Two Sides of The Same Coin: Weak-Keys and More Efficient Variants of CRAFT

Gregor Leander[1] and Shahram Rasoolzadeh[1,2†]

[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, i.e., with half state non-linear operation and half state key addition. This has two interesting consequences: First, the new structure of the cipher is less resistant against differential and linear cryptanalyses. Second, it allows a more efficient implementation of the 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 of the any reported round-based implementation of a lightweight block cipher. 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 their analysis 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.

Known Results on CRAFT

Hadipour et al. [HSN+19] presented a detailed security analysis on CRAFT. In particular, they presented 14-round related-tweak zero-correlation linear hull distinguishers. Using

†Part of this work was done while the second author was at Ruhr University Bochum.

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 CryptoSMT [Köl14] and 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 a 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 secuirty for CRAFT in the related-key model.

More recently, and most relevant for our work, Guo et al. [GSS+20] studied the combination of the involuntary 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 use 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 the later one 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%.

1.1 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-key scenario, mainly thanks to the involuntary S-box and the special choice of MixColumns, used in CRAFT, the equivalent representation of the cipher makes a Feistel network that the non-linear operation (S-box layer) only applied on half of the state.

We analyze the security of the new weak-key structure of the cipher and show that comparing 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 explain the weak-keys identified there.

In [GSS+20], the authors focused on the differential weakness caused by those weak keys while. Our representation reveal a general weak-key structure that could be exploited by an attacker also with other techniques.

Finally, we focus on an interesting constructive side of the new structure. We show that when decryption is added on top of the encryption, the area overhead of the implementation is almost zero. Besides, to protect against fault analysis attacks, the area overhead for implementing the countermeasure is about half of the case for CRAFT original structure. Therefore, by probably increasing number of rounds and applying an appropriate tweakey schedule, it is possible to use the new structure as a new block cipher with more efficiency in hardware implementation.

1.2 Outline

First in Section 2, we specify 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 a MILP tool to find all

the activity patterns with the minimum number of active S-boxes in differential and linear trails of the weak-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-key space for the weak-key structure. Later in Section 5, we show that the new structure allows an even better implementation of the cipher specially when the encryption is combined with decryption and also when it is combined with the countermeasures against fault analysis. 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 31 identical round functions $\mathcal{R}_i$ with $0 \leq i \leq 30$ together with 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 MIDOR [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: Each column of the state is multiplied with the following involutory binary matrix :

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

- 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 nibbles, respectively.

- $\mathtt{A}_{TK_i}$: The cipher uses 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

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

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 that for the simplicity, we will use TK_i notion.

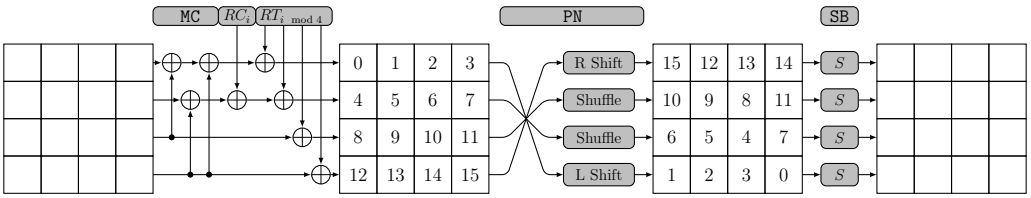

Figure 1: One full round of CRAFT.

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

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

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

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

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

3 CRAFT Weak-Key Structure

In the following, based on the given properties, first, we present an equivalent definition for CRAFT round function. Then, using the new representation, we introduce a weak-key structure for the cipher.

We use X' and X'' to denote the left half and right half 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.,

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

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

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

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

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

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 $\mathtt{PN}'' = \mathtt{PN}'^{-1}$.

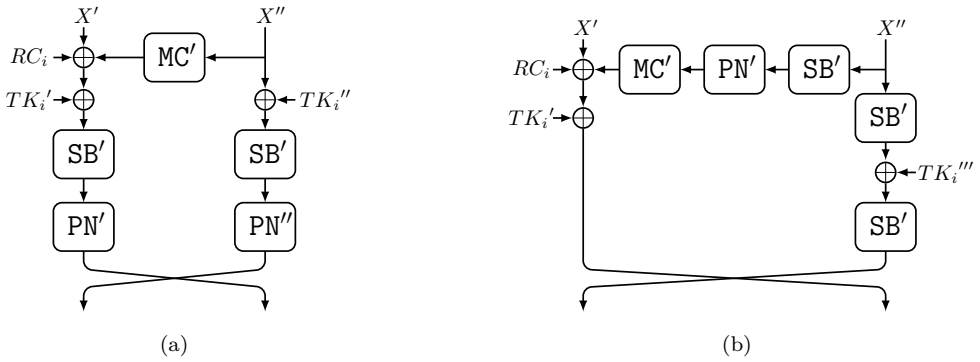

Figure 2: Representation and equivalent round functions of CRAFT.

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*

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

that $TK_i''' = \mathtt{PN}''(TK_i'')$, except in the last round, similar to the Feistel network, the final swapping between the right and left halves are 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 CRAFT round function in Figure 2(a). Similarly for the last linear round, we have:

$$\mathcal{R}_{31}(X' \,\|\, X'') = \Big(\mathtt{A}'_{TK_3'}\big(X' \oplus \mathtt{MC}'(X'') \oplus RC'_{31}\big) \,\|\, \mathtt{A}'_{TK_3''}(X'')\Big).$$

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 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 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

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

it is possible to simplify the equivalent round functions:

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

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

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

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. □

The equivalent representation of 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 $\mathtt{SB}' \circ \mathtt{A}_{TK'''_i} \circ \mathtt{SB}'$ function while in the case of a Feistel network, this functions is the identity function.

On the other hand, this suggests a weak-key structure for CRAFT. If $\mathtt{SB}' \circ \mathtt{A}_{TK'''_i} \circ \mathtt{SB}'$ is equal to the identity function, CRAFT will be a Feistel network which includes partial nonlinear round and partial key-addition.

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 $X' \in \mathbb{F}_2^{32}$,

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

□

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

Proposition 2. *If all the right halves of the tweakeys in CRAFT encryption are equal to zero, i.e. all $TK''_i = 0$ with $0 \le i < 4$, then the encryption is equivalent to the Feistel network with the following round function*

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

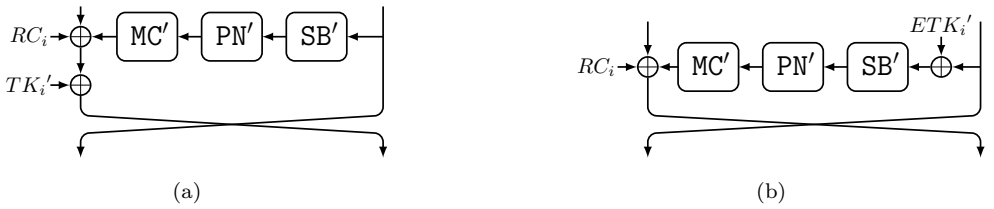

Figure 3: Feistel round function of weak-key CRAFT.

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

$$ETK_0 = 0 \quad , \quad ETK_1 = K_0' \oplus T' \ , \quad ETK_2 = K_1' \oplus T' \ , \quad ETK_3 = T' \oplus T''',$$
$$ETK_4 = T' \oplus T''', \quad ETK_5 = K_0' \oplus T''', \quad ETK_6 = K_1' \oplus T''', \quad ETK_7 = 0$$

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., $T''' = \big(\mathtt{QN}(T)[0], \ldots, \mathtt{QN}(T)[7]\big)$.

Proof. The behavior of 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 $i > 1$,

$$ERK_i = RK_{i-1} \oplus ERK_{i-2} \text{ with } ERK_0 = 0, ERK_1 = RK_0\,,$$

together with a whitening key in the ciphertext. And the whitening key is ERK_{r-1} on the right half and ERK_i on the left half.

The CRAFT tweakey schedule uses four 32-bit round tweakey, TK_0', TK_1', TK_2' and TK_3' repeatedly. For the equivalent round tweakeys, we would have the following eight round tweakeys:

$$\begin{aligned}
ETK_0 &= && = && = 0 \ ,\\
ETK_1 &= && = TK_0' && = K_0' \oplus T' \ ,\\
ETK_2 &= ETK_0 \oplus TK_1' = TK_1' && = K_1' \oplus T' \ ,\\
ETK_3 &= ETK_1 \oplus TK_2' = TK_2' \oplus TK_0' && = T' \oplus T''' \ ,\\
ETK_4 &= ETK_2 \oplus TK_3' = TK_3' \oplus TK_1' && = T' \oplus T''' \ ,\\
ETK_5 &= ETK_3 \oplus TK_0' = TK_2' && = K_0' \oplus T''' \ ,\\
ETK_6 &= ETK_4 \oplus TK_1' = TK_3' && = K_1' \oplus T''' \ ,\\
ETK_7 &= ETK_5 \oplus TK_2' = && = 0 \ ,
\end{aligned}$$

that are uses 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-key structure of CRAFT to be key-less rounds. This means, the security of the 32-round CRAFT cipher with Feistel network structure is based on the middle 30 rounds and the other two rounds are actually useless.

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-key CRAFT structure. We apply the methods introduced in [ELR20] to estimate the EDP for the differential effect within these differential activity patterns. Later, we show that the actual weak-key space in the differential activity patterns can be larger than the weak-key space for the weak-key structure (with size of 2^{64} weak keys).

Table 1: The minimum number of active S-boxes in differential and linear activity patterns in up to 31 rounds of CRAFT. n_1 and n_2 denote the numbers for the original and the weak-key structures, resp.

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

4.1 Minimum Number of Active S-boxes

To compare the resistance of CRAFT cipher in its original structure and in the weak-key structure against the differential and linear attacks, we compute the minimum number of active S-boxes in the single-tweak model.[1] 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 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 structure of the cipher, in the MILP codes, in each round, we need to consider all the 16 corresponding variables to the S-box layer as objective variables, in the case for the weak-key structure, we must consider the 8 corresponding variables to the right half variables in the S-box layer. Beyond this difference in the two structures of CRAFT, by taking benefit of the Feistel-like design of the cipher, 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, by 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 1 up to 31 rounds in both differential and linear activity patterns. We use n_1 and n_2 to denote the numbers for the original and the weak-key structures, respectively. One interesting observation from Table 1 is that for most of the number of rounds, n_2 is exactly half of n_1. Intuitively saying, this makes sense because in the weak-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 structure, after 9 rounds, all the numbers of active S-boxes are higher than 32, for the weak-key structure, 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

[1] We recall that in a differential (resp. linear) activity pattern, an S-box is called *active*, if the input difference (resp. mask) is non-zero.

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
p	25.79	29.79	35.54	41.42	41.19	45.19	50.42	56.42	54.00	58.11

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 the single characteristics which follow the same activity pattern with the minimum number of active S-boxes. Then by summing all theses probabilities of each single characteristics, we find a lower bound for the probability of corresponding differential. We repeat this computation for all the different values for the input and output differences in the differential to find the differential with the maximum EDP.

It is noteworthy to mention again 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 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 18 rounds. For 19 rounds and more, there is no differentials within the activity patterns of minimum number of active S-boxes that has EDP of higher than 2^{-64}. Note that in this table, instead of showing value of the 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-Key Space 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 the i values which leads to 2^{64} weak-keys out of the 2^{128} keys. But for the differentials within an activity pattern, considering $TK_i''' = 0$ is a generous condition. Considering the 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''' 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 verse.

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

Example 1. Consider a distinguisher corresponding to the trail 1 from Table 3. This

distinguisher works for the weak-key space which leads to

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

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

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

Satisfying these conditions is possible by fixing 4 nibbles (indexed by 8, 11, 13, 15) from K_0 and 3 nibbles (indexed by 11, 14, 15) from K_1 to a value dependent on the tweak value. Hence, it works for the weak-key space of 2^{100} keys.

It is noteworthy to mention that the size of the weak-key space for other distinguishers from Table 3 is also 2^{100} keys. Besides, it is important to mention that there are some other distinguishers with a larger size for the weak-key space, but with a lower value for the EDP. By appending some rounds before and some after, one can use these distinguishers to

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.

	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 $(xyz) \in \{(\texttt{5,a,0}), (\texttt{7,5,f}), (\texttt{7,d,7}), (\texttt{a,5,f}), (\texttt{a,a,0}), (\texttt{a,d,7}), (\texttt{a,f,5}), (\texttt{d,a,0}), (\texttt{f,a,0}), (\texttt{f,f,5})\}$		$t = z \oplus \texttt{a}$, $x \in \{\texttt{5,a,d,f}\}$ and $(y, z) \in \{(\texttt{5,f}), (\texttt{a,0}), (\texttt{d,7}), (\texttt{f,5})\}$	

do a key recovery attack on the reduced-round CRAFT. For all the distinguishers in Table 3, appending 3 (resp. 4) rounds before (resp. after) the distinguisher activates all the nibbles in the plaintext (resp. ciphertext) difference. All together, it might be possible for the attacker to do a key recovery attack on 25 rounds of the CRAFT cipher.

Differential Properties of $S_c^* := S(S(\cdot) \oplus c)$: 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 4 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 $c = \alpha =$ 2 and $c = \alpha =$ 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 key value is not in the weak-key space), it may lead to a high EDP. But this EDP is always smaller than the one we computed for the weak-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 $c =$ a, while only the input difference value of a can transit with probability 1, input difference with a value in {s, 7, d, 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 keys within the weak-key spaces, but it might be possible to be applied for the keys out of the weak-key spaces by using a smaller EDP value.

It is noteworthy to mention that the transition probability for $c = \alpha =$ a, previously was observed and applied in [GSS+20] to enlarge the weak-key space. While their technique makes it possible to enlarge the weak-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-key space larger. This can be used as a trade-off between the EDP and the size of weak-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

Table 4: 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
	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
c	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

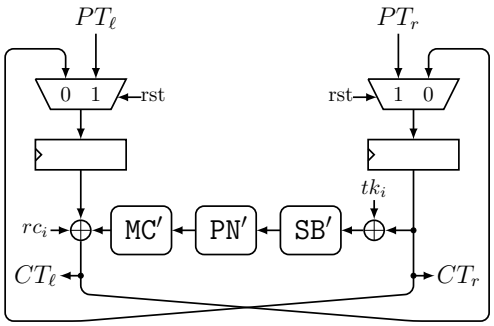

Figure 4: Round-based design architecture of the new structure.

this to reduce the data and/or time complexity of the attack.

Note that even though, we only analyzed security of the weak-key structure against differential and linear cryptanalyses, but since 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), in contrary to [GSS+20], we keep the general weakness as it is and do not specialize the weakness only for differential attack.

5 Hardware Implementation of the New Structure

Even though the Feistel network (weak-key structure) of the CRAFT block cipher provides less cryptographic security than the original design, in this section, we show that the new structure makes it possible to have a better implementation of the cipher specially when the encryption is combined with decryption and also when it is combined with the countermeasures against fault analysis. To this point, we start with recalling the specification of the new structure.

The new structure is a Feistel-network block cipher with the following round function:

$$\mathcal{R}_i(x_{i-1}, x_i) = (x_i, x_{i+1}) \text{ with } x_{i+1} = x_{i-1} \oplus \mathcal{F}_i(x_i) \text{ and } \mathcal{F}_i := \mathtt{A}_{RC'_i} \circ \mathtt{MC}' \circ \mathtt{PN}' \circ \mathtt{SB}' \circ \mathtt{A}_{tk_i} .$$

Note that since as it is shown in Proposition 2, the current tweakey schedule of the structure provides weak round tweakeys (the ones denoted by ETK_is), we recommend to use another stronger tweakey schedule that we denote them by tk_is.

Encryption Only: Since with an equal number of rounds, the new structure is less cryptographically secure than the original CRAFT structure, it is necessary to use more number of rounds to provide a secure cipher using the new structure. In the round-based design architecture (which was the target implementation of the CRAFT cipher), the overhead of increasing number of rounds is on the increasing number of clock cycles used for an encryption and does not affect the area cost of implementation.

Figure 4 illustrates the round-based implementation of the new structure excluding the circuit for the tweakey structure. Compared to the round-based implementation of CRAFT shown in [BLMR19, Figure 8], the new design decreases the area of implementation and it is because of using 8 S-boxes per round (instead of 16) and using 32 XOR gates for the key addition (instead of 64).

Encryption & Decryption Thanks to the being a Feistel network, in the new structure, the decryption is the same as the encryption with the reverse order of the round constants and round tweakeys. Therefore, to provide the decryption on top of the encryption, we

only need to add extra circuits for reversing the order of the round constants and the round tweakeys. For reversing order of the round constants, we only need to implement the inverse of the updating function for the LFSRs. Besides, if the tweakey schedule iterates the round-tweakeys (similar to the one in the CRAFT cipher), it is possible to reverse the order of the round tweakeys by updating the circuits for the selector bits of the multiplexer choosing the round tweakey. Note that both of these techniques are already already applied in the CRAFT design with a very small area overhead.

The prominent difference of combining encryption and decryption for the new structure and the CRAFT cipher is that in the case of the latter one, we need to modify the round tweakeys from TK_i to $\texttt{MC}(TK_i)$. This modification needs extra 48 XOR (to implement MC circuit) and 32 MUX gates (to choose between these two round tweakeys). All together, the encryption+decryption circuit of the new structure is smaller than the one for CRAFT cipher with difference of 8 Sboxes, 80 XOR and 32 MUX gates which in the IBM 130nm ASIC library equals to 328 GE.

It is noteworthy to mention that here, we do not consider the difference in the implementation of the new tweakey schedule. Just as an example, consider the 128-bit master key as four 32-bit keys $(k_0\|k_1\|k_2\|k_3)$, and the 64-bit tweak as two 32-bit tweakeys $(t_0\|t_1)$. The similar tweakey schedule to the CRAFT tweakey schedule would be iterating following 12 round tweakeys:

$$\begin{array}{lllllll} tk_0 = k_0 \oplus t_0, & tk_1 = k_1, & tk_2 = k_2 \oplus t_1, & tk_3 = k_3 \oplus t_0, & tk_4 = k_0, & tk_5 = k_1 \oplus t_1, \\ tk_6 = k_2 \oplus t_0, & tk_7 = k_3, & tk_8 = k_0 \oplus t_1, & tk_9 = k_1 \oplus t_0, & tk_{10} = k_2, & tk_{11} = k_3 \oplus t_1. \end{array}$$

Please note that the new tweakey schedule not only has smaller footprint in the implementation, it also makes the Time-Data-Memory Trade-off attack mentioned in [BLMR19, Section 5.3] impossible.

Applying Fault Analysis Countermeasures In [AMR+20, SRM20], it is shown that applying concurrent error detection-based countermeasures against fault analysis have overhead of more than 100% (even if there is one bit redundancy per nibble) and this gets worse in the case of error correction-based countermeasures. The main reason for this is that the *check points* or the *correction points* in these countermeasures are several times larger than the circuit for the round operation.

While in the fault analysis countermeasures for the CRAFT cipher, it is necessary to have 16 check points (or correction points), for the new structure having only 8 of these points suffices to protect against a univariate adversary model. We recall that the univariate adversary model considers an adversary that can inject up-to a *known* limited number of faults in the entire of encryption. Therefore, the new structure makes it possible to have even more efficient design for protection against fault analysis attacks.

6 Conclusion

While most of the security analysis for new block cipher designs are usually based on the assumption of independent round keys, it may happen to overestimate resistant of the design 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-key scenario. Consequently, in the same number of number of rounds, the weak-key structure of the cipher is less resistant against differential and linear cryptanalyses. However, the new structure suggests some efficiencies in the hardware implementation of the cipher: providing decryption on top of encryption with almost zero overhead, and reducing the overhead of protecting against FA attacks by factor of about two. Therefore, it is possible to use the new structure as a new block cipher with

improved hardware efficiency. All that is left here is to develop an appropriate tweakey schedule.

Acknowledgments

The work described in this paper has been partially supported by the German Research Foundation (DFG) within the project LE 3372/5-1. Besides, we want to thank Christof Beierle for his valuable comments while preparing this work.

References

[AMR+20] Anita Aghaie, Amir Moradi, Shahram Rasoolzadeh, Aein Rezaei Shahmirzadi, Falk Schellenberg, and Tobias Schneider. Impeccable circuits. *IEEE Trans. Computers*, 69(3):361–376, 2020.

[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.

[Köl14] Stefan Kölbl. CryptoSMT: An easy to use tool for cryptanalysis of symmetric primitives. Available at https://github.com/kste/cryptosmt, 2014.

[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.

[SRM20] Aein Rezaei Shahmirzadi, Shahram Rasoolzadeh, and Amir Moradi. Impeccable circuits II. In *DAC 2020*, pages 1–6. IEEE, 2020.