Related-Key Differential Cryptanalysis of Full Round CRAFT

Muhammad ElSheikh and Amr M. Youssef(✉)

Concordia Institute for Information Systems Engineering,
Concordia University, Montréal, QC, Canada
{m_elshei,youssef}@ciise.concordia.ca

Abstract. CRAFT is a lightweight tweakable block cipher introduced in FSE 2019. One of the main design criteria of CRAFT is the efficient protection of its implementations against differential fault analysis. While the authors of CRAFT provide several cryptanalysis results in several attack models, they do not claim any security of CRAFT against related-key differential attacks. In this paper, we utilize the simple key schedule of CRAFT to propose a systematic method for constructing several repeatable 2-round related-key differential characteristics with probability 2^{-2}. We then employ one of these characteristics to mount a key recovery attack on full-round CRAFT using 2^{31} queries to the encryption oracle and 2^{85} encryptions, and 2^{41} 64-bit blocks of memory.. Additionally, we manage to use 8 related-key differential distinguishers, with 8 related-key differences, in order to mount a key recovery attack on the full-round cipher with $2^{35.17}$ queries to the encryption oracle, 2^{32} encryptions and about 2^{6} 64-bit blocks of memory. Furthermore, we present another attack that recovers the whole master key with $2^{36.09}$ queries to the encryption oracle and only 11 encryptions with 2^{7} blocks of memory using 16 related-key differential distinguishers.

1 Introduction

Modern symmetric-key cryptographic primitives, such as the Advanced Encryption Standard (AES), which are likely designed for desktops and servers, cannot be easily implemented on resource-constrained devices such as sensor networks, healthcare equipment, Internet of Things (IoT) devices, and RFIDs. With the rapidly increasing demand for such devices, the National Institute for Standards and Technology (NIST) has initiated a standardization process for new lightweight cryptographic algorithms for use in resource-constrained devices. SKINNY [3], PRESENT [7], SIMON [2], and GIFT [1] are examples of such lightweight block ciphers that have been recently proposed.

The resistance against the differential cryptanalysis [6] is essential for any proposed cryptographic block ciphers. In differential cryptanalysis, for an n-bit primitive, an attacker is looking for a distinguisher ($\Delta P \rightarrow \Delta C$) where an XOR difference of two plaintexts (ΔP) gives, after some rounds, another

S. Bhasin et al. (Eds.): SPACE 2019, LNCS 11947, pp. 50–66, 2019.
https://doi.org/10.1007/978-3-030-35869-3_6

XOR difference (ΔC) with probability higher than 2^{-n}. Using this distinguisher, a key recovery attack can be performed by guessing the round keys. One of the variations of this attack is the related-key differential cryptanalysis [5] in which the attacker has the ability to query the encryption oracle asking for the encryption of two plaintexts, the first plaintext is encrypted using the secret key, and the other one is encrypted using another key related to the secret key, where such relation is known or even chosen by the attacker.

At FSE 2019, Beierle *et al.* presented CRAFT [4], a new lightweight tweakable block cipher with a block size of 64 bits and a key length of 128 bits associated with 64 bits as a tweak. One of the main design criteria of CRAFT is the efficient protection of its implementations against differential fault analysis. In the design paper, the authors provide the security analysis of CRAFT against several cryptanalysis techniques such as differential, linear, impossible differential, zero correlation, and integral cryptanalysis in the single-key and related-tweak settings. While they do not claim any security of CRAFT against the related-key differential attacks, they presented a deterministic related-key/related-tweak differential characteristic. However, this characteristic cannot be used to mount a key recovery attack. In this paper, we study in details the security of CRAFT against the related-key differential attack. More precisely,

1. We utilize the simple key schedule of CRAFT to present a systematic method of how to select the key difference in addition to the input and the output differences of the 2-round structure of CRAFT such that the input difference is the same as the output difference. Thus, the resulting 2-round characteristic is repeatable. In the same time, we also try to maximize the probability of that characteristic. Thereby, we use it as a building block for constructing a longer characteristic. To illustrate the effectiveness of this method, we present 17 repeatable 2-round characteristics, each one of them has only one active Sbox and holds with probability equals to the maximum differential probability of an active Sbox of CRAFT (2^{-2}).
2. We extend one of these characteristics to a 28-round related-key differential characteristic with probability 2^{-28}. After that, we employ it to mount a key recovery attack on full-round CRAFT using 2^{31} queries to the encryption oracle and 2^{85} encryptions, and 2^{41} 64-bit blocks of memory.
3. We can speed up the key recovery attack against the full-round CRAFT using $2^{35.17}$ queries to the encryption oracle and 2^{32} full-round encryptions. To this end, we manage to use 8 different related-key differential characteristics (with 8 related-key differences) in order to recover 96 bits from the secret master key and then we get the full master key by testing the right 96-bit key along with the remaining 32 bits of the key using 2 plaintext/ciphertext pairs.
4. Furthermore, we can perform the previous attack without the exhaustive search step and recover the whole master key with $2^{36.09}$ queries to the encryption oracle and only 11 full-round encryptions (instead of 2^{32} in the above attack) using 16 different related-key differential characteristics (with 16 related-key differences). This attack has been verified experimentally.

It should also be noted that, independent of our work, a related-key attack on CRAFT has been recently presented in [8] but with data and time complexities higher than the complexities of our attack.

The rest of this paper is organized as follows. In Sect. 2, we briefly revisit the specifications of CRAFT. A systematic method to build a repeatable 2-round related-key characteristic is explained in Sect. 3. In Sect. 4, we describe the key recovery attack against the full rounds of CRAFT using a single related-key differential characteristic. Then, the details of our attack using multiple related-key differential characteristics are presented in Sect. 5. Finally, the paper is concluded in Sect. 6.

2 Specifications of CRAFT

CRAFT [4] is a lightweight tweakable block cipher with a block size of 64 bits, a key length (K) of 128 bits, and a tweak (T) of 64 bits. The internal state of the cipher can be represented as a 4×4 square array of nibbles or as a 16-nibble vector by concatenating the rows of the square array. The notation $I_{i,j}$ is used to denote the nibble located at row i and column j of the 4×4 array. Also, a single subscript I_i denotes the nibble in the i-th position of 16-nibble vector, i.e., $I_{i,j} = I_{4i+j}$.

Tweakey Schedule. The 128-bit key K is split into two 64-bit subkeys K^0 and K^1. Similar to the internal state, the subkeys K^0 and K^1 in addition to the 64-bit input tweak T are represented as as 4×4 square array of nibbles or as a 16-nibble vector using a similar indexing technique as for the internal state. Then, four 64-bit tweakeys TK^0, TK^1, TK^2 and TK^3 are derived from K^0 and K^1 with the associated T as follows:

$$TK^0 = K^0 \oplus T, \quad TK^1 = K^1 \oplus T, \quad TK^2 = K^0 \oplus Q(T), \quad TK^3 = K^1 \oplus Q(T).$$

where $Q(T)$ is a permutation on the nibbles of the input tweak T using a permutation $\mathcal{Q} = [12, 10, 15, 5, 14, 8, 9, 2, 11, 3, 7, 4, 6, 0, 1, 13]$. In other words, the i-th nibble of $Q(T)$ $(T(Q)_i,\ 0 \leq i \leq 15)$ is equal to the $\mathcal{Q}(i)$-th nibble of T $(Q(T)_i = T_{\mathcal{Q}(i)})$. The tweakey $TK^{i \bmod 4}$ $(0 \leq i \leq 31)$ is used during the i-th round of the encryption operation in order to update the internal state.

Encryption Operation. The encryption operation proceeds as follows. First, the plaintext $m = m_0||m_1||\cdots||m_{14}||m_{15}$ (where m_i is a 4-bit nibble) is loaded into the internal state. Then, the internal state is updated by applying the full round function of CRAFT 31 times $(\mathcal{R}_i,\ 0 \leq i \leq 30)$. Finally, one more linear round$(\mathcal{R}^{'}_{31})$ is applied on the internal state to compute the ciphertext as shown in Fig. 1, where RC_i is the round constant. The full round of CRAFT $(\mathcal{R}_i)$ consists of the following five operations: MixColumn, AddConstant$_i$, AddTweakey$_i$ PermuteNibbles and SubBox as described in Fig. 2. The last round $(\mathcal{R}^{'}_{31})$ omits PermuteNibbles and SubBox operations from the full round. These operations are defined as follows,

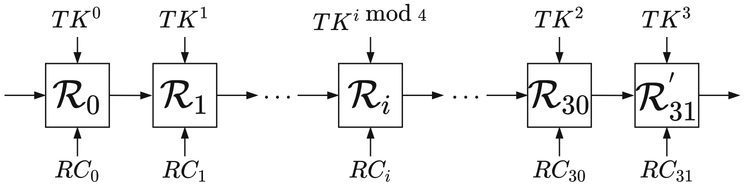

Fig. 1. Structure of `CRAFT`

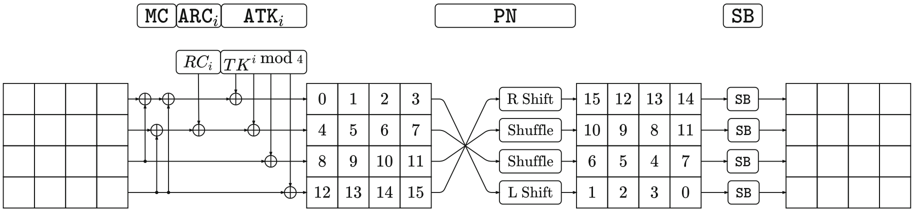

Fig. 2. One full round function of `CRAFT`

- `MixColumn (MC)`: Each column of the internal state is multiplied by a binary matrix M,

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

  This operation can be described using the XOR operation as follows. For each column j $(0 \leq j \leq 3)$,

$$\begin{bmatrix} I_{0,j} \\ I_{1,j} \\ I_{2,j} \\ I_{3,j} \end{bmatrix} \mapsto \begin{bmatrix} I_{0,j} \oplus I_{2,j} \oplus I_{3,j} \\ I_{1,j} \oplus I_{3,j} \\ I_{2,j} \\ I_{3,j} \end{bmatrix}$$

- `AddConstants`$_i$ (`ARC`$_i$): In the i-th round $(0 \leq i \leq 31)$, the internal state nibbles I_4 and I_5 are XOR-ed with the two nibbles (a and b), respectively, where a and b represented the 2-nibble round constant $RC_i = (a, b)$. These round constants are generated using 4-bit and 3-bit LFSRs. The details of generating the round constants can be found in [4].
- `AddTweakey`$_i$ (`ATK`$_i$): Each nibble of the internal state is XOR-ed with the corresponding nibble of the tweakey $TK^{i \bmod 4}$.
- `PermuteNibbles (PN)`: An permutation $\mathcal{P}$ is applied on the nibble positions of the internal state. In particular, for all $0 \leq i \leq 15$, I_i is replaced by $I_{\mathcal{P}(i)}$, where

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

- `SubBox (SB)`: A nonlinear bijective mapping applied on every nibble of the internal state in parallel using the Sbox given in Table 1.

Table 1. 4-bit Sbox of CRAFT

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

3 Related-Key Differential Characteristic of CRAFT

In this section, we describe our technique to build a repeatable 2-round related-key characteristic with a high probability p. A repeatable characteristic is a characteristic where the input difference is the same as the output difference. Hence, we can construct a long characteristic by repeating the short one n times and the probability of the long one will be p^n.

Denote the state at the input and the output of round i of CRAFT by x^i and x^{i+1}, respectively, and the state after MC, $\texttt{ARC}_i$ and $\texttt{ATK}_i$ operations by y^i. Thus we have

$$\begin{aligned} y^i &= \texttt{ATK}_i \circ \texttt{ARC}_i \circ \texttt{MC}(x^i) \\ x^{i+1} &= \texttt{SB} \circ \texttt{PN}(y^i) \end{aligned}$$

In the related-key with a single tweak model of CRAFT, the tweak (T) has zero difference, and the subkeys (K^0, K^1) have the nonzero differences ΔK^0 and ΔK^1, respectively. Thereby, the four tweaks have nonzero differences as follows

$$\Delta TK^0 = \Delta TK^2 = \Delta K^0, \qquad \Delta TK^1 = \Delta TK^3 = \Delta K^1$$

A 2-Round Characteristic. Consider two consecutive rounds, i and $i+1$, where i is even. Thus $\Delta TK^{i \bmod 4} = \Delta K^0$ and $\Delta TK^{(i+1) \bmod 4} = \Delta K^1$. We start building a repeatable 2-round characteristic by setting the input and the output differences (Δx^i and Δx^{i+2}) of the 2-round with arbitrary nonzero values such that $\Delta x^i = \Delta x^{i+2}$. Then, we deterministically propagate the input difference Δx^i forward through the MC and $\texttt{ARC}_i$ operations and choose ΔK^0 such that $\Delta K^0 = \texttt{ARC}_i \circ \texttt{MC}(\Delta x^i)$. Thereby, we ensure that $\Delta y^i = 0$, $\Delta x^{i+1} = 0$ and $\Delta y^{i+1} = \Delta K^1$. From the other direction, we propagate the output difference Δx^{i+2} backward through SB and PN operations to obtain Δy^{i+1} and select ΔK^1 such that $\Delta K^1 = \Delta y^{i+1} = \texttt{PN}_i^{-1} \circ \texttt{SB}^{-1}(\Delta x^{i+2})$. It should be noted that the probability of propagating Δx^{i+2} backward to ΔK^1 is the same as the probability of propagating ΔK^1 forward to Δx^{i+2} due to the properties of the Sbox of CRAFT. Therefore, the overall probability of this characteristic depends on the probability of propagating Δx^{i+2} through $\texttt{SB}^{-1}$ operation. In order to maximize the overall probability, we have to minimize the number of active nibbles in the input/output differences to only one active nibble with, e.g., difference value (α). Therefore, ΔK^1 also has a single active nibble with, e.g., difference value (β) such that $\Pr[\texttt{SB}^{-1}(\alpha) \rightarrow \beta] = p$. Finally, we select the value of the tuple (α, β) so that p is equal to the maximum differential probability for an active Sbox which is 2^{-2}.

Figure 3 depicts an example of such characteristics in which we set the input/output differences to zero except for the two nibbles Δx_{12}^{i} and Δx_{12}^{i+2}, which we set to α. Therefore, we select the difference of the subkey K^0 such that it has zero difference except the nibbles ΔK_0^0, ΔK_4^0 and ΔK_{12}^0 have a nonzero difference (α). For the subkey K^1, it will have zero difference in 15 nibbles and nonzero difference β in the nibble ΔK_1^1 such that $\Pr[\mathtt{SB}^{-1}(\alpha) \rightarrow \beta] = 2^{-2}$.

Based on the differential distribution table (DDT) of the CRAFT's Sbox, the unordered tuples (α, β) can take one of the values from the following set:

$$(\alpha,\beta) \text{ or } (\beta,\alpha) \in \{(\mathtt{1},\mathtt{2}),(\mathtt{2},\mathtt{4}),(\mathtt{2},\mathtt{9}),(\mathtt{2},\mathtt{c}),(\mathtt{3},\mathtt{6}),(\mathtt{5},\mathtt{7}),(\mathtt{5},\mathtt{a}), (\mathtt{7},\mathtt{d}),(\mathtt{a},\mathtt{a}),(\mathtt{a},\mathtt{d}),(\mathtt{a},\mathtt{f}),(\mathtt{b},\mathtt{b}),(\mathtt{e},\mathtt{e}),(\mathtt{f},\mathtt{f})\}. \quad (1)$$

We can also build a repeatable 2-round characteristic by setting the input and the output differences to zero differences ($\Delta x^i = \Delta x^{i+2} = 0$), then selecting ΔK^0 such that it has only one active nibble with nonzero difference (α). After that, we obtain the value of the difference ΔK^1 which will have only one active nibble with nonzero difference (β) such that $\Delta K^1 = \mathtt{ARC}_{i+1} \circ \mathtt{MC} \circ \mathtt{SB} \circ \mathtt{PN}(\Delta K^0)$. Finally, we select the value of the tuple (α, β) from the previously mentioned set. Table 2 summarizes some examples for the obtained 2-round related-key differential characteristics.

In the following sections, we utilize the repeatable 2-round related-key differential characteristics derived here to mount two key recovery attacks against the full round of CRAFT.

4 Related-Key Differential Attack Using Single Difference

In this section, we employ the repeatable 2-round characteristic ($\mathbf{RK}_0$) with, e.g., the tuple $(\alpha, \beta) = (\mathtt{4}, \mathtt{2})$ to present a related-key differential attack against the full round of CRAFT. By repeating $\mathbf{RK}_0$ (14) times as depicted in Fig. 4, we are able to construct a 28-round related-key differential characteristic (covered from round 0 to round 27) with probability $(2^{-2})^{14} = 2^{-28}$. We have verified this characteristic experimentally.

Since the characteristic ends at x^{28} with all nibbles have zero differences. After that, we propagate this difference through the last 4 rounds, and we obtain the difference at the ciphertext (ΔC) in form of

$$(\delta_4, \delta_3, \delta_9, \delta_6, \delta_4, 0, \delta_8, \delta_6, 0, \delta_3, 0, 0, \delta_4, 0, \delta_7, \delta_6).$$

Thus, we can derive the following conditions:

$$\begin{aligned}
&\Delta C_5 = \Delta C_8 = \Delta C_{10} = \Delta C_{11} = \Delta C_{13} = \mathtt{0},\\
&\Delta C_1 = \Delta C_9,\\
&\Delta C_0 = \Delta C_4 = \Delta C_{12},\\
&\Delta C_3 = \Delta C_7 = \Delta C_{15}.
\end{aligned}$$

Our attack has two phases: Data Collection phase and Key Recovery phase.

Table 2. Examples of repeatable 2-round related-key differential characteristics of `CRAFT`, all of them hold with probability 2^{-2} starting from an even round i. and (α, β) can take one of the values given by Eq. (1).

	$\Delta K^0 = \Delta TK^0 = \Delta TK^2$	$\Delta K^1 = \Delta TK^1 = \Delta TK^3$	$\Delta x^i = \Delta x^{i+2}$
$\mathbf{RK}_0$	$(0,0,0,\alpha,0,0,0,0,0,0,0,0,0,0,0,0)$	$(0,0,\beta,0,0,0,\beta,0,0,0,0,0,0,0,\beta,0)$	$(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$
$\mathbf{RK}_1$	$(\alpha,0,0,0,\alpha,0,0,0,0,0,0,0,\alpha,0,0,0)$	$(0,\beta,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,0,0,0,\alpha,0,0,0)$
$\mathbf{RK}_2$	$(0,\alpha,0,0,0,\alpha,0,0,0,0,0,0,0,\alpha,0,0)$	$(0,0,\beta,0,0,0,0,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,0,0,0,0,\alpha,0,0)$
$\mathbf{RK}_3$	$(0,0,\alpha,0,0,0,\alpha,0,0,0,0,0,0,0,\alpha,0)$	$(0,0,0,\beta,0,0,0,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,0,0,0,0,0,\alpha,0)$
$\mathbf{RK}_4$	$(0,0,0,\alpha,0,0,0,\alpha,0,0,0,0,0,0,0,\alpha)$	$(\beta,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\alpha)$
$\mathbf{RK}_5$	$(\alpha,0,0,0,0,0,0,0,\alpha,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,\beta,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,\alpha,0,0,0,0,0,0,0)$
$\mathbf{RK}_6$	$(0,\alpha,0,0,0,0,0,0,0,\alpha,0,0,0,0,0,0)$	$(0,0,0,0,0,\beta,0,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,\alpha,0,0,0,0,0,0)$
$\mathbf{RK}_7$	$(0,0,\alpha,0,0,0,0,0,0,0,\alpha,0,0,0,0,0)$	$(0,0,0,0,\beta,0,0,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,0,\alpha,0,0,0,0,0)$
$\mathbf{RK}_8$	$(0,0,0,\alpha,0,0,0,0,0,0,0,\alpha,0,0,0,0)$	$(0,0,0,0,0,0,0,\beta,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,0,0,\alpha,0,0,0,0)$
$\mathbf{RK}_9$	$(0,0,0,0,\alpha,0,0,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,0,\beta,0,0,0,0,0)$	$(0,0,0,0,\alpha,0,0,0,0,0,0,0,0,0,0,0)$
$\mathbf{RK}_{10}$	$(0,0,0,0,0,\alpha,0,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,\beta,0,0,0,0,0,0)$	$(0,0,0,0,0,\alpha,0,0,0,0,0,0,0,0,0,0)$
$\mathbf{RK}_{11}$	$(0,0,0,0,0,0,\alpha,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,\beta,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,\alpha,0,0,0,0,0,0,0,0,0)$
$\mathbf{RK}_{12}$	$(0,0,0,0,0,0,0,\alpha,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,0,0,\beta,0,0,0,0)$	$(0,0,0,0,0,0,0,\alpha,0,0,0,0,0,0,0,0)$
$\mathbf{RK}_{13}$	$(\alpha,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\beta)$	$(\alpha,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$
$\mathbf{RK}_{14}$	$(0,\alpha,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,0,0,0,\beta,0,0,0)$	$(0,\alpha,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$
$\mathbf{RK}_{15}$	$(0,0,\alpha,0,0,0,0,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,0,0,0,0,\beta,0,0)$	$(0,0,\alpha,0,0,0,0,0,0,0,0,0,0,0,0,0)$
$\mathbf{RK}_{16}$	$(0,0,0,\alpha,0,0,0,0,0,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,0,0,0,0,0,0,\beta,0)$	$(0,0,0,\alpha,0,0,0,0,0,0,0,0,0,0,0,0)$

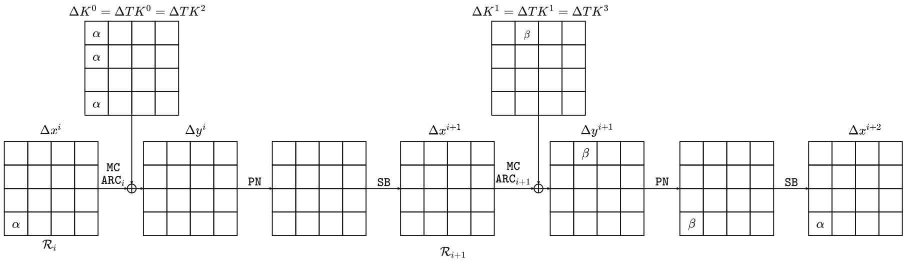

Fig. 3. A repeatable 2-round related-key characteristic of CRAFT with probability 2^{-2}.

4.1 Data Collection

We select a set of 2^m 64-bit plaintexts associated with a 64-bit tweak in which the plaintexts and the tweak can take any arbitrary values. Each plaintext is encrypted twice, using the secret master key $(K^0||K^1)$ and using the secret master key XORed with the key differences $((K^0 \oplus \Delta K^0)||(K^1 \oplus \Delta K^1))$. Then, we compute the difference at the ciphertext (ΔC) and filter out the plaintext/ciphertext pairs that do not satisfy the conditions, obtained above, on ΔC. This step provides a $5 \times 4 + 4 + 2 \times 4 + 2 \times 4 = 40$ bits filtration. Suppose the number of the remaining plaintext/ciphertext pairs after this filtration is $2^{m'}$, then on average, $2^{m'} = 2^m \times 2^{-40} = 2^{m-40}$.

4.2 Key Recovery

We first prepare $2^{11\times 4} = 2^{44}$ counters corresponding to the 44 bits of the key involved in the analysis. After that, for each ciphertext pair in the filtered $2^{m'}$ pairs obtained in the data collection phase, we apply the following procedure:

1. Guess the key nibbles (K^1_9, K^1_{12}) and partially decrypt the ciphertext to obtain the differences $(\Delta y^{30}_1, \Delta y^{30}_5)$. The average number of the guessed keys that satisfy the condition $(\Delta y^{30}_1 = \Delta y^{30}_5)$ is $2^{2\times 4} \times 2^{-4} = 2^4$.
2. Guess the key nibbles $(K^1_6, K^1_{14}, K^1_{15})$ and partially decrypt the ciphertext to obtain the values and differences at the nibbles $(y^{30}_0, y^{30}_3, y^{30}_8)$ and discard any key that does not lead to satisfy the condition of $(\Delta y^{30}_0 = \Delta y^{30}_8)$. The average number of the keys passing this filtration is $2^4 \times 2^{3\times 4} \times 2^{-4} = 2^{12}$.
3. Guess the value of $(K^1_2 \oplus K^1_{10})$ with associated value of K^1_{14} passed the filtration on the previous step (step 2) and partially decrypt the ciphertext to obtain the values and the differences at the nibbles (y^{30}_4, y^{30}_{13}). Then filter out the keys if the difference(Δy^{30}_{13}) is not the same as the differences $(\Delta y^{30}_1, \Delta y^{30}_5)$ that are obtained in the step (1). Thus, the average number of keys suggested by a pair after this step is $2^{12} \times 2^4 \times 2^{-4} = 2^{12}$.
4. Guess the key nibbles (K^0_8, K^0_{13}) and partially decrypt the nibbles (y^{30}_8, y^{30}_{13}) obtained on steps (2,3), respectively, and get the differences $(\Delta y^{29}_2, \Delta y^{29}_6)$. The average number of the guessed keys that satisfy the condition of $(\Delta y^{29}_2 = \Delta y^{29}_6)$ is $2^{12} \times 2^{2\times 4} \times 2^{-4} = 2^{16}$.

5. Guess the key nibble (K_7^1) and use the previous guessed value of K_{15}^1 to partially decrypt the ciphertext in order to obtain the value of y_{11}^{30}. Also, guess the key nibbles value of $(K_0^1 \oplus K_8^1)$ and use the previous guess of K_{12}^1 to obtain the value of y_{15}^{30}. The average number of keys suggested by a pair after this step is $2^{16} \times 2^{2\times 4} = 2^{24}$.
6. Use the value and the difference at (y_3^{30}) from step (2) with the values $(y_{11}^{30}, y_{15}^{30})$ obtained from the previous step to get the value and the difference at (y_{14}^{29}) by guessing the value of $(K_3^0 \oplus K_{11}^0 \oplus K_{15}^0)$. We then filter out the keys if the difference(Δy_{14}^{29}) is not the same as the differences $(\Delta y_2^{29}, \Delta y_6^{29})$ that are obtained in the step (4). Thus, the average number of keys suggested by a pair after this step is $2^{24} \times 2^4 \times 2^{-4} = 2^{24}$.
7. Use the previously guessed value of the key nibble (K_{14}^1) to partially decrypt the nibble y_{14}^{29} to obtain the difference Δy_3^{28} and discard the keys if the condition of $(\Delta y_3^{28} = 4)$ is not satisfied. Consequently, the average number of keys suggested by a pair after this procedure will be decreased to $2^{24} \times 2^{-4} = 2^{20}$. Thus, we increment the corresponding 2^{20} counters.

After repeating the above procedure for $2^{m'}$ pairs, we select the key corresponding to the highest counter as a 44-bit right key. Then, we recover the 128-bit master key by testing the 44-bit right key along with the remaining 84 bits of the master key that are not involved in the analysis using 2 plaintext/ciphertext pairs.

4.3 Attack Complexity and Success Probability

In what follows, we present the complexity analysis of the attack in order to determine the required number of chosen plaintexts and the memory required to launch this attack.

Data Complexity. We utilize the concept of signal-to-noise ratio (S/N) [6] in order to determine the required number of chosen plaintext/ciphertext pairs (2^m). $S/N = \frac{2^k \times p}{\alpha \times \beta}$, where k is the number of key bits involved in the analysis, p is the probability of the differential characteristic, α is the number of guessed keys by a pair, and β is the ratio of the pairs that are not discarded. In our analysis, $k = 44$, $p = 2^{-28}$, $\alpha = 2^{20}$, and $\beta = 2^{-40}$. Therefore, we have $S/N = \frac{2^{44} \times 2^{-28}}{2^{20} \times 2^{-40}} = 2^{36}$. Due to this high S/N, we can use the recommendation of Biham and Shamir [6] that 3–4 right pairs are sufficient enough to mount a successful differential attack. Therefore, we select the number of plaintext/ciphertext pairs (2^m) equal to $4 \times p^{-1} = 2^{30}$. Consequently, the data complexity will be 2^{31} chosen plaintexts.

During the data collection phase, we discard the pairs that do not satisfy the conditions on the differences of the ciphertext. The probability of satisfying these conditions is 2^{-40}, i.e., there are, on average, $2^{m-40} = 2^{30-40} = 2^{-10}$ remaining pairs. This means that the right pairs only pass this filtration and $2^{m'} = 4$.

According to [9] and due to the high S/N, the success probability of the attack (P_s) can be calculated as $P_s \approx \Phi(\sqrt{p \times 2^m})$ where Φ is the cumulative distribution function of the standard normal distribution. Therefore, our differential attack will succeed with probability $P_s \approx 0.9772$.

Time Complexity. During the key recovery phase, we perform several partial decryption of some nibbles which we can consider as $\frac{1}{16}$ of 1-round decryption. The dominant time complexity of the key recovery procedure comes from step 6 in which we perform $2^{m'} \times 2^4 \times 2^{24} \times 2 = 2^{31}$ partial decryption of 3 nibbles. This time equals to $\frac{1}{16} \times \frac{1}{32} \times 2^{31} = 2^{22}$ 32-round encryptions. Then, we perform the exhaustive search over the remaining 2^{84} keys using 2 plaintext/ciphertext pairs. The time complexity of this step is $2 \times 2^{84} = 2^{85}$ 32-round encryptions. Therefore, the total time complexity of the attack is $2^{22} + 2^{85} \approx 2^{85}$ encryptions.

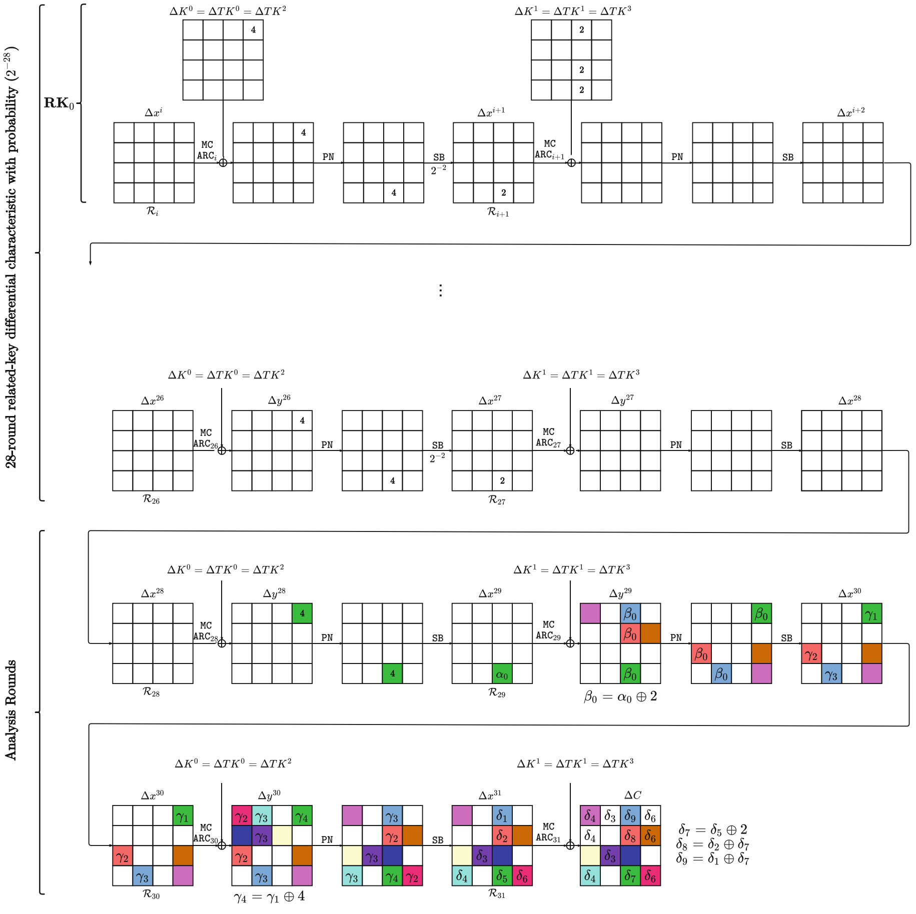

Fig. 4. The related-key differential attack against Full CRAFT using the repeatable 2-round related-key differential characteristic ($\mathbf{RK}_0$) where the colored cells are known values and differences.

Memory Complexity. The dominant part of the memory complexity comes from storing 2^{44} counters. Since the upper limit of each counter is $2^{m'} = 4$, we can store each counter in one byte. Therefore, we need $2^{44} \times \frac{8}{64} = 2^{41}$ 64-bit blocks of memory.

5 Related-Key Differential Attack Using Multiple Differences

In this section, we present a key recovery attack in the related-key model against the full-round `CRAFT` with $2^{35.17}$ queries to the encryption oracle and 2^{32} full-round encryptions. To this end, we manage to use 8 different related-key differential characteristics in order to recover 96 bits (represented in 24 nibbles) from the secret master key and then we get the full master key by testing the right 96-bit key along with the remaining 32 bits of the key using 2 plaintext/ciphertext pairs. Moreover, we can omit the exhaustive search step and recover the whole master key with $2^{36.09}$ queries to the encryption oracle and only 11 full-round encryptions.

30-Round Related-Key Differential Characteristics. We employ the repeatable 2-round characteristics ($\mathbf{RK}_1 - \mathbf{RK}_8$) (see Table 2) with the tuple $(\alpha, \beta) = (\mathtt{4}, \mathtt{2})$ in order to build eight 30-round characteristics as follows. First, we repeat each $\mathbf{RK}_i$ $(1 \leq i \leq 8)$ 14 times to build a 28-round characteristic with probability $(2^{-2})^{14} = 2^{-28}$. Then, we append another 2 rounds with probability of (2^{-2}). Thus, we are able to construct a 30-round characteristic with total probability (p) of 2^{-30}. Figure 5 depicts the 30-round characteristic that is built using $\mathbf{RK}_1$.

Consequently, we use these characteristics one by one to collect 8 datasets $(\mathcal{D}_i, 1 \leq i \leq 8)$ (Data Collection phase) and then apply a partial-key recovery process to determine a part of the master secret key (Key Recovery phase).

5.1 Data Collection

We use the 30-round characteristic based on the repeatable 2-round characteristic, e.g., $\mathbf{RK}_1$ to build the dataset $\mathcal{D}_1$ as follows. This characteristic ends at x^{30} with zero differences in all nibbles except $\Delta x^{30}_{12} = 1$ as depicted in Fig. 5. After that, by propagating this difference through the last two rounds, we are able to obtain the difference at the ciphertext (ΔC) in the form

$$(0, \delta_0, \beta_0, \gamma_0, 0, 0, 0, \gamma_0, 0, 0, \beta_0, 0, 0, 0, 0, \gamma_0)$$

where $\delta_0 = \alpha_0 \oplus 2$ and based on the DDT of `CRAFT` Sbox, $\alpha_0, \beta_0, \gamma_0 \in \{\mathtt{0}, \mathtt{4}, \mathtt{7}, \mathtt{9}, \mathtt{a}, \mathtt{c}\}$. Thus, we can derive the following conditions on the difference of the ciphertext:

$$\Delta C_i = \mathtt{0},\ i \in \{0, 4, 5, 6, 8, 9, 11, 12, 13, 14\}, \qquad \Delta C_1 = \delta_0,$$
$$\Delta C_2 = \Delta C_{10} = \beta_0, \qquad \Delta C_3 = \Delta C_7 = \Delta C_{15} = \gamma_0.$$

Consequently, we first select a set of $4 \times p^{-1} = 4 \times 2^{30} = 2^{32}$ arbitrary plaintexts ($\mathcal{L}_0$) and then we create another set of 2^{32} plaintexts ($\mathcal{L}_1$) by XORing each plaintext in the first set $\mathcal{L}_0$ with the input difference. After encrypting the two sets ($\mathcal{L}_0, \mathcal{L}_1$) using $(K^0||K^1)$ and $((K^0 \oplus \Delta K^0)||(K^1 \oplus \Delta K^1))$, respectively, we discard the pairs where the output difference does not match the required output difference (ΔC). The probability of getting (ΔC) is $2^{-(10\times4+4+2\times4)} \times (\frac{6}{16})^3 \approx 2^{-56.25}$. In other words, only the right pairs can pass this filtration. Thus, we collect, on average, 4 right pairs that follow the characteristic.

We repeat the same approach using the same set of plaintexts ($\mathcal{L}_0$) with other sets of plaintexts $\mathcal{L}_i, (2 \leq i \leq 8)$, selected like $\mathcal{L}_1$, in order to construct the datasets $\mathcal{D}_i$, $(1 \leq i \leq 8)$ using the 30-round characteristic that has been built using $\mathbf{RK}_i$, $(1 \leq i \leq 8)$ in order to get 4 right pairs per each dataset.

5.2 Key Recovery

We first prepare 24 groups of counters in which each group consists of 16 counters. Each group corresponds to a nibble of the key involved in the analysis. After that, we perform the attack in three sequential stages as follows.

First Stage. In this stage, we manage to determine the nibbles $K^1_i, (8 \leq i \leq 15)$. For example, we determine the right value of K^1_{15} as follows. We consider the group of counters corresponding to K^1_{15}, then for each right pair in the datasets $\mathcal{D}_1$ and $\mathcal{D}_5$, we guess K^1_{15} and decrypt the ciphertext nibble (C_{15}) (see Figs. 5 and 6), then increment the counter corresponding to the guessed value if the difference $\Delta y^{30}_0 = 5$. After repeating these steps for all the pairs, we select the value corresponding to the highest counters as the right value for K^1_{15}.

By repeating these steps, we are able to obtain the right values of the nibbles $K^1_i, (8 \leq i \leq 15)$. Table 3 summarizes which datasets are used to recover these nibbles.

Second Stage. After finishing the first stage, we have the right value of the key nibbles $K^1_8, K^1_9, K^1_{10}, K^1_{11}, K^1_{12}, K^1_{13}, K^1_{14}, K^1_{15}$. During this stage, we obtain the right value of another 8 nibbles $K^1_0, K^1_1, K^1_2, K^1_3, K^0_{12}, K^0_{13}, K^0_{14}, K^0_{15}$. To this end, we consider, for example, the groups of counters corresponding to the key nibbles K^1_1 and K^0_{12}, respectively. After that, we reuse the dataset $\mathcal{D}_1$ (see Fig. 5) in order to carry out the following steps:

1. Use the key nibbles K^1_9 and K^1_{13} determined in the first stage to partially decrypt the ciphertext nibbles (C_9, C_{13}) and obtain the values of the nibbles x^{31}_9 and x^{31}_{13}, respectively.
2. Guess K^1_1 and partially decrypt the ciphertext nibble C_1 to get the value and the difference at y^{30}_{12}, after that, increment the counter corresponding to the value of K^1_1 in case of $\Delta y^{30}_{12} = 5$.
3. Determine the right value of the key nibble K^1_1 by observing the highest counter.

Table 3. Key recovery

Key nibble	Dataset used	Key nibble	Dataset used
K_0^0	$\mathcal{D}_{13}$	K_0^1	$\mathcal{D}_4$
K_1^0	$\mathcal{D}_{14}$	K_1^1	$\mathcal{D}_1$
K_2^0	$\mathcal{D}_{15}$	K_2^1	$\mathcal{D}_2$
K_3^0	$\mathcal{D}_{16}$	K_3^1	$\mathcal{D}_3$
K_4^0	$\mathcal{D}_9$	K_4^1	$\mathcal{D}_7$
K_5^0	$\mathcal{D}_{10}$	K_5^1	$\mathcal{D}_6$
K_6^0	$\mathcal{D}_{11}$	K_6^1	$\mathcal{D}_5$
K_7^0	$\mathcal{D}_{12}$	K_7^1	$\mathcal{D}_8$
K_8^0	$\mathcal{D}_5$	K_8^1	$\mathcal{D}_3$
K_9^0	$\mathcal{D}_6$	K_9^1	$\mathcal{D}_2$
K_{10}^0	$\mathcal{D}_7$	K_{10}^1	$\mathcal{D}_1$
K_{11}^0	$\mathcal{D}_8$	K_{11}^1	$\mathcal{D}_4$
K_{12}^0	$\mathcal{D}_1$	K_{12}^1	$\mathcal{D}_2$, $\mathcal{D}_6$
K_{13}^0	$\mathcal{D}_2$	K_{13}^1	$\mathcal{D}_3$, $\mathcal{D}_7$
K_{14}^0	$\mathcal{D}_3$	K_{14}^1	$\mathcal{D}_4$, $\mathcal{D}_8$
K_{15}^0	$\mathcal{D}_4$	K_{15}^1	$\mathcal{D}_1$, $\mathcal{D}_5$

4. Guess K_{12}^0 and decrypt y_{12}^{30} to get the difference Δy_1^{29}, then increment the counter corresponding to the value of K_{12}^0 if $\Delta y_1^{29} = 2$.
5. Determine the right value of the key nibble K_{12}^0 by observing the highest counter.

In the same manner, we reuse the datasets $\mathcal{D}_2, \mathcal{D}_3$ and $\mathcal{D}_4$ to determine the right values of the key nibbles (K_2^1, K_{13}^0), (K_3^1, K_{14}^0), (K_0^1, K_{15}^0), respectively.

Third Stage. Similar to the second stage, we reuse the datasets $\mathcal{D}_5, \mathcal{D}_6$, $\mathcal{D}_7$ and $\mathcal{D}_8$ to recover the key nibbles $K_i^1, (4 \le i \le 7)$ and $K_j^0, (8 \le j \le 11)$ as follows. To recover the nibbles K_6^1 and K_8^0, we consider the groups of counters corresponding them, and we reuse the dataset $\mathcal{D}_5$ (see Fig. 6) in order to carry out the following steps:

1. Use the key nibble K_{14}^1 determined in the first stage to partially decrypt the ciphertext nibbles (C_{14}) to obtain the value of the nibble x_{14}^{31}.
2. Guess K_6^1 and get the value and difference at y_8^{30}, then increment the counter corresponding to the value of K_6^1 in case of $\Delta y_8^{30} = 5$.
3. Determine the right value of the key nibble K_6^1 by observing the highest counter.
4. Guess K_8^0 and decrypt y_8^{30} to get the difference Δy_6^{29}, then increment the counter corresponding to the value of K_8^0 if $\Delta y_6^{29} = 2$.

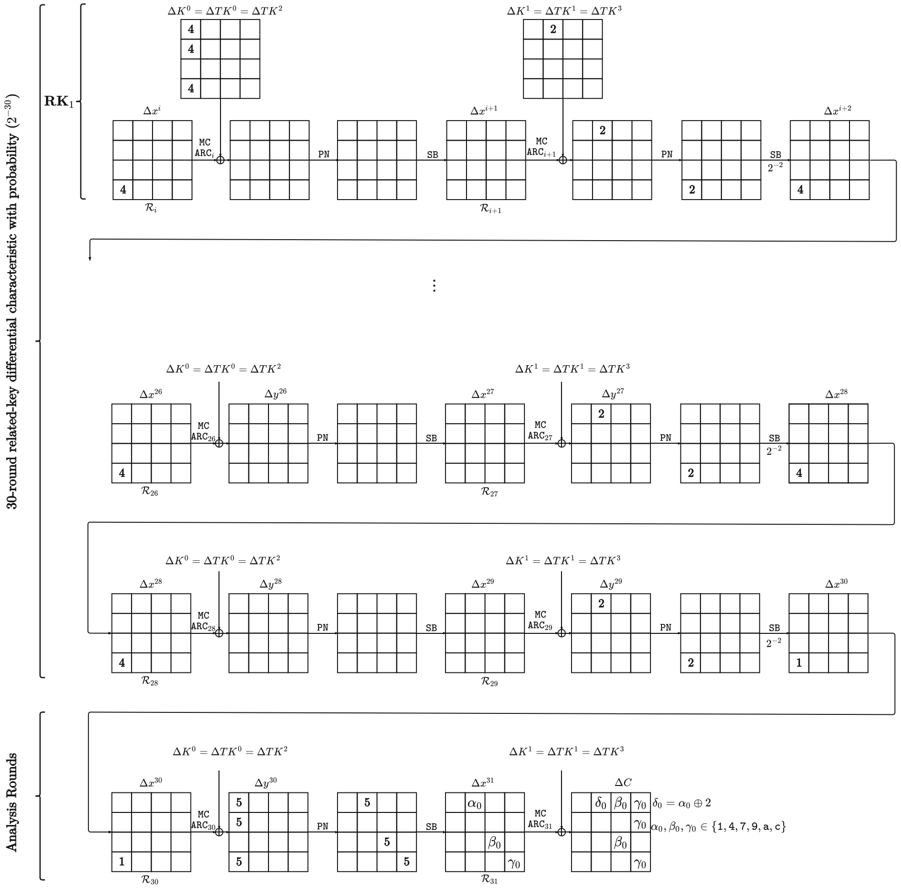

Fig. 5. Related-key differential attack against Full CRAFT using the dataset $(\mathcal{D}_1)$ to recover $K_1^1, K_{10}^1, K_{15}^1$, and K_{12}^0.

5. Determine the right value of the key nibble K_6^0 by observing the highest counter.

Using the same approach, we are able to determine the right values of the key nibbles (K_5^1, K_9^0), (K_4^1, K_{10}^0) and (K_7^1, K_{11}^0) using the datasets $\mathcal{D}_6, \mathcal{D}_7$ and $\mathcal{D}_8$, respectively.

5.3 Attack Complexity

Each set of plaintexts $\mathcal{L}_0, \cdots, \mathcal{L}_8$ contains 2^{32} plaintexts. Thus, we need $9 \times 2^{32} \approx 2^{35.17}$ queries to the encryption oracle.

During the first stage of the key recovery phase, we determine 4 nibbles using 32 right pairs and another 4 nibbles using 16 right pairs, therefore, we execute $2 \times (32 + 16) \times 2^4 = 2^{10.58}$ single nibble encryptions. For the second stage, we

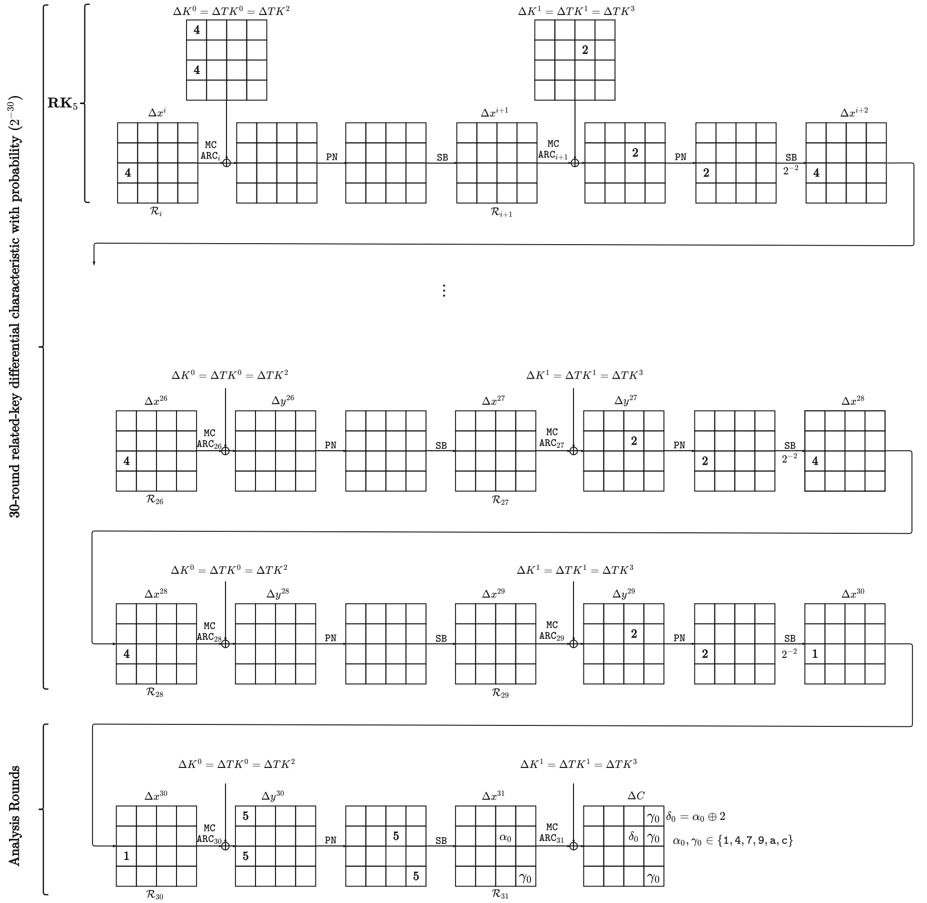

Fig. 6. Related-key differential attack against Full CRAFT using the dataset ($\mathcal{D}_5$) to recover K_6^1, K_{15}^1, and K_8^0.

recover another 8 nibbles using 4 right pairs per each nibble. This process needs $2 \times 4 \times 4 \times (2 + 2^4 + 2^4) = 2^{10.08}$ single nibble encryptions. The third stage needs $2 \times 4 \times 4 \times (1 + 2^4 + 2^4) = 2^{10.04}$ single nibble encryptions. Therefore, these three stages need $2^{12.32}$ single nibble encryptions which is equivalent to $2^{11.83} \times \frac{1}{16} \times \frac{1}{32} \approx 8$ full-round encryptions. After these stages, we run exhaustive search over the remaining 2^{32} keys using 2 plaintext/ciphertext pairs and this step needs $2^{32} = 2^{33}$ full-round encryptions.

The dominant part of the memory complexity of this stage is for storing $4 \times 8 = 32$ right pairs in addition to the 128-bit right key. Therefore, the memory complexity is $2 \times 32 + 2 = 66$ 64-bit blocks.

5.4 Omitting the Exhaustive Search Step

In this section, we describe how we can omit the exhaustive search over 2^{32} keys. To this end, we utilize the repeatable 2-round characteristics $\mathbf{RK}_9 - \mathbf{RK}_{16}$ to build another 8 30-round characteristics. Then, we employ these characteristics to construct the datasets $\mathcal{D}_1 - \mathcal{D}_{16}$ to get, on average, 4 right pairs per each dataset as we do before.

To determine the right value of the key nibbles $K_i^0, (0 \leq i \leq 7)$, we first prepare 16 counters per each nibble. Then, we partially decrypt some nibbles of the ciphertexts. After that, we guess the key nibble and increment the counters if a specific nibble at the state y^{29} has a difference equal to 2, as we do in the second and the third stages before. The ciphertext nibbles to be decrypted in addition to the position of the checked nibble at the state y^{29} and the used dataset depend on which key nibble we recover (see Table 3).

In this case, we need $17 \times 2^{32} \approx 2^{36.09}$ queries to the encryption oracle. In addition to the 8 full-round encryptions required during the previous three stages, we need $2 \times 4 \times 4 \times (6 + 2^4) = 2^{9.46}$ single nibble encryptions to recover the nibbles $K_0^0 - K_3^0$ and $2 \times 4 \times 4 \times (4 + 2^4) = 2^{9.32}$ single nibble encryptions to recover the nibbles $K_4^0 - K_7^0$. Thus, we need $8 + ((2^{9.46} + 2^{9.32}) \times \frac{1}{16} \times \frac{1}{32}) \approx 11$ full-round encryptions. Also, we need more $2 \times 4 \times 8 = 64$ block of memory to store the right pairs. Thus, the total memory complexity will be $66 + 64 = 130$ blocks of memory.

6 Conclusion

In this paper, we studied the security of the lightweight tweakable block cipher CRAFT against the related-key differential cryptanalysis. More precisely, we described a systematic method to build a repeatable 2-round related-key differential characteristic that holds with the probability of 2^{-2}. We utilized this method to build several 30-round related-key differential characteristics with probability 2^{-30}. Then, we employed these characteristics to mount a key recovery attack against the full round of CRAFT in practical time. Moreover, we have verified this attack experimentally.

References

1. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT: a small present. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 321–345. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66787-4_16
2. Beaulieu, R., Treatman-Clark, S., Shors, D., Weeks, B., Smith, J., Wingers, L.: The SIMON and SPECK lightweight block ciphers. In: 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–6 (2015)
3. Beierle, C., et al.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 123–153. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_5

4. Beierle, C., Leander, G., Moradi, A., Rasoolzadeh, S.: CRAFT: lightweight tweakable block cipher with efficient protection against DFA attacks. IACR Trans. Symmetric Cryptol. **2019**(1), 5–45 (2019). https://tosc.iacr.org/index.php/ToSC/article/view/7396
5. Biham, E.: New types of cryptanalytic attacks using related keys. J. Cryptol. **7**(4), 229–246 (1994)
6. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. Springer, New York (1993)
7. Bogdanov, A., et al.: PRESENT: an ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74735-2_31
8. Hadipour, H., Sadeghi, S., Niknam, M.M., Bagheri, N.: Comprehensive security analysis of CRAFT. Cryptology ePrint Archive, Report 2019/741 (2019). https://eprint.iacr.org/2019/741
9. Selçuk, A.A.: On probability of success in linear and differential cryptanalysis. J. Cryptol. **21**(1), 131–147 (2008)