Received: 26 November 2019 | Revised: 9 September 2020 | Accepted: 5 October 2020

DOI: 10.4218/etrij.2019-0539

ORIGINAL ARTICLE

ETRI Journal WILEY

Key-dependent side-channel cube attack on CRAFT

Kok-An Pang | **Shekh Faisal Abdul-Latip**

INSFORNET, Centre for Advanced Computing Technology (C-ACT), Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

Correspondence
Kok-An Pang and Shekh Faisal Abdul-Latip, INSFORNET, Centre for Advanced Computing Technology (C-ACT), Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia.
Email: pangkokan@gmail.com (Kok-An Pang), shekhfaisal@utem.edu.my (Shekh Faisal Abdul-Latip)

Funding information
This research was supported by the UTeM Zamalah Scheme and Fundamental Research Grant Scheme (FRGS) of Universiti Teknikal Malaysia Melaka (FRGS/1/2015/ICT05/FTMK/02/F00293) funded by the Ministry of Higher Education, Malaysia

Abstract

CRAFT is a tweakable block cipher introduced in 2019 that aims to provide strong protection against differential fault analysis. In this paper, we show that CRAFT is vulnerable to side-channel cube attacks. We apply side-channel cube attacks to CRAFT with the Hamming weight leakage assumption. We found that the first half of the secret key can be recovered from the Hamming weight leakage after the first round. Next, using the recovered key bits, we continue our attack to recover the second half of the secret key. We show that the set of equations that are solvable varies depending on the value of the key bits. Our result shows that 99.90% of the key space can be fully recovered within a practical time.

KEYWORDS

Block cipher, CRAFT, cryptanalysis, cube attack, side-channel attack

1 | INTRODUCTION

Resistance against known standard attacks has become one of the criteria for measuring the security of a block cipher. Cryptanalytic attacks such as linear and differential cryptanalysis [1,2] have been used widely to facilitate such security evaluations [3–7]. However, a cipher that can resist standard attacks may not necessarily be secure against side-channel attacks, which exploits the weaknesses in its physical implementation. Leaked information such as timing information [8], power consumption [9,10], and electromagnetic leaks [11] can be exploited for key recovery. Ciphers which can resist standard attacks [12] are not necessarily secure. They can be broken from the weaknesses of their implementation, Which have been shown in [17]. However, the feasibility of side-channel attacks varies depending on the implementation, even if the same cipher is adopted. Nevertheless, it is important to study the capabilities of available ciphers to protect communications across various devices. Unfortunately, not all ciphers are designed to resist side-channel attacks. In practice, additional countermeasures against side-channel attacks are implemented, which are mostly inefficient and costly [24]. Resistance against side-channel attacks at the algorithmic level helps reduce implementation costs by avoiding extra countermeasures, as can be seen in ciphers such as PICARO [25], ZORRO [26], and FIDES [27], where countermeasures against side-channel analysis are defined at the algorithmic level.

CRAFT [24] is a new tweakable block cipher introduced in 2019, which can resist differential fault analysis. CRAFT is also resistant to various known standard attacks [1,28–31].

Moreover, the CRAFT algorithm is nearly involutory. The encryption algorithm of CRAFT can be turned into decryption with minimal area cost.

In this paper, we analyze the security of the CRAFT implementation against side-channel cube attacks using the Hamming weight leakage model. The cube attack used in this study is key-dependent, that is, the superpoly equations used vary depending on the secret key. The selection of superpoly equations depends on the value of their right-hand sides, which are known during the online phase. Moreover, the recovery of the second half of the secret key depends on the first half of the secret key.

The superpoly equations obtained during the preprocessing phase are sufficient for recovering all secret key bits within a practical time with a success probability of 0.9990. According to Liskov and colleagues in Ref. [31], the additional tweak on a tweakable block cipher should not be considered as another uncertainty to the adversary. Furthermore, the security of a block cipher should not be compromised even if the adversary has control of the tweak. However, our result shows that the secret key of CRAFT can be recovered using side-channel cube attacks when an adversary has control of its tweak.

Our Contribution. We analyze the security of CRAFT against side-channel cube attacks with the Hamming leakage assumption. Although the algorithm of CRAFT is claimed by the designers to be resistant to differential fault analysis, which is a type of side-channel attack, our work shows that CRAFT is not secure against cube attacks within the side-channel attack model. To the best of our knowledge, our attack on CRAFT is the first attack on CRAFT within a side-channel attack model. We point out that most of the key space (that is, 99.90%) can be recovered within a practical time, but the remaining 0.1% can only be recovered with a complexity faster than brute force.

Organization of the Paper. In Section 2, we describe the notation used in this study. In Section 3, we briefly review the design of the CRAFT block cipher. Section 4 presents an outline of a side-channel cube attack. Sections 5 and 6 show the application of side-channel cube attacks on CRAFT and describe the result of our work. Section 7 concludes the paper.

2 | NOTATION

We distinguish between the addition of $\mathbb{F}_2$ and the addition of $\mathbb{Z}$, and we use $\oplus$ as the addition of $\mathbb{F}_2$ and $+$ as the addition of $\mathbb{Z}$, respectively. For summation, we denote $\sum$ as the summation of $\mathbb{Z}$, whereas $\oplus$ is the summation of $\mathbb{F}_2$. Table 1 shows the notation used in this study, unless otherwise stated. Note that all indexing in this paper is based on zero-based numbering (that is, $\mathscr{R}_0$ is the first round). Bit positions in a vector are labeled in Big Endian with zero-based numbering, that is, the most significant bit of a vector is labeled with index 0. Throughout this paper, capital letters are used to represent vectors, unless otherwise stated. For representing terms in a polynomial, bold letters are used. Let B denote a vector and $\mathbf{b}$ be a polynomial term. Next, we denote Bi as the i-th bit of B and $\mathbf{b}\ i$ as the i-th bit of $\mathbf{b}$, respectively, unless otherwise stated.

TABLE 1 Notation

k_i	The i-th bit of the secret key
t_i	The i-th bit of the tweak
$\mathscr{R}_i$	The round function at round i
V_i	The plaintext nibble $(v_{4i}, v_{4i+1}, v_{4i+2}, v_{4i+3})$
T_i	The tweak nibble $(t_{4i}, t_{4i+1}, t_{4i+2}, t_{4i+3})$
E_i^r	The state nibble $(e_{4i}^r, e_{4i+1}^r, e_{4i+2}^r, e_{4i+3}^r)$ before $\mathscr{R}_r$
K	The secret key
V	The plaintext, also known as E^0
C	The ciphertext, also known as E^{32}
T	The tweak
E^r	The internal state after r rounds
K_0	The first half of K consists of $(k_0, k_1, \ldots, k_{63})$
K_1	The second half of K consists of $(k_{64}, k_{65}, \ldots, k_{127})$
n	Key size
D	Degree of superpoly
I	The set of cube indices
C_I	The $\|I\|$-dimensional Boolean cube of $2^{\|I\|}$ vectors
$\mathbb{S}_{(p,i,j)}$	The set of integers $\{i, i+p, i+2p, \ldots, q\}$ for any $0 < j - q \le p$
$HW(X)$	Hamming weight of any vector X

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

FIGURE 1 State of CRAFT

3 | BRIEF DESCRIPTION OF THE CRAFT BLOCK CIPHER

In this section, we briefly describe the structure of CRAFT. For details on the design rationale, performance, and security evaluation of CRAFT, please refer to [24].

CRAFT operates with a 64-bit block size, a 128-bit secret key, and a 64-bit tweak. During the encryption, a plaintext is divided into 16 nibbles, which are grouped as a matrix similar to the Advanced Encryption Standard (AES) [32]. Figure 1 shows the AES-like representation for the internal state of CRAFT, where each number $0 \le i \le 15$ in each square box represents E_i^r for any $0 \le r \le 32$.

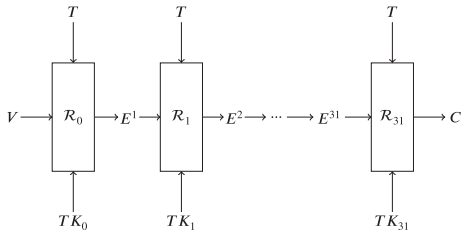

FIGURE 2 Encryption in CRAFT

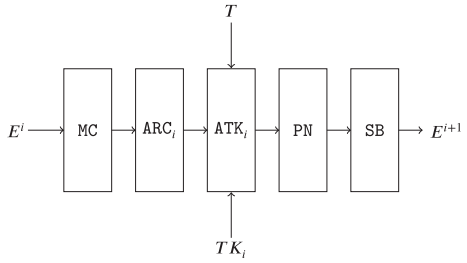

FIGURE 3 The i-th round function $\mathcal{R}_i$ of CRAFT

TABLE 2 Round constants in hexadecimal for each round

Round	Round constant	Round	Round constant
0	0 × 11	16	0 × 82
1	0 × 84	17	0 × 45
2	0 × 42	18	0 × 26
3	0 × 25	19	0 × 97
4	0 × 96	20	0 × c3
5	0 × c7	21	0 × 61
6	0 × 63	22	0 × b4
7	0 × b1	23	0 × 52
8	0 × 54	24	0 × a5
9	0 × a2	25	0 × d6
10	0 × d5	26	0 × e7
11	0 × a6	27	0 × e7
12	0 × f7	28	0 × 71
13	0 × 73	29	0 × 34
14	0 × 31	30	0 × 12
15	0 × 14	31	0 × 85

The i-th round function $\mathcal{R}_i$ of CRAFT consists of multiple operations such as MC, ARC_i, ATK_i, PN, and SB, and is defined as follows.

$$\mathcal{R}_i = \mathrm{SB} \circ \mathrm{PN} \circ \mathrm{ATK}_i \circ \mathrm{ARC}_i \circ \mathrm{MC}$$

for $i \in 0, 1, 2, \ldots, 30$ with the exception on the last round R_{31}, which does not have PN and SB. Round R_{31} is defined as follows.

$$\mathcal{R}_{31} = \mathrm{ATK}_{31} \circ \mathrm{ARC}_{31} \circ \mathrm{MC}.$$

Figures 2 and 3 show the full encryption of CRAFT and its i-th round function, respectively.

MC is a MixColumn function, which multiplies the state matrix E^i for any $0 \le i \le 32$ with an involutory matrix $\mathcal{M}$ such that $E^{i+1} = \mathrm{MC}(E^i) = \mathcal{M} \cdot E$, where.

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

Further, ARC_i is the addition of the i-th round constant RC_i (refer to Table 2), with the state $E^{i+1} = E^i \oplus RC_i$, whereas ATK_i is the addition of the i-th round tweakey TK_i with E^i such that $E^{i+1} = E^i \oplus TK_i$. Although there are 32 rounds in CRAFT, the same set of round keys are used periodically every four rounds. The round key for each round is generated as follows.

$$TK_i = K_0 \oplus T,$$

$$TK_{i+1} = K_1 \oplus T,$$

$$TK_{i+2} = K_0 \oplus Q(T),$$

$$TK_{i+3} = K_1 \oplus Q(T).$$

where Q is a permutation of tweak with all $i \in S_{(4,0,32)}$. Figure 4 shows permutation Q on tweak T, where each index i in each square box represents T_i.

PN is the permutation layer that permutes the bit positions in state E^r for any $0 \le r < 32$ such that $E_i^{r+1} = P\left(E_i^r\right)$ for all $0 \le r < 15$. Figure 5 shows the permutation of CRAFT.

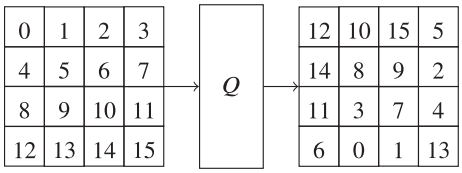

FIGURE 4 Permutation Q in key scheduling of the CRAFT block cipher

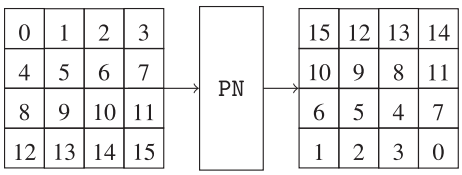

FIGURE 5 Permutation of the CRAFT block cipher

TABLE 3 Nibble substitution

E_i^r	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(E_i^r)$	C	A	D	3	E	B	F	7	8	9	1	5	0	2	4	6

SB is the nibble substitution, which acts as the non-linear layer of the state such that $E_i^{r+1} = S\left(E_i^r\right)$, which is shown in Table 3.

Every component in the round function is involutory. The decryption of CRAFT can be performed as follows.

$$\mathscr{R}_i = \mathrm{MC} \cdot \mathrm{ARC}_i \cdot \mathrm{ATK}_i \cdot \mathrm{PN} \cdot \mathrm{SB},$$

for $i = \{30, 29, \ldots, 0\}$ with the exception on the first round R_{31}, where

$$\mathscr{R}_{31} = \mathrm{MC} \cdot \mathrm{ARC}_{31} \cdot \mathrm{ATK}_{31}.$$

The decryption procedure for CRAFT can also be shown as follows.

$$\mathscr{R}_i = \mathrm{SB} \cdot \mathrm{PN} \cdot \mathrm{MC}, \left(\mathrm{ATK}_i\right) \cdot \mathrm{ARC}_i \cdot \mathrm{MC},$$

for $i = \{31, 30, 32, \ldots, 1\}$ with the exception on the last round $\mathscr{R}_0$, for which

$$\mathscr{R}_0 = \mathrm{MC}\ \mathrm{ATK}_0 \cdot \mathrm{ARC}_0 \cdot \mathrm{MC}.$$

The above decryption is similar to its encryption counterpart with round keys and round constants be placed in the reverse order while each i-th round key is generated from $\mathrm{MC}(TK_i)$.

4 | OVERVIEW OF SIDE-CHANNEL CUBE ATTACK

The cube attack proposed by Dinur and Shamir at EUROCRYPT 2009 [33] is an algebraic attack that recovers key variables from low-degree equations. The idea of a cube attack is that any master polynomial describing a cipher can be factorized by a monomial t_I, as in (1).

$$f(K, V) = t_I\left(p\left(K, V\right)\right) \oplus q\left(K, V\right), \tag{1}$$

where

$$t_I = iI \in \prod v_i \tag{2}$$

and where I represents a subset of cube indices in which all plaintext bits with such indices are active and take all possible combinations of values. As pointed out in [33], when summing $f(K,V)$ over all variables in t_I,

$$t_I C_I \in \oplus\left(t_I\left(p\left(K, V\right)\right) \oplus q\left(K, V\right)\right) = p\left(K, V\right), \tag{3}$$

where C_I is defined as an $|I|$-dimensional Boolean cube of $2^{|I|}$ vectors. The resultant equation will be a low-degree equation with respect to key variables and is easier to solve than the master polynomial of a higher degree. Such a low-degree equation is called the superpoly of t_I in $f(K,V)$ (denoted by $p(K,V)$). Having sufficient low-degree superpoly equations that are solvable from different t_Is enables us to recover the value of the key bits during the online phase using, for example, Gaussian elimination.

The problem of analyzing a block cipher with cube attacks is that the degree of the polynomial describing the cipher increases exponentially with the number of rounds. Thus, a cube attack becomes ineffective if one considers the attack within the standard attack model. Nevertheless, considering a cube attack within the side-channel attack model, the attack becomes more effective, as the adversary only requires analysis of the leaked state bits after a few rounds. Dinur and Shamir [34] introduced the concept of a side-channel cube attack in which the adversary is assumed to have access to only one bit of information about the internal state instead of the ciphertext bits. For detailed information on side-channel cube attacks and their applications, refer to [35–41].

Side-channel attacks using Hamming weight leakage have been investigated in various studies [18,42,43]. Dinur and Shamir [34] proposed reading the second Hamming weight bit from the least significant bit (LSB), which can also be found in several other works, such as in [35,37,44]. In other related work, Zhao and others [40] proposed a practical method to measure the Hamming weight leakage from a cipher implementation on an 8-bit microcontroller. This method can accurately measure the value of Hamming weight leakage with probability 1 if we collect the power traces six or more times for the same plaintext.

5 | APPLICATION OF SIDE-CHANNEL CUBE ATTACKS ON CRAFT

In this section, we describe the cube attack on CRAFT within side-channel scenarios. As described in Section 3, we realize that all round keys for even rounds in CRAFT are derived from K_0, whereas all round keys for odd rounds are derived from K_1. Therefore, we apply a side-channel cube attack on CRAFT considering the Hamming weight leakage assumption about the internal states for specific odd and even rounds of CRAFT. It is well known that measuring the leakage of internal states for specific rounds is possible in realistic scenarios. This can be achieved by controlling the clock frequency of the device, as shown in [45]. More precisely, we consider the leakage of the internal states after $\mathscr{R}_0$ and $\mathscr{R}_1$ representing the odd and even rounds, respectively. Considering these earlier rounds, we are able to obtain low-degree polynomials describing the cipher. Figure 6 shows the Hamming weight leakage in the first two rounds of CRAFT. The framework of our approach is shown in Figure 7.

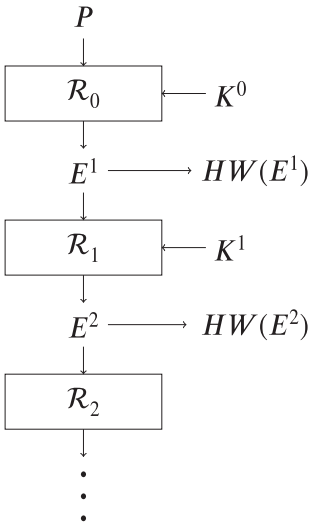

FIGURE 6 Hamming weight leakage taken at the first two rounds

STAGE 1: Consider leakage at E^1 (Figure 6)

- **Step 1**: Run the preprocessing phase of a cube attack to find sufficient superpoly equations in the key variables of K^0
- **Step 2**: Move to the online phase to find the right-hand side of the superpoly equations;
- **Step 3**: Solve the system of equations in variables of K^0.

STAGE 2: Consider leakage at E^2 (Figure 6)

- **Step 1**: Add key variables of K^0 to R^0;
- **Step 2**: Run the preprocessing phase of the cube attack to find solvable superpoly equations (Observation 1, 2, 3, and 4) in the variables of K^1 by selecting the variables of E^1 as the cubes;
- **Step 3**: Using the values of cubes at E^1 and K^0, decrypt E^1 to determine the corresponding chosen plaintext;
- **Step 4**: Move to the online phase of the cube attack to find the right-hand side of the superpoly equations using chosen plaintexts representing each cube at E^1;
- **Step 5**: Solve the system of equations for the variables of K^1.

FIGURE 7 Framework of the key-dependent side-channel cube attack on CRAFT

We divide our attack into preprocessing and online phases. During the preprocessing phase, we find superpoly equations of degree 1 and 3 from the second bits of $HW(E^1)$ and $HW(E^2)$ (from the LSB) using various cubes of sizes 2 and 3. However, we are unable to find superpoly equations of degree 2. After finding sufficient superpoly equations, we filter out those equations that are not solvable. Next, we move to the online phase to find the value of the right-hand side of each equation by summing the master polynomial over the same set of cube indices that were obtained during the preprocessing phase. To recover the value of tweakey TK_0, we solve the system of equations using Gaussian elimination. Knowledge about TK_0 enables us to control the intermediate state E^1 "implicitly" to further recover TK_1. More precisely, we determine the cube indices chosen from the tweak bits of T that yield superpoly equations of degree 1 and 3. In this case, we assume the second bit from the LSB of $HW(E^2)$ is a master polynomial over the bits in E^1 and T. For each possible value of the cube variables in T and the value of TK_0, we decrypt E^1 to obtain the corresponding plaintexts that will be used in the online phase. The selection of the cube indices from the public variables (that is, the tweak) is described formally in Observations 1–4.

Dinur and Shamir [33] used the Blum–Luby–Rubinfield (BLR) test to find linear superpoly equations. The BLR test can be further generalized to find non-linear superpoly equations, with the number of required function evaluations growing exponentially with the degree of the equation. As a result, to capture superpoly equations of degree $d \leq 3$, we choose four vectors $W, X, Y, Z \in \mathbb{F}_2^n$ independently and uniformly at random, representing samples of the n-bit key. This generalized BLR test is given by the relation

$$\begin{aligned} p(W \oplus X \oplus Y \oplus Z, V) = p(0, V) \oplus p(W, V) \oplus p(X, V) \\ \oplus p(Y, V) \oplus p(Z, V) \oplus p(W \oplus X, V) \oplus p(W \oplus Y, V) \\ \oplus p(W \oplus Z, V) \oplus p(X \oplus Y, V) \oplus p(X \oplus Z, V) \\ \oplus p(Y \oplus Z, V) \oplus p(W \oplus X \oplus Y, V) \\ \oplus p(W \oplus X \oplus Z, V) \oplus p(W \oplus Y \oplus Z, V) \\ \oplus p(X \oplus Y \oplus Z, V) \end{aligned}$$

To determine whether a particular selected cube t_I results in a superpoly equation of degree $d \leq 3$, we choose a total of 100 pairs of vectors $W, X, Y, Z \in \mathbb{F}_2^n$. If all of these vectors satisfy the generalized relation, with high probability, the superpoly equation is of degree $d \leq 3$. Because this relation will also capture constant superpoly equations (constant 0 and constant 1), we need to distinguish and eliminate them from our experiment. The constant 0 superpoly equation occurs when given any values to vectors W, X, Y, and Z, the superpoly equation is always equal to 0. However, the constant 1 superpoly equation occurs when given any values to vectors W, X, Y, and Z, the superpoly equation is always equal to 1. Next, to recover the superpoly equation of degree d, we find all terms of degree 1 until degree d and a constant term within a superpoly equation. For a detailed description of how to find the terms in the superpoly equation, we refer to Lemma 2 in [35].

After conducting the above experiment, we were able to find a substantial number of superpoly equations of degrees 1 and 3. However, the number of equations required to find the value of the key bits varies depending on the value of the right-hand side (similarly, the value of the key bits). Thus, it is impossible for us to list all the sets of equations required for all keys. Nevertheless, the set of equations required can be generalized in Observations 1, 2, 3, and 4. These generalizations are true for finding both sets of key bits, K_0 and K_1, which only differ in the indices of the key bits and the internal state used (that is, either E^0 or E^1) in selecting the cubes. In Observation 1, we generalize the linear superpoly equations found at the second bit from the LSB of $HW(E^1)$.

Observation 1. Summing a master polynomial over $t_{\alpha+1}t_{\alpha+3}$ while setting other public variables to 0 yields a superpoly equation $k_\alpha \oplus k_{\alpha+2}$ for any $\alpha \in S_{(4,0,64)}$ (to find K_0) or $\alpha \in S_{(4,64,128)}$ (to find K_1).

On the other hand, in Observation 2, we generalize cubic superpoly equations found at the same bit position of $HW(E^1)$.

Observation 2. Summing the master polynomial over $t_\beta t_{\beta+1}t_{\alpha+1}$ while setting other public variables to 0 yields a superpoly equation $k_{\beta+2} \oplus k_{\beta+2}k_\alpha k_{\alpha+2}$ for any $\beta \in S_{(4,0,64)}$ (to find K_0) or $\beta \in S_{(4,64,128)}$ (to find K_1) with $\alpha \neq \beta$.

From Observations 1 and 2, by choosing any α such that $k_\alpha \oplus k_{\alpha+2} = 1$, we can recover k_i for all $i \in S_{(2,0,128)}$. Further observations are shown in Observations 3 and 4, which can be exploited to recover k_i for all $i \in S_{(2,1,128)}$.

Observation 3. Summing the master polynomial over $t_\gamma t_{\gamma+1}t_\varphi$ while setting other public variables to 0 yields a superpoly equation $k_{\gamma+2} \oplus k_{\gamma+2}k_{\varphi+2} \oplus k_{\gamma+2}k_{\varphi+3} \oplus k_{\gamma+2}k_{\varphi+1}k_{\varphi+2} \oplus k_{\gamma+2}k_{\varphi+2}k_{\varphi+3}$ for any $\gamma, \varphi \in S_{(4,0,64)}$ (to find K_0) or $\gamma, \varphi \in S_{(4,64,128)}$ (to find K_1) with $\gamma \neq \varphi$.

Observation 4. Summing the master polynomial over $t_\varphi t_{\varphi+2}t_{\alpha+1}$ while setting other public variables to 0 yields a superpoly equation is $1 \oplus k_{\varphi+1} \oplus k_{\varphi+3} \oplus k_\alpha k_{\alpha+2} \oplus k_{\varphi+1}k_\alpha k_{\alpha+2} \oplus k_{\varphi+3}k_\alpha k_{\alpha+2}$ with $\varphi \neq \alpha$.

By choosing α and γ such that $k_\alpha \oplus k_{\alpha+2} = 1$ and $k_{\gamma+2} = 1$, we can solve the superpoly equations obtained from Observations 3 and 4 to recover k_i for all $i \in S_{(2,1,128)}$. Recovering k_i for all $i \in S_{(2,0,128)} \cup S_{(2,1,128)}$, we can fully recover secret key K.

Considering the Hamming weight leakage after $\mathcal{R}_1$, and applying the similar cube indices as used in E^0 to E^1, we have been able to find a system of equations similar to those after $\mathcal{R}_0$, only differing in the indexes of key variables. Tables A1 and A2 in Appendix A list superpoly equations resulting from the summation of the Hamming weights for E^0 and E^1.

The complexity for recovering the right-hand side of all linear superpoly equations from Observation 1 is $2^2 \cdot 16 = 2^6$ (as there are $|S_{(4,0,64)}| = 16$ different values of α). On the other hand, we require 16 non-linear superpoly equations of degree 3 for recovering all k_i with any $i \in S_{(2,0,64)}$ in Observation 2. Note that two different values of α are used here such that $\alpha_1, \alpha_2 \in S_{(4,0,64)}$. More precisely, 15 superpoly equations are chosen such that $\alpha = \alpha_1$ (as there are $|S_{(4,0,64)}| - 1 = 15$ different values of β, excluding the instance where $\beta = \alpha_1$), whereas only one superpoly is chosen, where $\alpha = \alpha_2$. Thus, this requires a $2^3 \cdot 16 = 2^7$ number of encryptions. Considering Observation 3, $|S_{(4,0,64)}| = 16$ different values of φ and two different values of γ with $\gamma_1, \gamma_2, \in S_{(4,0,64)}$ are used. Of these 16 equations, 15 superpoly equations are chosen such that $\gamma = \gamma_1$ (as there are $|S_{(4,0,64)}| - 1 = 15$ different values of φ, excluding the instances where $\varphi = \gamma_1$), whereas only one superpoly is chosen where $\gamma = \gamma_2$. As a result, this yields $2^3.16 = 2^7$ encryptions. Finally, following the same rule, the superpoly equation in Observation 4 requires $|S_{(4,0,64)}| = 16$ different values of φ and two different values of φ such that $\alpha_1, \alpha_2, \in S_{(4,0,64)}$. Similarly, 15 superpoly equations are chosen where $\alpha = \alpha_1$ (as there are $|S_{(4,0,64)}| - 1 = 15$ different values of φ, excluding the instances where $\varphi = \alpha_1$), whereas one superpoly equation is chosen where $\alpha = \alpha_2$. This also results in $2^3.16 = 2^7$ encryptions.

Summing the above number of encryptions reveals that finding K_0 requires $2^6 + 2^7 + 2^7 + 2^7 = 2^{8.81}$ computations. Recovering K_1, which also uses the same number of chosen plaintexts as in K_0, requires additional computations for partial decryption of the predetermined values of E^1. This results in recovering K_1, requiring $2^{8.81} + 2^{8.81}/32 = 2^{8.85}$ computations. Considering that we have been able to find the whole secret key K, the total number of computations required to find the correct 128-bit key of CRAFT is reduced to $2^{8.81} + 2^{8.85} = 2^{9.83}$. Applying our side-channel cube attack based on the Hamming weight leakage assumption using the method of [40] requires on average $2^{9.83} \cdot 6 = 2^{12.41}$ computations.

6 | SECRET KEYS THAT CANNOT BE RECOVERED WITHIN PRACTICAL TIME

For the secret key to be recovered in a practical time, all bits of K_0 must be recovered with probability 1, as this enables us to control the intermediate state after the first round. If K_0 cannot be recovered completely from the leakages, the whole secret key must be recovered using brute force. Because K_1 is not known, the time complexity for recovering the secret key would be larger than 2^{64} as all key bits in K_1 and some key bits in K_0 would be recovered by brute force. Note that it is impossible to recover K_1 directly from $HW(E^2)$ over bits in V, T, and K as the degree of the master polynomial is quite high because of the exponential increase in degree after each round.

There are stronger keys that cannot be recovered using side-channel cube attacks in practical time. Propositions 6 and 6 show the characteristics of such keys.

Proposition 1. Secret keys with $\sum_{i=0}^{15}\left(k_{4i} \oplus k_{4i+2}\right) \leq 1$ or $\sum_{i=16}^{31}\left(k_{4i} \oplus k_{4i+2}\right) \leq 1$ can prevent full recovery by side-channel cube attacks.

Proof. According to Observation 2, to recover $k_{\beta+2}$, the value of $k_\alpha \oplus k_{\alpha+2}$ must be equal to 1. Thus, guessing $k_{\beta+2}$ for any $\beta \in S_{(4,0,128)}$ in practical time is only possible when $\sum_{i=0}^{15}\left(k_{4i} \oplus k_{4i+2}\right) \geq 1$ and $\sum_{i=16}^{31}\left(k_{4i} \oplus k_{4i+2}\right) \geq 1$. However, recovering $k_{\beta+2}$ for

all $\beta \in S_{(4,0,128)}$ requires $\sum_{i=0}^{15}(k_{4i} \oplus k_{4i+2}) \geq 2$ and $\sum_{i=16}^{31}(k_{4i} \oplus k_{4i+2}) \geq 2$. This is because $k_{\beta+2}$ cannot be recovered practically when $\beta = \alpha$. Thus, another α such that $k_\alpha \oplus k_{\alpha+2} = 1$ and $\beta \neq \alpha$ is needed. Therefore, it is only possible to recover K_0 or K_1 completely in practical time when $\sum_{i=0}^{15}(k_{4i} \oplus k_{4i+2}) \geq 2$ and $\sum_{i=16}^{31}(k_{4i} \oplus k_{4i+2}) \geq 2$. Even if this property is known to the adversary, the adversary still needs to recover k_i for all $i \in S_{(2,0,128)}$ in a time complexity of 2^{32} (because the right-hand side of $k_\alpha \oplus k_{\alpha+2}$ is known, guessing the values of k_α and $k_{\alpha+2}$ only requires 2 encryptions). For each possible value of k_i for all $i \in S_{(2,1,128)}$, 2^{64} encryptions are required to guess all k_i for all $i \in S_{(2,1,128)}$. The total time complexity for recovering is $2^{32} \cdot 2^{64} = 2^{96}$, which is impractical. □

Preposition 2. Secret keys with $\sum_{i=0}^{15} k_{4i+2} \leq 1$ or $\sum_{i=16}^{31} k_{4i+2} \leq 1$ can prevent full recovery by side-channel cube attacks.

Proof. According to Observation 3, when $k_{\gamma+2} = 0$, the superpoly equation would have a value of 0, which is not exploitable for key recovery. Thus, guessing $k_{\varphi+1}$ and $k_{\varphi+3}$ from Observation 3 requires $\sum_{i=0}^{15} k_{4i+2} \geq 1$ and $\sum_{i=16}^{31} k_{4i+2} \geq 1$. However, it is not possible to recover $k_{\varphi+1}$ and $k_{\varphi+3}$ when $\gamma = \varphi$. To recover such key bits, another value of γ such that $k_{\gamma+2} = 1$, where $\gamma \neq \varphi$ is needed. Hence, it is only possible to recover $k_{\varphi+1}$ and $k_{\varphi+3}$ practically if $\sum_{i=0}^{15} k_{4i+2} \geq 2$ and $\sum_{i=16}^{31} k_{4i+2} \geq 2$. To recover such a secret key, an adversary must recover the remaining 96 secret key bits. The total complexity of recovering secret keys with this property requires a time complexity of 2^{96}, which is impractical. □

Considering the effectiveness of the key-dependent side-channel cube attack on CRAFT, we compute the number of secret keys fulfilling the properties mentioned in Propositions 1 and 2. As a result, for any $s \in \{0, 1\}$, there are $\binom{16}{1} = 16$ combinations of K_S in which $\sum_{i=16s}^{16s+15}(k_{4i} \oplus k_{4i+2}) = 1$, while there are only $\binom{16}{0} = 1$ combination such that $\sum_{i=16s}^{16s+15}(k_{4i} \oplus k_{4i+2}) = 0$ and the rest $2^{16} - \binom{16}{0} - \binom{16}{1} \approx 2^{16}$ combinations in which $\sum_{i=16s}^{16s+15}(k_{4i} \oplus k_{4i+2}) \geq 2$. Similarly, there are $\binom{16}{1} = 16$ combinations of K_S, where $\sum_{i=16s}^{16s+15} k_{4i+2} = 1$, $\binom{16}{0} = 1$ combination of $\sum_{i=16s}^{16s+15} k_{4i+2} = 0$ and $2^{16} - \binom{16}{0} - \binom{16}{1} \approx 2^{16}$ combinations of $\sum_{i=16s}^{16s+15} k_{4i+2} \geq 2$. Table B1 in Appendix B shows all characteristics of the secret keys that cannot be recovered within practical time and their respective number of combinations. Having all combinations of K for every condition shown in Table B1, the number of such secret keys is $2^{118.09}$. Because the selection of the secret key should be random, the probability that these keys are used is 0.001.

7 | CONCLUSIONS

In this paper, we studied the security of the CRAFT block cipher against side-channel cube attacks considering the Hamming weight leakage assumption. In the preprocessing phase, we extract superpoly equations from various combinations of cube variables. The degree of the superpoly equations is determined using the BLR test with four independent random vectors. During the online phase, we find the right-hand side of the equations extracted during the preprocessing phase and solve the system of equations using Gaussian elimination. First, we determine the right-hand side of the equations from the Hamming weight leakage after the first round, which can be used to recover the first half of the secret key, K_0. Then, with the recovered K_0, we can recover the second half of the secret key from the Hamming weight leakage after the second round by tweaking the internal state E^1. Thus, on average, the secret key can be recovered in $2^{12.41}$ computations using $2^{9.83}$ data. We also show the properties of stronger keys that can prevent side-channel cube attacks within practical time, as demonstrated in Propositions 6 and 6, and which make up only 0.1% of the key space. From our initial observation, it seems that these stronger keys could be recovered within practical time if we could recover more useful superpoly equations that are not bounded by the characteristics as in the aforementioned propositions. One could consider the leakage after E^2 and add the predetermined values of variables in K_0 and K_1, performing further analysis to extract the key considering the plaintext variables as cubes. We believe that this may simplify the master polynomials, which could make recovering the remaining secret keys within the 0.1% of the key space more practical. However, our result does not imply the actual security weakness of CRAFT within the real-world implementation environment, as our attack was only conducted within an abstract model based on the Hamming weight leakage assumption. Further investigation should be conducted to determine the actual security implications for CRAFT within the real-world attack scenario. We leave extending the attack and exploiting such superpoly equations as an interesting work for future research.

ORCID

Kok-An Pang https://orcid.org/0000-0001-5277-6925
Shekh Faisal Abdul-Latip https://orcid.org/0000-0003-3280-2422

REFERENCES

1. E. Biham and A. Shamir, *Differential cryptanalysis of des-like cryptosystems*, J. Cryptol. **4** (1991), no. 1. 3–72.
2. M. Matsui, *Linear cryptanalysis method for des cipher*, in Advances in Cryptology—EUROCRYPT '93, vol. 765, Springer, Heidelberg, Berlin, 1993, pp. 386–397.
3. Z. Liu et al., *New insights on linear cryptanalysis*, Sci. China Inform. Sci. **63** (2020), no. 1, 112104.
4. A. Flórez-Gutiérrez and M. Naya-Plasencia, *Improving key-recovery in linear attacks: Application to 28-round present*, in Advances in Cryptology-EUROCRYPT 2020, vol. 12105, Springer, Cham, Switzerland, 2020, pp. 221–249.
5. M. Huang and L. Wang, *Automatic search for the linear (hull) characteristics of arx ciphers: Applied to speck, sparx, chaskey, and cham-64*, Secur. Commun. Netw. **2020** (2020), 1–14.

6. Y. Igarashi, S. Nakazawa, and T. Kaneko, *Differential cryptanalysis of block cipher halka*, Int. J. Inform. Electron. Eng. **10** (2020), no. 2, 40–43.
7. H. Zhao et al., *Milpbased differential cryptanalysis on round-reduced midori64*, IEEE Access **8** (2020), 95888–95896.
8. D. Brumley and D. Boneh, *Remote timing attacks are practical*, Comput. Netw. **48** (2005), no. 5, 701–716.
9. P. Kocher, J. Jaffe, and B. Jun, *Differential power analysis*, in Advances in Cryptology–CRYPTO '99, vol. 1666, Springer, Heidelberg, Berlin, 1999, pp. 388–397.
10. P. Kocher et al., *Introduction to differential power analysis*, J. Cryptograph. Eng. **1** (2011), no. 1, 5–27.
11. E. De Mulder et al., *Differential electromagnetic attack on an FPGA implementation of elliptic curve cryptosystems*, in Proc. World Autom. Congress (Budapest, Hungary), July 2006, pp. 1–6.
12. O. Dunkelman et al., *Single tweakey cryptanalysis of reduced-round skinny-64*, in Cyber Security Cryptography and Machine Learning, vol. 12161, Springer, Cham, Switzerland, 2020, pp. 1–17.
13. A. Bogdanov, D. Khovratovich, and C. Rechberger, *Biclique cryptanalysis of the full AES*, in Proc. Int. Conf. Theory Applicat. Cryptol. Inf. Secur. (Seoul, Rep. of Korea), Dec. 2011, pp. 344–371.
14. K. B. Jithendra and T. K. Shahana, *New biclique cryptanalysis on fullround present-80 block cipher*, SN Comput. Sci. **1** (2020), no. 2, 1–7.
15. B. Zhu, X. Dong, and Y. Hongbo, *Milp-based differential attack on round-reduced GIFT*, in Proc. Cryptogr. Track RSA Conf. (San Francisco, CA, USA,), Mar. 2019, pp. 372–390.
16. R. Rohit, R. AlTawy, and G. Gong, *Milp-based cube attack on the reduced-round WG-5 lightweight stream cipher*, in Proc. IMA Int. Conf. Cryptogr. Coding (Oxford, UK), Dec. 2017, pp. 333–351.
17. Y. Xiao, J. Xin, and Y. Shen, *CNN based electromagnetic side channel attacks on SoC*, MS&E **782** (2020), no. 3, e032055.
18. F. Durvaux and M. Durvaux, *SCA-pitaya: A practical and affordable side-channel attack setup for power leakage–based evaluations*, Digital Threats Res. Pract. **1** (2020), no. 1, 1–16.
19. S. Saha et al., *Fault template attacks on block ciphers exploiting fault propagation*, in Proc. Annu. Int. Conf. Theory Applicat. Cryptogr. Tech. (Zagreb, Croatia), May 2020, pp. 612–643.
20. M. Hell and O. Westman, *Electromagnetic side-channel attack on AES using low-end equipment*, ECTI Transac. Comput. Inform. Technol. **14** (2020), no. 2, 139–148.
21. J. Zhang et al., *Power analysis attack on a lightweight block cipher GIFT*, in Proc. Int. Conf. Comput. Eng. Netw. (Changsha, China), Oct. 2019, pp. 565–574.
22. M. A. Orumiehchiha et al., *A differential fault attack on the WG family of stream ciphers*, J. Cryptograph. Eng. **10** (2020), no. 2, 189–195.
23. S. Bhasin et al., *SITM: See-in-the-middle sidechannel assisted middle round differential cryptanalysis on SPN block ciphers*, IACR Transac. Cryptograph. Hardw. Embedded Syst. **2020** (2020) no. 1, 95–122.
24. C. Beierle et al., *CRAFT: Lightweight tweakable block cipher with efficient protection against DFA attacks*, IACR Transac. Symmetr. Cryptol. **2019** (2019), no. 1, 5–45.
25. G. Piret, T. Roche, and C. Carlet, *PICARO—A block cipher allowing efficient higher-order side-channel resistance*, in Proc. Int. Conf. Appl. Cryptogr. Netw. Security (Singapore), June 2012, pp. 311–328.
26. B. Gérard et al., *Block ciphers that are easier to mask: How far can we go?*, in Proc. Int. Workshop Cryptogr. Hardw. Embed. Syst. (Santa Barbara, CA, USA), Aug. 2013, pp. 383–399.
27. B. Bilgin et al., *FIDES: Lightweight authenticated cipher with side-channel resistance for constrained hardware*, in Proc. Int. Workshop Cryptographic Hardw. Embedded Syst. (Santa Barbara, CA, USA), Aug. 2013, pp. 142–158.
28. A. Bogdanov and V. Rijmen, *Linear hulls with correlation zero and linear cryptanalysis of block ciphers*, Des. Codes Crypt. **70** (2014), no. 3, 369–383.
29. M. Hellman, *A cryptanalytic time-memory trade-off*, IEEE Trans. Inf. Theory **26** (1980), no. 4, 401–406.
30. L. Jiqiang et al., *New impossible differential attacks on AES*, in Proc. Int. Conf. Cryptol. (Kharagpur, India), Dec. 2008, pp. 279–293.
31. M. Liskov et al., *Tweakable block ciphers*, in Proc. Annu. Int. Cryptol. Conf. (Santa Barbara, CA, USA), Aug. 2002, pp. 31–46.
32. J. Daemen and V. Rijmen, The Design of Rijndael: The Advanced Encryption Standard (AES), 2nd ed. Springer, Heidelberg, Berlin, 2020.
33. I. Dinur and A. Shamir, *Cube attacks on tweakable black box polynomials*, in Proc. Annu. Int. Conf. Theory Applicat. Cryptographic Techniques (Cologne, Germany), Apr. 2009, pp. 278–299.
34. I. Dinur and A. Shamir, *Side channel cube attacks on block ciphers*, IACR Cryptol. ePrint Archive **2009** (2009), 1–15.
35. S. F. Abdul-Latip et al., *Extended cubes: Enhancing the cube attack by extracting low-degree non-linear equations*, in Proc. ACM Symp. Inf., Comput. Commun. Security (Hong Kong), Mar. 2011, pp. 296–305.
36. G. V. Bard et al., *Algebraic, AIDA/cube and side channel analysis of KATAN family of block ciphers*, in Proc. Int. Conf. Cryptol. (Hyderabad, India), Dec. 2010, pp. 176–196.
37. A. G. Buja, S. FaisalAbdul-Latip, and R, Ahmad, *A security analysis of iot encryption: Side-channel cube attack on simeck32/64*, arXiv preprint arXiv:1808.03557, 2018, pp. 79–90.
38. X. Fan and G. Gong, *On the security of hummingbird-2 against side channel cube attacks*, in Proc. Western Eur. Workshop Res. Cryptol. (Weimar, Germany), July. 2011, pp. 18–29.
39. L. Yang, M. Wang, and S. Qiao, *Side channel cube attack on present*, in Cryptology and Network Security, vol. 5888, Springer, Heidelberg, Berlin, 2009, pp. 379–391.
40. X. Zhao et al., *Efficient hamming weight-based sidechannel cube attacks on present*, J. Syst. Softw. **86** (2013), no. 3, 728–743.
41. E. Aghaee et al., *A practical iterative side channel cube attack on aes-128/256*, J. Comput. Technol. Appl. **5** (2019), no. 3, 31–45.
42. P. Saravanan and B. M. Mehtre, *A novel approach to detect hardware malware using hamming weight model and one class support vector machine*, in VLSI Design and Test, vol. 892, Springer, Singapore, 2019, pp. 159–172.
43. E. de Chérisey et al., *Best information is most successful*, IACR Transac. Cryptogr. Hardw. Embed. Syst. **2019** (2019) no. 2, 49–79.
44. Z. Li et al., *Cube cryptanalysis of LBlock with noisy leakage*, in Proc. Int. Conf. Inf. Security Cryptol. (Seoul, Rep. of Korea), Nov. 2012, pp. 141–155.
45. S. M. Del Pozo et al., *Side-channel attacks from static power: When should we care?*, in Proc. Design, Autom. Test Eur. Conf. Exhibition (Grenoble, France), Apr. 2015, pp. 145–150.

AUTHOR BIOGRAPHIES

Kok-An Pang is a Masters student at the Faculty of Information Technology and Communication, Universiti Teknikal Malaysia Melaka, Malaysia. He received Bachelor of Computer Science (Computer Network Security) with Honours from Universiti Sultan Zainal Abidin (UniSZA). His research interests include cryptography and cryptanalysis of cryptographic primitives.

Shekh Faisal Abdul-Latip is currently working at the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM). He received his PhD degree in 2012 from the University of Wollongong, Australia, in the field of Cryptography. Prior to his PhD studies, he obtained M.Sc in Information Security from the Royal Holloway, University of London in 2003. He obtained both B.Sc (Hons) in Computer Science (2000) and Diploma in Electronic Engineering (1994) from Universiti Teknologi Malaysia (UTM). His main research interest focuses on symmetric-key cryptography (ie, design and cryptanalysis of block ciphers, stream ciphers, hash functions and MACs). He is currently a member of focus group and one of the evaluation panel experts for the MySEAL project, that is, a project to recommend a list of trusted cryptographic algorithms to be used by the public and private sectors in Malaysia. To promote new ideas and activities in cryptology related areas in Malaysia, he joint and became a member of executive committee for the Malaysian Society for Cryptology Research (MSCR).

APPENDIX A

CUBES AND SUPERPOLYS

TABLE A1 List of cubes and the corresponding superpolys considering the second Hamming weight bit from the LSB after $\mathscr{R}_0$

Cube	Superpoly
$t_0t_1t_{60}$	$k_2 \oplus k_2k_{62} \oplus k_2k_{63} \oplus k_2k_{61}k_{62} \oplus k_2k_{62}k_{63}$
$t_0t_4t_5$	$k6 \oplus k2k6 \oplus k3k6 \oplus k1k2k6 \oplus k2k3k6$
$t_4t_5t_8$	$k6 \oplus k6k10 \oplus k_6k_{11} \oplus k_6k_9k_{10} \oplus k_6k_{10}k_{11}$
$t_4t_5t_{12}$	$k_6 \oplus k_6k_{14} \oplus k_6k_{15} \oplus k_6k_{13}k_{14} \oplus k_6k_{14}k_{15}$
$t_4t_5t_{16}$	$k_6 \oplus k_6k_{18} \oplus k_6k_{19} \oplus k_6k_{17}k_{18} \oplus k_6k_{18}k_{19}$
$t_4t_5t_{20}$	$k_6 \oplus k_6k_{22} \oplus k_6k_{23} \oplus k_6k_{21}k_{22} \oplus k_6k_{22}k_{23}$
$t_0t_2t_5$	$1 \oplus k_1 \oplus k_3 \oplus k_4k_6 \oplus k_1k_4k_6 \oplus k_3k_4k_6$
$t_0t_2t_9$	$1 \oplus k_1 \oplus k_3 \oplus k_8k_{10} \oplus k_1k_8k_{10} \oplus k_3k_8k_{10}$
$t_0t_2t_{13}$	$1 \oplus k_1 \oplus k_3 \oplus k_{12}k_{14} \oplus k_1k_{12}k_{14} \oplus k_3k_{12}k_{14}$
$t_0t_2t_{17}$	$1 \oplus k_1 \oplus k_3 \oplus k_{16}k_{18} \oplus k_1k_{16}k_{18} \oplus k_3k_{16}k_{18}$
$t_0t_2t_{21}$	$1 \oplus k_1 \oplus k_3 \oplus k_{20}k_{22} \oplus k_1k_{20}k_{22} \oplus k_3k_{20}k_{22}$
$t_0t_2t_{25}$	$1 \oplus k_1 \oplus k_3 \oplus k_{24}k_{26} \oplus k_1k_{24}k_{26} \oplus k_3k_{24}k_{26}$
$t_0t_2t_{29}$	$1 \oplus k_1 \oplus k_3 \oplus k_{28}k_{30} \oplus k_1k_{28}k_{30} \oplus k_3k_{28}k_{30}$
$t_0t_2t_{33}$	$1 \oplus k_1 \oplus k_3 \oplus k_{32}k_{34} \oplus k_1k_{32}k_{34} \oplus k_3k_{32}k_{34}$
$t_0t_2t_{37}$	$1 \oplus k_1 \oplus k_3 \oplus k_{36}k_{38} \oplus k_1k_{36}k_{38} \oplus k_3k_{36}k_{38}$
$t_0t_2t_{41}$	$1 \oplus k_1 \oplus k_3 \oplus k_{40}k_{42} \oplus k_1k_{40}k_{42} \oplus k_3k_{40}k_{42}$
$t_0t_2t_{45}$	$1 \oplus k_1 \oplus k_3 \oplus k_{44}k_{46} \oplus k_1k_{44}k_{46} \oplus k_3k_{44}k_{46}$
$t_0t_2t_{49}$	$1 \oplus k_1 \oplus k_3 \oplus k_{48}k_{50} \oplus k_1k_{48}k_{50} \oplus k_3k_{48}k_{50}$
$t_0t_2t_{53}$	$1 \oplus k_1 \oplus k_3 \oplus k_{52}k_{54} \oplus k_1k_{52}k_{54} \oplus k_3k_{52}k_{54}$
$t_0t_2t_{57}$	$1 \oplus k_1 \oplus k_3 \oplus k_{56}k_{58} \oplus k_1k_{56}k_{58} \oplus k_3k_{56}k_{58}$
$t_0t_2t_{61}$	$1 \oplus k_1 \oplus k_3 \oplus k_{60}k_{62} \oplus k_1k_{60}k_{62} \oplus k_3k_{60}k_{62}$
$t_1t_4t_6$	$1 \oplus k_5 \oplus k_7 \oplus k_0k_2 \oplus k_0k_2k_5 \oplus k_0k_2k_7$
$t_4t_6t_9$	$1 \oplus k_5 \oplus k_7 \oplus k_8k_{10} \oplus k_5k_8k_{10} \oplus k_7k_8k_{10}$
$t_4t_6t_{13}$	$1 \oplus k_5 \oplus k_7 \oplus k_{12}k_{14} \oplus k_5k_{12}k_{14} \oplus k_7k_{12}k_{14}$
$t_4t_6t_{17}$	$1 \oplus k_5 \oplus k_7 \oplus k_{16}k_{18} \oplus k_5k_{16}k_{18} \oplus k_7k_{16}k_{18}$
$t_4t_6t_{21}$	$1 \oplus k_5 \oplus k_7 \oplus k_{20}k_{22} \oplus k_5k_{20}k_{22} \oplus k_7k_{20}k_{22}$
t_1t_3	$k_0 \oplus k_2$
t_5t_7	$k_4 \oplus k_6$
t_9t_{11}	$k_8 \oplus k_{10}$
$t_{13}t_{15}$	$k_{12} \oplus k_{14}$
$t_{17}t_{19}$	$k_{16} \oplus k_{18}$
$t_{21}t_{23}$	$k_{20} \oplus k_{22}$
$t_{25}t_{27}$	$k_{24} \oplus k_{26}$
$t_{29}t_{31}$	$k_{28} \oplus k_{30}$
$t_{33}t_{35}$	$k_{32} \oplus k_{34}$
$t_{37}t_{39}$	$k_{36} \oplus k_{38}$
$t_{41}t_{43}$	$k_{40} \oplus k_{42}$
$t_{45}t_{47}$	$k_{44} \oplus k_{46}$
$t_{49}t_{51}$	$k_{48} \oplus k_{50}$
$t_{53}t_{55}$	$k_{52} \oplus k_{54}$
$t_{57}t_{59}$	$k_{56} \oplus k_{58}$
$t_{61}t_{63}$	$k_{60} \oplus k_{62}$
$t_0t_1t_5$	$k_2 \oplus k_2k_4k_6$
$t_0t_1t_9$	$k_2 \oplus k_2k_8k_{10}$
$t_0t_1t_{13}$	$k_2 \oplus k_2k_{12}k_{14}$
$t_0t_1t_{17}$	$k_2 \oplus k_2k_{16}k_{18}$
$t_0t_1t_{21}$	$k_2 \oplus k_2k_{20}k_{22}$
$t_0t_1t_{25}$	$k_2 \oplus k_2k_{24}k_{26}$
$t_0t_1t_{29}$	$k_2 \oplus k_2k_{28}k_{30}$
$t_0t_1t_{33}$	$k_2 \oplus k_2k_{32}k_{34}$
$t_0t_1t_{37}$	$k_2 \oplus k_2k_{36}k_{38}$
$t_0t_1t_{41}$	$k_2 \oplus k_2k_{40}k_{42}$
$t_0t_1t_{45}$	$k_2 \oplus k_2k_{44}k_{46}$
$t_0t_1t_{49}$	$k_2 \oplus k_2k_{48}k_{50}$
$t_0t_1t_{53}$	$k_2 \oplus k_2k_{52}k_{54}$
$t_0t_1t_{57}$	$k_2 \oplus k_2k_{56}k_{58}$
$t_0t_1t_6l$	$k_2 \oplus k_2k_{60}k_{62}$
$t_1t_4t_5$	$k_6 \oplus k_0k_2k_6$
$t_4t_5t_9$	$k_6 \oplus k_6k_8k_{10}$
$t_4t_5t_{13}$	$k_6 \oplus k_6k_{12}k_{14}$
$t_4t_5t_{17}$	$k_6 \oplus k_6k_{16}k_{18}$
$t_0t_1t_4$	$k_2 \oplus k_2k_6 \oplus k_2k_7 \oplus k_2k_5k_6 \oplus k_2k_6k_7$
$t_0t_1t_8$	$k_2 \oplus k_2k_{10} \oplus k_2k_{11} \oplus k_2k_9k_{10} \oplus k_2k_{10}k_{11}$
$t_0t_1t_{12}$	$k_2 \oplus k_2k_{14} \oplus k_2k_{15} \oplus k_2k_{13}k_{14} \oplus k_2k_{14}k_{15}$
$t_0t_1t_{16}$	$k_2 \oplus k_2k_{18} \oplus k_2k_{19} \oplus k_2k_{17}k_{18} \oplus k_2k_{18}k_{19}$
$t_0t_1t_{20}$	$k_2 \oplus k_2k_{22} \oplus k_2k_{23} \oplus k_2k_{21}k_{22} \oplus k_2k_{22}k_{23}$
$t_0t_1t_{24}$	$k_2 \oplus k_2k_{26} \oplus k_2k_{27} \oplus k_2k_{25}k_{26} \oplus k_2k_{26}k_{27}$
$t_0t_1t_{28}$	$k_2 \oplus k_2k_{30} \oplus k_2k_{31} \oplus k_2k_{29}k_{30} \oplus k_2k_{30}k_{31}$
$t_0t_1t_{32}$	$k_2 \oplus k_2k_{34} \oplus k_2k_{35} \oplus k_2k_{33}k_{34} \oplus k_2k_{34}k_{35}$
$t_0t_1t_{36}$	$k_2 \oplus k_2k_{38} \oplus k_2k_{39} \oplus k_2k_{37}k_{38} \oplus k_2k_{38}k_{39}$
$t_0t_1t_{40}$	$k_2 \oplus k_2k_{42} \oplus k_2k_{43} \oplus k_2k_{41}k_{42} \oplus k_2k_{42}k_{43}$
$t_0t_1t_{44}$	$k_2 \oplus k_2k_{46} \oplus k_2k_{47} \oplus k_2k_{45}k_{46} \oplus k_2k_{46}k_{47}$
$t_0t_1t_{48}$	$k_2 \oplus k_2k_{50} \oplus k_2k_{51} \oplus k_2k_{49}k_{50} \oplus k_2k_{50}k_{51}$
$t_0t_1t_{52}$	$k_2 \oplus k_2k_{54} \oplus k_2k_{55} \oplus k_2k_{53}k_{54} \oplus k_2k_{54}k_{55}$
$t_0t_1t_{56}$	$k_2 \oplus k_2k_{58} \oplus k_2k_{59} \oplus k_2k_{57}k_{58} \oplus k_2k_{58}k_{59}$

(Continues)

TABLE A2 List of cubes and the corresponding superpolys considering the second Hamming weight bit from the LSB after $\mathscr{R}_1$

Cube	Superpoly
t_1t_3	$k_{64} \oplus k_{66}$
t_5t_7	$k_{68} \oplus k_{70}$
t_9t_{11}	$k_{72} \oplus k_{74}$
$t_{13}t_{15}$	$k_{76} \oplus k_{78}$
$t_{17}t_{19}$	$k_{80} \oplus k_{82}$
$t_{21}t_{23}$	$k_{84} \oplus k_{86}$
$t_{25}t_{27}$	$k_{88} \oplus k_{90}$
$t_{29}t_{31}$	$k_{92} \oplus k_{94}$
$t_{33}t_{35}$	$k_{96} \oplus k_{98}$
$t_{37}t_{39}$	$k_{100} \oplus k_{102}$
$t_{41}t_{43}$	$k_{104} \oplus k_{106}$
$t_{45}t_{47}$	$k_{108} \oplus k_{110}$
$t_{49}t_{51}$	$k_{112} \oplus k_{114}$
$t_{53}t_{55}$	$k_{116} \oplus k_{118}$
$t_{57}t_{59}$	$k_{120} \oplus k_{122}$
$t_{61}t_{63}$	$k_{124} \oplus k_{126}$
$t_0t_1t_5$	$k_{66} \oplus k_{66}k_{68}k_{70}$
$t_0t_1t_9$	$k_{66} \oplus k_{66}k_{72}k_{74}$
$t_0t_1t_{13}$	$k_{66} \oplus k_{66}k_{76}k_{78}$
$t_0t_1t_{17}$	$k_{66} \oplus k_{66}k_{80}k_{82}$
$t_0t_1t_{21}$	$k_{66} \oplus k_{66}k_{84}k_{86}$
$t_0t_1t_{25}$	$k_{66} \oplus k_{66}k_{88}k_{90}$
$t_0t_1t_{29}$	$k_{66} \oplus k_{66}k_{92}k_{94}$
$t_0t_1t_{33}$	$k_{66} \oplus k_{66}k_{96}k_{98}$
$t_0t_1t_{37}$	$k_{66} \oplus k_{66}k_{100}k_{102}$
$t_0t_1t_{41}$	$k_{66} \oplus k_{66}k_{104}k_{106}$
$t_0t_1t_{45}$	$k_{66} \oplus k_{66}k_{108}k_{110}$
$t_0t_1t_{49}$	$k_{66} \oplus k_{66}k_{112}k_{114}$
$t_0t_1t_{53}$	$k_{66} \oplus k_{66}k_{116}k_{118}$
$t_0t_1t_{57}$	$k_{66} \oplus k_{66}k_{120}k_{122}$
$t_1t_4t_5$	$k_{70} \oplus k_{64}k_{66}k_{70}$
$t_4t_5t_9$	$k_{70} \oplus k_{70}k_{72}k_{74}$
$t_4t_5t_{13}$	$k_{70} \oplus k_{70}k_{76}k_{78}$
$t_4t_5t_{17}$	$k_{70} \oplus k_{70}k_{80}k_{82}$
$t_0t_1t_4$	$k_{66} \oplus k_{66}k_{70} \oplus k_{66}k_{71} \oplus k_{66}k_{69}k_{70} \oplus k_{66}k_{70}k_{71}$
$t_0t_1t_8$	$k_{66} \oplus k_{66}k_{74} \oplus k_{66}k_{75} \oplus k_{66}k_{73}k_{74} \oplus k_{66}k_{74}k_{75}$
$t_0t_1t_{12}$	$k_{66} \oplus k_{66}k_{78} \oplus k_{66}k_{79} \oplus k_{66}k_{77}k_{78} \oplus k_{66}k_{78}k_{79}$
$t_0t_1t_{16}$	$k_{66} \oplus k_{66}k_{82} \oplus k_{66}k_{83} \oplus k_{66}k_{81}k_{82} \oplus k_2k_{82}k_{83}$
$t_0t_1t_{20}$	$k_{66} \oplus k_{66}k_{86} \oplus k_{66}k_{87} \oplus k_{66}k_{85}k_{86} \oplus k_{66}k_{86}k_{87}$
$t_0t_1t_{24}$	$k_{66} \oplus k_{66}k_{90} \oplus k_{66}k_{91} \oplus k_{66}k_{89}k_{90} \oplus k_{66}k_{90}k_{91}$
$t_0t_1t_{28}$	$k_{66} \oplus k_{66}k_{94} \oplus k_{66}k_{95} \oplus k_{66}k_{93}k_{94} \oplus k_{66}k_{94}k_{95}$
$t_0t_1t_{32}$	$k_{66} \oplus k_{66}k_{98} \oplus k_{66}k_{99} \oplus k_{66}k_{97}k_{98} \oplus k_2k_{98}k_{99}$
$t_0t_1t_{36}$	$k_{66} \oplus k_{66}k_{102} \oplus k_{66}k_{103} \oplus k_{66}k_{101}k_{102} \oplus k_{66}k_{102}k_{103}$
$t_0t_1t_{40}$	$k_{66} \oplus k_{66}k_{106} \oplus k_{66}k_{107} \oplus k_{66}k_{105}k_{106} \oplus k_{66}k_{106}k_{107}$
$t_0t_1t_{44}$	$k_{66} \oplus k_{66}k_{110} \oplus k_{66}k_{111} \oplus k_{66}k_{109}k_{110} \oplus k_{66}k_{110}k_{111}$
$t_0t_1t_{48}$	$k_{66} \oplus k_{66}k_{114} \oplus k_{66}k_{115} \oplus k_{66}k_{113}k_{114} \oplus k_{66}k_{114}k_{115}$
$t_0t_1t_{52}$	$k_{66} \oplus k_{66}k_{118} \oplus k_{66}k_{119} \oplus k_{66}k_{117}k_{118} \oplus k_{66}k_{118}k_{119}$
$t_0t_1t_{56}$	$k_{66} \oplus k_{66}k_{122} \oplus k_{66}k_{123} \oplus k_{66}k_{121}k_{122} \oplus k_{66}k_{122}k_{123}$
$t_0t_1t_{60}$	$k_{66} \oplus k_{66}k_{126} \oplus k_{66}k_{127} \oplus k_{66}k_{125}k_{126} \oplus k_{66}k_{126}k_{127}$
$t_0t_4t_5$	$k_{70} \oplus k_{66}k_{70} \oplus k_{67}k_{70} \oplus k_{65}k_{66}k_{70} \oplus k_{66}k_{67}k_{70}$
$t_4t_5t_8$	$k_{70} \oplus k_{70}k_{74} \oplus k_{70}k_{75} \oplus k_{70}k_{73}k_{74} \oplus k_{70}k_{74}k_{75}$
$t_4t_5t_{12}$	$k_{70} \oplus k_{70}k_{78} \oplus k_{70}k_{79} \oplus k_{70}k_{77}k_{78} \oplus k_{70}k_{78}k_{79}$
$t_4t_5t_{16}$	$k_{70} \oplus k_{70}k_{82} \oplus k_{70}k_{83} \oplus k_{70}k_{81}k_{82} \oplus k_6k_{82}k_{83}$
$t_4t_5t_{20}$	$k_{70} \oplus k_{70}k_{86} \oplus k_{70}k_{87} \oplus k_{70}k_{85}k_{86} \oplus k_{70}k_{86}k_{87}$
$t_0t_2t_5$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{68}k_{70} \oplus k_{65}k_{68}k_{70} \oplus k_{67}k_{68}k_{70}$
$t_0t_2t_9$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{72}k_{74} \oplus k_{65}k_{72}k_{74} \oplus k_{67}k_{72}k_{74}$
$t_0t_2t_{13}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{76}k_{78} \oplus k_{65}k_{76}k_{78} \oplus k_{67}k_{76}k_{78}$
$t_0t_2t_{17}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{80}k_{82} \oplus k_{65}k_{80}k_{82} \oplus k_{67}k_{80}k_{82}$
$t_0t_2t_{21}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{84}k_{86} \oplus k_{65}k_{84}k_{86} \oplus k_{67}k_{84}k_{86}$
$t_0t_2t_{25}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{88}k_{90} \oplus k_{65}k_{88}k_{90} \oplus k_{67}k_{88}k_{90}$
$t_0t_2t_{29}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{92}k_{94} \oplus k_{65}k_{92}k_{94} \oplus k_{67}k_{92}k_{94}$
$t_0t_2t_{33}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{96}k_{98} \oplus k_{65}k_{96}k_{98} \oplus k_{67}k_{96}k_{98}$
$t_0t_2t_{37}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{100}k_{102} \oplus k_{65}k_{100}k_{102} \oplus k_{67}k_{100}k_{102}$
$t_0t_2t_{41}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{104}k_{106} \oplus k_{65}k_{104}k_{106} \oplus k_{67}k_{104}k_{106}$
$t_0t_2t_{45}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{108}k_{110} \oplus k_{65}k_{108}k_{110} \oplus k_{67}k_{108}k_{110}$
$t_0t_2t_{49}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{112}k_{114} \oplus k_{65}k_{112}k_{114} \oplus k_{67}k_{112}k_{114}$
$t_0t_2t_{53}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{116}k_{118} \oplus k_{65}k_{116}k_{118} \oplus k_{67}k_{116}k_{118}$
$t_0t_2t_{57}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{120}k_{122} \oplus k_{65}k_{120}k_{122} \oplus k_{67}k_{120}k_{122}$
$t_0t_2t_{61}$	$_1 \oplus k_{65} \oplus k_{67} \oplus k_{124}k_{126} \oplus k_{65}k_{124}k_{126} \oplus k_{67}k_{124}k_{126}$
$t_1t_4t_6$	$_1 \oplus k_{69} \oplus k_{71} \oplus k_{64}k_{66} \oplus k_{64}k_{66}k_{69} \oplus k_{64}k_{66}k_{71}$
$t_4t_6t_9$	$_1 \oplus k_{69} \oplus k_{71} \oplus k_{72}k_{74} \oplus k_{69}k_{72}k_{74} \oplus k_{71}k_{72}k_{74}$
$t_4t_6t_{13}$	$_1 \oplus k_{69} \oplus k_{71} \oplus k_{76}k_{78} \oplus k_{69}k_{76}k_{78} \oplus k_{71}k_{76}k_{78}$
$t_4t_6t_{17}$	$_1 \oplus k_{69} \oplus k_{71} \oplus k_{80}k_{82} \oplus k_{69}k_{80}k_{82} \oplus k_{71}k_{80}k_{82}$

(Continues)

APPENDIX B

CHARACTERISTICS OF STRONGER SECRET KEYS

TABLE B1 Characteristics of stronger secret keys, with $a = \sum_{i=0}^{15} (k_{4i} \oplus k_{4i+2})$, $b = \sum_{i=16}^{31} (k_{4i} \oplus k_{4i+2})$, $c = \sum_{i=0}^{15} (k_{4i+2})$ and $d = \sum_{i=16}^{31} (k_{4i+2})$

a	*b*	*c*	*d*	Number of combinations
0	0	0	0	$1 \cdot 2^{64} = 2^{64}$
0	0	0	1	$2^4 \cdot 2^{64} = 2^{68}$
0	0	0	≥2	$2^{16} \cdot 2^{64} = 2^{80}$
0	0	1	0	$2^4 \cdot 2^{64} = 2^{68}$
0	0	1	1	$2^4 \cdot 2^4 \cdot 2^{64} = 2^{72}$
0	0	1	≥2	$2^4 \cdot 2^{16} \cdot 2^{64} = 2^{84}$
0	0	≥2	0	$2^{16} \cdot 2^{64} = 2^{80}$
0	0	≥2	1	$2^{16} \cdot 2^4 \cdot 2^{64} = 2^{84}$
0	0	≥2	≥2	$2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{96}$
0	1	0	0	$2^4 \cdot 2^{64} = 2^{68}$
0	1	0	1	$2^4 \cdot 2^4 \cdot 2^{64} = 2^{72}$
0	1	0	≥2	$2^4 \cdot 2^{16} \cdot 2^{64} = 2^{84}$
0	1	1	0	$2^4 \cdot 2^4 \cdot 2^{64} = 2^{72}$
0	1	1	1	$2^4 \cdot 2^4 \cdot 2^4 \cdot 2^{64} = 2^{76}$
0	1	1	≥2	$2^4 \cdot 2^4 \cdot 2^{16} \cdot 2^{64} = 2^{88}$
0	1	≥2	0	$2^4 \cdot 2^{16} \cdot 2^{64} = 2^{84}$
0	1	≥2	1	$2^4 \cdot 2^{16} \cdot 2^4 \cdot 2^{64} = 2^{88}$
0	1	≥2	≥2	$2^4 \cdot 2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{100}$
0	≥2	0	0	$2^{16} \cdot 2^{64} = 2^{80}$
0	≥2	0	1	$2^{16} \cdot 2^4 \cdot 2^{64} = 2^{84}$
0	≥2	0	≥2	$2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{96}$
0	≥2	1	0	$2^{16} \cdot 2^4 \cdot 2^{64} = 2^{84}$
0	≥2	1	1	$2^{16} \cdot 2^4 \cdot 2^4 \cdot 2^{64} = 2^{88}$
0	≥2	1	≥2	$2^{16} \cdot 2^4 \cdot 2^{16} \cdot 2^{64} = 2^{100}$
0	≥2	≥2	0	$2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{96}$
0	≥2	≥2	1	$2^{16} \cdot 2^{16} \cdot 2^4 \cdot 2^{64} = 2^{100}$
0	≥2	≥2	≥2	$2^{16} \cdot 2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{112}$
1	0	0	0	$2^4 \cdot 2^{64} = 2^{68}$
1	0	0	1	$2^4 \cdot 2^4 \cdot 2^{64} = 2^{72}$
1	0	0	≥2	$2^4 \cdot 2^{16} \cdot 2^{64} = 2^{84}$
1	0	1	0	$2^4 \cdot 2^4 \cdot 2^{64} = 2^{72}$
1	0	1	1	$2^4 \cdot 2^4 \cdot 2^4 \cdot 2^{64} = 2^{76}$
1	0	1	≥2	$2^4 \cdot 2^4 \cdot 2^{16} \cdot 2^{64} = 2^{88}$
1	0	≥2	0	$2^4 \cdot 2^{16} \cdot 2^{64} = 2^{84}$
1	0	≥2	1	$2^4 \cdot 2^{16} \cdot 2^4 \cdot 2^{64} = 2^{88}$
1	0	≥2	≥2	$2^4 \cdot 2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{100}$
1	1	0	0	$2^4 \cdot 2^4 \cdot 2^{64} = 2^{72}$
1	1	0	1	$2^4 \cdot 2^4 \cdot 2^4 \cdot 2^{64} = 2^{76}$
1	1	0	≥2	$2^4 \cdot 2^4 \cdot 2^{16} \cdot 2^{64} = 2^{88}$
1	1	1	0	$2^4 \cdot 2^4 \cdot 2^4 \cdot 2^{64} = 2^{76}$
1	1	1	1	$2^4 \cdot 2^4 \cdot 2^4 \cdot 2^4 \cdot 2^{64} = 2^{80}$

(Continues)

TABLE B1 (Continued)

a	*b*	*c*	*d*	Number of combinations
1	1	1	≥2	$2^4 \cdot 2^4 \cdot 2^4 \cdot 2^{16} \cdot 2^{64} = 2^{92}$
1	1	≥2	0	$2^4 \cdot 2^4 \cdot 2^{16} \cdot 2^{64} = 2^{88}$
1	1	≥2	1	$2^4 \cdot 2^4 \cdot 2^{16} \cdot 2^4 \cdot 2^{64} = 2^{92}$
1	1	≥2	≥2	$2^4 \cdot 2^4 \cdot 2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{104}$
1	≥2	0	0	$2^4 \cdot 2^{16} \cdot 2^{64} = 2^{84}$
1	≥2	0	1	$2^4 \cdot 2^{16} \cdot 2^4 \cdot 2^{64} = 2^{88}$
1	≥2	0	≥2	$2^4 \cdot 2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{100}$
1	≥2	1	0	$2^4 \cdot 2^{16} \cdot 2^4 \cdot 2^{64} = 2^{88}$
1	≥2	1	1	$2^4 \cdot 2^{16} \cdot 2^4 \cdot 2^4 \cdot 2^{64} = 2^{92}$
1	≥2	1	≥2	$2^4 \cdot 2^{16} \cdot 2^4 \cdot 2^{16} \cdot 2^{64} = 2^{104}$
1	≥2	≥2	0	$2^4 \cdot 2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{100}$
1	≥2	≥2	1	$2^4 \cdot 2^{16} \cdot 2^{16} \cdot 2^4 \cdot 2^{64} = 2^{104}$
1	≥2	≥2	≥2	$2^4 \cdot 2^{16} \cdot 2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{116}$
≥2	0	0	0	$2^{16} \cdot 2^{64} = 2^{80}$
≥2	0	0	1	$2^{16} \cdot 2^4 \cdot 2^{64} = 2^{84}$
≥2	0	0	≥2	$2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{96}$
≥2	0	1	0	$2^{16} \cdot 2^4 \cdot 2^{64} = 2^{84}$
≥2	0	1	1	$2^{16} \cdot 2^4 \cdot 2^4 \cdot 2^{64} = 2^{88}$
≥2	0	1	≥2	$2^{16} \cdot 2^4 \cdot 2^{16} \cdot 2^{64} = 2^{100}$
≥2	0	≥2	0	$2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{96}$
≥2	0	≥2	1	$2^{16} \cdot 2^{16} \cdot 2^4 \cdot 2^{64} = 2^{100}$
≥2	0	≥2	≥2	$2^{16} \cdot 2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{112}$
≥2	1	0	0	$2^{16} \cdot 2^4 \cdot 2^{64} = 2^{84}$
≥2	1	0	1	$2^{16} \cdot 2^4 \cdot 2^4 \cdot 2^{64} = 2^{88}$
≥2	1	0	≥2	$2^{16} \cdot 2^4 \cdot 2^{16} \cdot 2^{64} = 2^{100}$
≥2	1	1	0	$2^{16} \cdot 2^4 \cdot 2^4 \cdot 2^{64} = 2^{88}$
≥2	1	1	1	$2^{16} \cdot 2^4 \cdot 2^4 \cdot 2^4 \cdot 2^{64} = 2^{92}$
≥2	1	1	≥2	$2^{16} \cdot 2^4 \cdot 2^4 \cdot 2^{16} \cdot 2^{64} = 2^{104}$
≥2	1	≥2	0	$2^{16} \cdot 2^4 \cdot 2^{16} \cdot 2^{64} = 2^{100}$
≥2	1	≥2	1	$2^{16} \cdot 2^4 \cdot 2^{16} \cdot 2^4 \cdot 2^{64} = 2^{104}$
≥2	1	≥2	≥2	$2^{16} \cdot 2^4 \cdot 2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{116}$
≥2	≥2	0	0	$2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{96}$
≥2	≥2	0	1	$2^{16} \cdot 2^{16} \cdot 2^4 \cdot 2^{@64} = 2^{@100}$
≥2	≥2	0	≥2	$2^{16} \cdot 2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{112}$
≥2	≥2	1	0	$2^{16} \cdot 2^{16} \cdot 2^4 \cdot 2^{64} = 2^{100}$
≥2	≥2	1	1	$2^{16} \cdot 2^{16} \cdot 2^4 \cdot 2^4 \cdot 2^{64} = 2^{104}$
≥2	≥2	1	≥2	$2^{16} \cdot 2^{16} \cdot 2^4 \cdot 2^{16} \cdot 2^{64} = 2^{116}$
≥2	≥2	≥2	0	$2^{16} \cdot 2^{16} \cdot 2^{16} \cdot 2^{64} = 2^{112}$
≥2	≥2	≥2	1	$2^{16} \cdot 2^{16} \cdot 2^{16} \cdot 2^4 \cdot 2^{64} = 2^{116}$