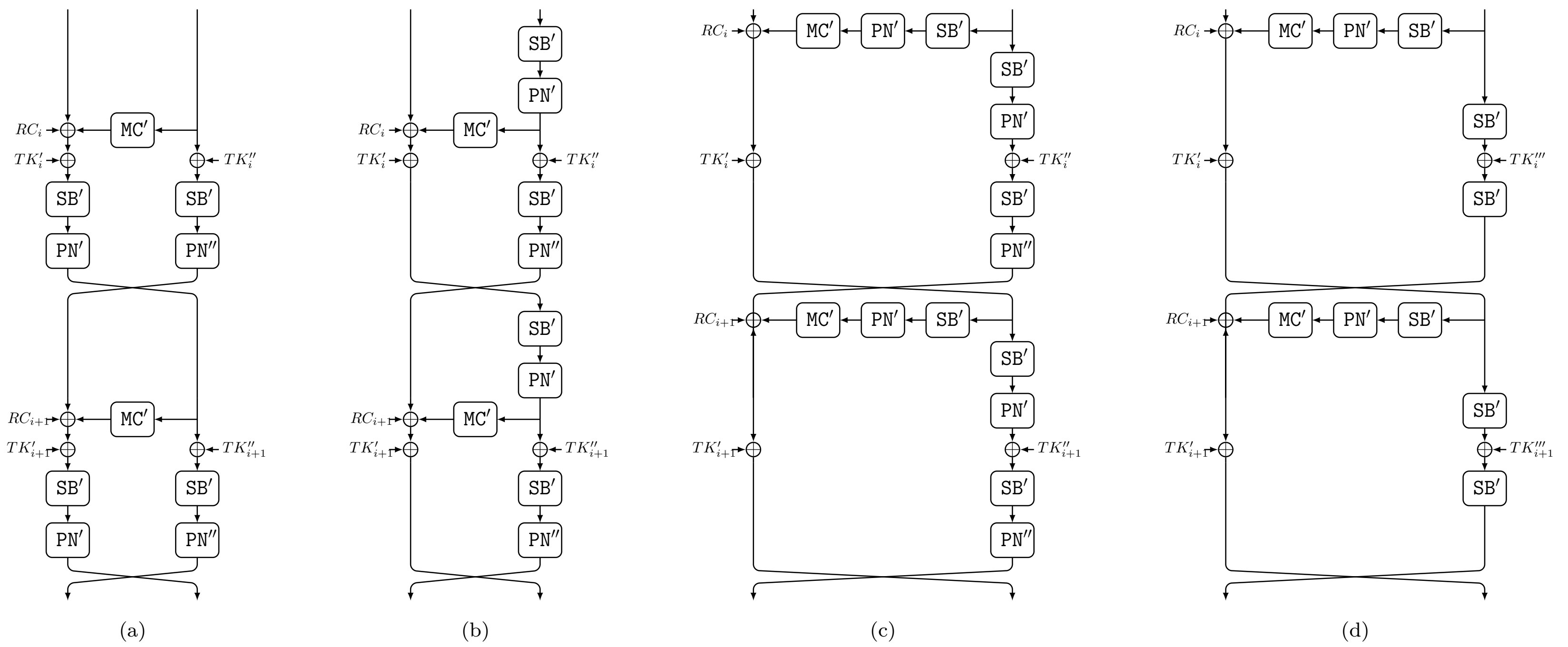


Figure 6: Transforming two representation rounds of CRAFT to two equivalent rounds: (a) two consecutive rounds, (b) bringing SB′ and PN′ from end of left branch of each round to the beginning of the right branch in the next round, (c) passing both SB′ and PN′ through the bridge point of each round, (d) removing both PN′ and PN″ in the right branch of each round by replacing TK''_i with $TK'''_i = \mathtt{PN}''(TK''_i)$.

References

[BBI$^+$15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 411–436. Springer, Heidelberg, November / December 2015.

[BLMR19] Christof Beierle, Gregor Leander, Amir Moradi, and Shahram Rasoolzadeh. CRAFT: Lightweight tweakable block cipher with efficient protection against DFA attacks. *IACR Trans. Symm. Cryptol.*, 2019(1):5–45, 2019.

[BS97] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 513–525. Springer, Heidelberg, August 1997.

[ELR20] Maria Eichlseder, Gregor Leander, and Shahram Rasoolzadeh. Computing expected differential probability of (truncated) differentials and expected linear potential of (multidimensional) linear hulls in SPN block ciphers. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *INDOCRYPT 2020*, volume 12578 of *LNCS*, pages 345–369. Springer, Heidelberg, December 2020.

[EY19] Muhammad ElSheikh and Amr M. Youssef. Related-key differential cryptanalysis of full round CRAFT. Cryptology ePrint Archive, Report 2019/932, 2019. https://eprint.iacr.org/2019/932.

[GSS$^+$20] Hao Guo, Siwei Sun, Danping Shi, Ling Sun, Yao Sun, Lei Hu, and Meiqin Wang. Differential attacks on CRAFT exploiting the involutory s-boxes and tweak additions. *IACR Trans. Symmetric Cryptol.*, 2020(3):119–151, 2020.

[HSN$^+$19] Hosein Hadipour, Sadegh Sadeghi, Majid M. Niknam, Ling Song, and Nasour Bagheri. Comprehensive security analysis of CRAFT. *IACR Trans. Symm. Cryptol.*, 2019(4):290–317, 2019.

[MA19] AmirHossein E. Moghaddam and Zahra Ahmadian. New automatic search method for truncated-differential characteristics: Application to midori, SKINNY and CRAFT. Cryptology ePrint Archive, Report 2019/126, 2019. https://eprint.iacr.org/2019/126.

[MWGP11] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Inscrypt 2011*, volume 7537 of *LNCS*, pages 57–76. Springer, December 2011.

[SHW$^+$14] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 158–178. Springer, Heidelberg, December 2014.