New Differential Cryptanalysis Results for the Lightweight Block Cipher BORON

Je Sen Teh[a,b,*], Li Jing Tham[a], Norziana Jamil[c], Wun-She Yap[d,*]

[a]School of Computer Sciences, Universiti Sains Malaysia, 11800 Gelugor, Malaysia
[b]Interdisciplinary Centre for Security, Reliability and Trust (SnT), Universiti of Luxembourg, 4365 Esch-sur-Alzette, Luxembourg
[c]Department of Computing, College of Computing and Informatics, Universiti Tenaga Nasional, 43000 Kajang, Malaysia
[d]Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, 43000 Kajang, Malaysia

Abstract

BORON is a 64-bit lightweight block cipher based on the substitution-permutation network that supports an 80-bit (BORON-80) and 128-bit (BORON-128) secret key. In this paper, we revisit the use of differential cryptanalysis on BORON in the single-key model. Using an SAT/SMT approach, we look for differentials that consist of multiple differential characteristics with the same input and output differences. Each characteristic that conforms to a given differential improves its overall probability. We also implemented the same search using Matsui's algorithm for verification and performance comparison purposes. We identified high-probability differentials which were then used in key recovery attacks against BORON-80/128. We used 8-round differentials with a probability of $2^{-58.16}$ and $2^{-62.42}$ in key recovery attacks against 9 and 10 rounds of BORON-80 and BORON-128 with time/data/memory complexities of $2^{59.18}/2^{59.16}/2^{24}$ and $2^{111.34}/2^{63.42}/2^{71}$ respectively. Our key recovery framework provides a more accurate estimate of the attack complexity as compared to previous work. The attacks proposed in this paper are the best differential attacks against BORON-80/128 in the single-key model to date.

Keywords:
BORON, cryptanalysis, differential cryptanalysis, branch-and-bound, SAT solver, SMT solver

1. Introduction

Block ciphers are ubiquitous cryptographic primitives found in protocols such as Transport Layer Security (TLS) [1], OpenPGP [2], and SSH Transport Layer Protocol [3]. They are symmetric-key enchttps://www.overleaf.com/project/613cb14c46a7fb3b9ba5875lryption algorithms that use one secret key, K to perform both encryption, $E_K(X) = Y$ and decryption, $D_K(Y) = X$, where X and Y represent the plaintext (message) and ciphertext (encrypted message) respectively. Block ciphers can be used as building blocks to construct random number generators (using the CTR mode of operation), message authentication codes [4, 5] and authenticated encryption algorithms [6, 7]. Block ciphers are also expected to be secure in a post-quantum world, with just the selection of larger keys or block sizes [8].

Recently, the design and analysis of lightweight cryptographic algorithms have gained popularity among researchers, which includes the ongoing standardization effort for lightweight cryptography by the National Institute of Standards and Technology (NIST)[9]. These lightweight algorithms are designed to have reduced power and memory consumption to fulfill the requirements of resource-constrained applications such as RFID tags and Internet of Things (IoT) devices [10]. Lightweight block ciphers commonly have block sizes of 32 to 64 bits with key sizes of 80 or 128 bits. Some well-studied lightweight block ciphers include PRESENT [11], TWINE [12], LBlock [13], SIMON/Speck [14] and KATAN/KTANTAN [15]. The block cipher analyzed in this paper, BORON, is an example of a recently proposed 64-bit lightweight block cipher [16].

A block cipher is considered secure enough for practical use after it has undergone extensive third-party cryptanalysis. One of the main techniques used to cryptanalyze block ciphers is differential cryptanalysis [17]. Resistance against differential cryptanalysis is widely considered a mandatory design criterion for any block cipher. A successful differential attack relies on finding a highly probable differential characteristic (propagation of an input difference through the cipher to produce a corresponding output difference) that will be used as a statistical distinguisher for key recovery. Identifying these differential characteristics is a time-consuming and highly technical task, which has been greatly simplified with the availability of search algorithms such as Matsui's branch-and-bound search, mixed-integer linear programming (MILP) and Boolean satisfiability (SMT/SAT) approaches.

One of the earliest automated differential search tools was introduced by Matsui, whose branch-and-bound approach was used to find differential and linear characteristics for DES [18]. The search algorithm was later modified by other researchers to target other block ciphers, incorporate other types of bounding requirements, time-data complexity trade-offs, and parallel processing [19, 20, 21, 22]. The use of Matsui's algorithm requires sophisticated programming to adapt it to different ciphers. Later, Mouha et al. used MILP to perform differential

*Corresponding author
Email addresses: jesen_teh@usm.my (Je Sen Teh), yapws@utar.edu.my (Wun-She Yap)

and linear cryptanalysis on a stream cipher, Encoro-128v2 and also count the number of active S-boxes for AES [23]. The MILP approach was further enhanced by Sun et. al to automatically enumerate differential or linear characteristics to construct differentials or linear hulls [24]. Since then, MILP has been widely used to aid differential cryptanalysis efforts of block ciphers such as GIFT [25], CRAFT [26], PRESENT, RECTANGLE, LBlock and TWINE [27]. The general idea of the MILP approach is to model a cipher as a series of linear inequalities that involve (integer) decision variables that describe the various operations of the cipher. Then, a mathematical programming solver such as the IBM ILOG CPLEX Optimizer can be used to solve the MILP model. The use and implementation of the MILP approach are less complex as compared to Matsui's algorithm.

An alternative approach that models the differential search problem as a Boolean satisfiability problem (SAT) was used by Mouha and Preneel in their efforts to find optimal differential characteristics for addition-rotation-XOR (ARX) ciphers [28]. Their SAT approach involved writing Boolean equations that represent the operations of a cipher as well as the objective function (number of active S-boxes or differential probability). An SAT or satisfiability modulo theories (SMT) solver such as CryptoMiniSat [29] or STP is then invoked to find a solution. Ankele and Kölbl used the SAT/SMT approach to analyze the security of various block ciphers (LBlock, Midori, PRESENT, Prince, Rectangle, Simon, Skinny Sparx, Speck, TWINE) against differential cryptanalysis, taking into consideration the notion of differentials rather than single characteristics to provide accurate security bounds [30]. More recently, Sun et al. enhanced the standard SAT method by incorporating Matsui's bounding criteria as additional Boolean constraints, which greatly accelerated the search for optimal differential characteristics [31]. They were not only able to provide the complete bounds for various lightweight block ciphers, but also provided empirical evidence that their SAT approach outperforms MILP. The previous cryptanalysis attempt against BORON also utilized an SMT solver [32].

1.1. Contribution

In this paper, we take an in-depth look at BORON's resistance against differential cryptanalysis in the single-key model, using the SMT method to automatically enumerate differentials rather than just relying on a single optimal characteristic. To date, there has only been one prior cryptanalysis attempt [32]. We are interested to further cryptanalyze BORON due to its unique design which supposedly maximizes the number of active S-boxes in fewer rounds as compared to a regular bitwise permutation. By having more cryptanalysis results, its security strength and potential for real-life application would be better understood.

Finding BORON differentials. We investigate BORON's security against differential cryptanalysis by considering the notion of *differentials* rather than a single differential *characteristic*. We search for high-probability differentials based on differential characteristics identified using an SMT model. By including additional constraints to the SMT model, we restrict the search to a specific input and output difference before adding more constraints to block solutions (characteristics) that have already been found. This allowed us to find the high-probability differentials for BORON. A detailed look at several of these differentials is provided in Section 3.4.

Comparison of differential search approaches. We implement the same differential search using a variant of Matsui's branch-and-bound search [20] for performance comparison purposes. Without considering parallelization, the SMT approach was found to be more efficient than Matsui's. Notably, finding a differential for 9 rounds of BORON required less than half an hour while Matsui's algorithm required over two days.

New key recovery attacks. We propose new key recovery attacks against round-reduced BORON-80 and BORON-128 with more accurate estimates of the attack complexity in Section 4. By using an improved 8-round differential with a probability of $2^{-58.156}$, our best attack against 9 rounds of BORON-80 has a time/memory/data complexity of $2^{59.18}/2^{59.16}/2^{24}$. We show that the previous attack proposed in [32] actually has an attack complexity of approximately $2^{62.94}$ when the data preparation phase is taken into consideration. We also attack 10 rounds of BORON-128 using a different 8-round differential with a probability of $2^{-62.415}$ to achieve a time/memory/ data complexity of $2^{111.34}/2^{63.42}/2^{71}$. A summary of differential cryptanalysis results for BORON in the single-key model is provided in Table 1.

Software The SMT model used to find differentials is publicly available at `https://github.com/jesenteh/boron-smt`.

1.2. Outline

Section 2 first introduces BORON, differential cryptanalysis, and automated differential search approaches. Then in Section 3, we describe our differential search strategy based on SMT solvers and provide the high-probability differentials corresponding to various rounds of BORON. A performance comparison between SMT and Matsui approaches is also provided in the same section. Finally, Section 4 details the key recovery attacks against BORON-80/128. We conclude the paper in Section 5 with some final remarks and future work.

2. Preliminaries

2.1. Notations

In this paper, we adopt the convention of numbering the rightmost bits or blocks as 0, and increment the index as we move from right to left. Also, plaintext and ciphertext differences are represented as hexadecimal values. The following notations and symbols are used throughout the paper:

1. $\oplus$: Binary exclusive OR (XOR) operation
2. $x <<< i$: Left rotation (circular shift) of x by i bits
3. $S_i[x]$: i-th 4-bit S-box with a 4-bit input, x

Table 1: Differential cryptanalysis results for BORON in the single-key model

Reference	Rounds	Version	Time	Data	Memory
[32]	9	BORON-80	$2^{62.94}$	2^{63}	2^{24}
Section 4.2	9	BORON-80	$2^{59.18}$	$2^{59.16}$	2^{24}
Section 4.3	10	BORON-128	$2^{111.34}$	$2^{63.415}$	2^{71}

4. 80/128-bit secret key, $K = \{k_{79/127}, k_{78/126}, ..., k_1, k_0\}$
5. 64-bit plaintext, $X = \{x_{63}, x_{62}, ..., x_1, x_0\}$
6. 64-bit ciphertext, $Y = \{y_{63}, y_{62}, ..., y_1, y_0\}$
7. 64-bit r-th round key, $RK_r = \{rk_{r,63}, rk_{r,62}, ..., rk_{r,1}, rk_{r,0}\}$

2.2. BORON Revisited

BORON is a 64-bit lightweight block cipher with 25 rounds, based on the substitution-permutation network (SPN) proposed by Bansod et al. [16]. It supports key sizes of 80 and 128 bits, which we will refer to as BORON-80 and BORON-128 respectively. It consists of a key addition (XOR) layer, substitution layer of 16 S-boxes, and a linear layer that consists of a block shuffle, bitwise rotation, and block XOR operations. The four blocks involved in the linear layer are 16 bits each. The designers claimed that BORON's linear layer activates many S-boxes in fewer rounds as compared to its peers apart from having low area and power requirements, and high throughput. These properties would make BORON a great alternative to existing lightweight block ciphers if more third-party cryptanalysis work was available to analyze its security. One round of BORON is shown in Figure 1. The substitution layer uses the same S-box depicted in Table 2 in hexadecimal format.

The key schedule generates a total of 26 64-bit round keys, RK_r where $r = \{0, 2, ..., 25\}$. Taking RK_0 as the whitening key, the remaining keys are used for each of the remaining rounds. Each round, the 64-bit round key will be XOR-ed with the entire 64-bit block of data before proceeding to the substitution and linear layer. The 128-bit key is as follows:

1. RK_r=$\{k_{r,63}, rk_{r,62}, ..., rk_{r,1}, rk_{r,0}\}$
2. RK_r <<< 13
3. $[rk_{r,3}\ rk_{r,2}\ rk_{r,1}\ rk_{r,0}] \leftarrow S[rk_{r,3}\ rk_{r,2}\ rk_{r,1}\ rk_{r,0}]$
4. $[rk_{r,7}\ rk_{r,6}\ rk_{r,5}\ rk_{r,4}] \leftarrow S[rk_{r,7}\ rk_{r,6}\ rk_{r,5}\ rk_{r,4}]$
5. $[rk_{r,63}\ rk_{r,62}\ rk_{r,61}\ rk_{r,60}\ rk_{r,59}] \leftarrow [rk_{r,63}\ rk_{r,62}\ rk_{r,61}\ rk_{r,60}\ rk_{r,59}] \oplus RC_i$,

where RC_i is a round counter. The 80-bit key schedule is exactly the same as the 128-bit key schedule with Step 4 omitted.

2.3. Differential Cryptanalysis and Automated Differential Search

The goal of an attacker when using differential cryptanalysis is to identify a differential or differential characteristic that holds with sufficiently high probability, which is then used as a statistical distinguisher to recover key bits. An r-round *differential* denoted by

$$\Delta X \xrightarrow{r} \Delta Y \tag{1}$$

has an input difference, $\Delta X = X' \oplus X''$ and corresponding output difference, $\Delta Y = Y' \oplus Y''$, where (X', X'') and (Y', Y'') are known as plaintext and ciphertext pairs respectively. The r-round propagation of ΔX to ΔY can be represented as a sequence of intermediate differences δx_r,

$$\Delta X \rightarrow \delta x_1 \rightarrow \delta x_2 \rightarrow ... \rightarrow \delta x_{r-2} \rightarrow \delta x_{r-1} \rightarrow \Delta Y. \tag{2}$$

Each sequence corresponding to a unique set of intermediate differences is known as a *differential characteristic*. Thus, a *differential* can consist of more than one *differential characteristic* that has the same endpoints $\Delta X, \Delta Y$ but differs in terms of their intermediate differences.

The difference propagation of a differential characteristic is probabilistic due to the interactions between the round subkeys and nonlinear layers (such as the substitution layer). For block ciphers that use S-boxes, a probability penalty of 2^{-p} is incurred for each active S-box (S-boxes that receive nonzero differences as inputs). The value of p depends on the differential distribution table (DDT) of the S-box. For more details about how to derive the DDT for an S-box, readers can refer to [17]. The product of these probability penalties results in the probability of a single characteristic. The overall *differential probability*, $Pr(\Delta X \xrightarrow{r} \Delta Y)$ can be obtained by summing up the individual probabilities of each characteristic that forms the differential. This concept of differentials was first put forward by Lai et al. based on the assumption that an iterated block cipher is Markov, whose round subkeys are independent, and the output differences of each round form a Markov chain [33].

2.3.1. The SAT/SMT Approach

To use an SAT or SMT solver to search for differential characteristics or differentials, a cryptanalyst needs to develop an SAT or SMT model of the cipher. This involves representing all possible intermediate states of each round as variables, then use them to form constraints that describe the differential behavior of the cipher. The objective functions (target differential probability or the number of active S-boxes) also need to be encoded as constraints. Then, a solver is invoked to check for the satisfiability of these constraints. SMT solvers are considered more powerful and are easier to use for block ciphers because they support both bitwise and word-based operations. An SMT solver first encodes the constraints using languages such as CVC or SMTLIB2 before invoking an underlying SAT solver to check for satisfiability. We utilize the SMT solver known as STP for our cryptanalysis task [34].

2.3.2. Matsui's Branch-and-Bound Algorithm

Matsui's branch-and-bound algorithm can be used to identify the best differential or linear characteristics for a block cipher. Generally, Matsui's algorithm traverses all possible differential trails for a cipher, then bounds (cuts off) trails that are

Table 2: 4-bit BORON S-box

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	E	4	B	1	7	9	C	A	D	2	0	F	8	5	3	6

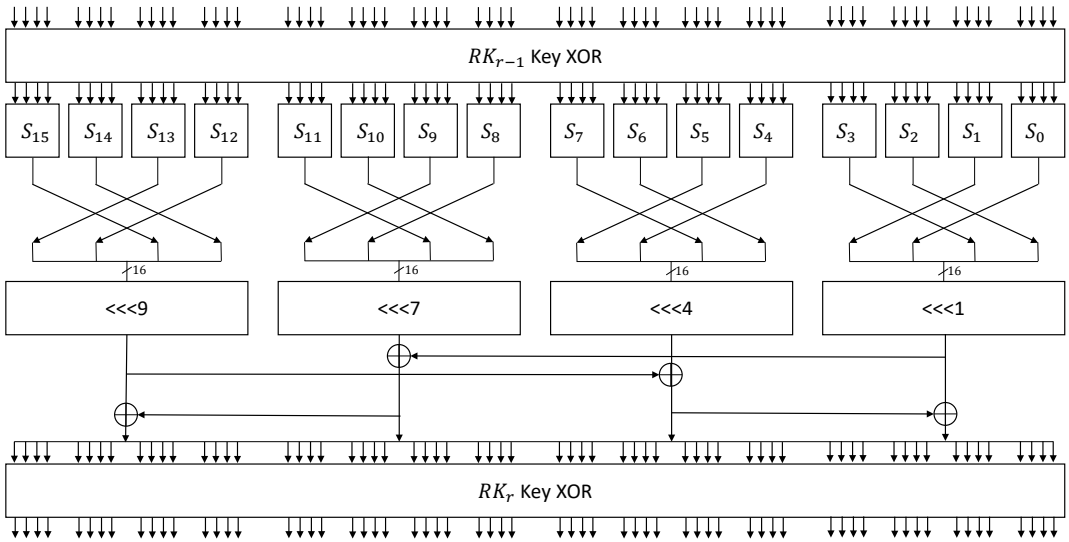

Figure 1: The round function of BORON.

unlikely to lead to an improved differential probability. This is performed by bounding trails that have probabilities less than $\overline{B_n}$, which is the best probability that has been found so far. $\overline{B_n}$ is constantly updated during the algorithm execution to further limit the differential search space. In this paper, we compare our SMT-based approach to a variant of the Matsui algorithm proposed by Chen et al., which was proposed to efficiently enumerate all characteristics corresponding to a given differential [20].

3. New Differentials for BORON

3.1. Finding the Best Differential Characteristic

The same SMT model described in [32] (available at `https://github.com/CatherineLiang/Cryptanalysis-of-BORON`) was used to find the single best differential characteristics. Variables were created to represent intermediate states, which include the inputs and outputs of every encryption operation (substitution, block shuffle, rotation, and block XOR). For the convenience of the reader, we provide a summary of all the variables relevant to the differential search:

1. $before\_sbox\_value1\_i\_j$: Input value of the j-th S-box of the i-th round for the first input of the difference pair
2. $after\_sbox\_value1\_i\_j$: output value of the j-th S-box of the i-th round whose input value is $before\_sbox\_value1\_i\_j$
3. $before\_sbox\_value2\_i\_j$: Input value of the j-th S-box of the i-th round for the second input of the difference pair
4. $after\_sbox\_value2\_i\_j$: output value of the j-th S-box of the i-th round whose input value is $before\_sbox\_value1\_i\_j$
5. $before\_sbox\_difference\_i\_j = before\_sbox\_value1\_i\_j \oplus before\_sbox\_value2\_i\_j$
6. $after\_sbox\_difference\_i\_j = after\_sbox\_value1\_i\_j \oplus after\_sbox\_value2\_i\_j$
7. $before\_rotation\_difference\_i\_i$: Difference of the j-th block before performing bit rotation in the i-th round
8. $after\_rotation\_difference\_i\_i$: Difference of the j-th block after performing bit rotation in the i-th round
9. $after\_blockxor\_difference\_i\_j$: Difference of the j-th block after performing the block XOR operation
10. $probability\_i\_j$: Probability parameter of the j-th S-box of the i-th round, set based on BORON's DDT
11. $total\_probability$: Sum of all $probability\_i\_j$ to obtain the *weight* of the differential characteristic (objective function)

Using the SMT model is a straightforward process - Set the number of rounds to search for and target weight ($total\_probability$), then run a script that prints all the constraints in a format (CVC) that can be understood by the SMT solver. The weight of a differential is defined as

$$total\_probability = -\log_2(Pr(\Delta X \xrightarrow{r} \Delta Y)). \tag{3}$$

To identify the best differential characteristic for a particular round, an attacker will need to try all possible weights starting from a initial value, and increment the weight until the constraints are satisfiable. If the optimal number of active S-boxes is known, it can be used to derive the initial weight value for the search (2× the number of active S-box). More details about the basic SMT model can be found in [32].

3.2. Finding the Best Differential

To find the best differential, we include additional constraints to the SMT model to fix the input and output differences (which is straightforward to encode), and exclude differential characteristics that have already been found. When excluding differential characteristics that have already been found, we need to ensure that only characteristics that have the same set of intermediate differences (Eq. 2) will be excluded. If one of the intermediate differences is distinct, a new differential characteristic has been found.

To exclude characteristics that have already been found, we used a *sum of modulo* approach. For each difference going into the S-box, we perform a modulo operation with intermediate differences that have been found so far. At the end of the search, the results of all modulo operations (corresponding to a particular differential characteristic) are summed up. This is performed for all of the differential characteristics that have been found so far. If all sums of modulo values are larger than 1, a new differential characteristic has been found. The intermediate differences for this new differential characteristic will be encoded into the SMT model and the search is executed again. This is repeated until the constraints are no longer satisfiable. New variables that need to be encoded are as follows:

1. *inter_c_i_j*: Parameter to store the intermediate difference for the *j*-th S-box of the *i*-th round of the *c*-th differential characteristic
2. *mod_c_i_j*: Parameter to store the result of all modulo operations
3. *total_mod_c*: Parameter to store the sum of modulo values for the *c*-th characteristic

The pseudocode for finding differential characteristics corresponding to a given differential is detailed in Algorithm 1.

3.3. General Differential Search Strategy using SMT Solvers

To enumerate high probability differentials for BORON, the following steps were performed:

1. Run the SMT solver (STP) to identify the single best differential characteristic for round *r* (Sec. 3.1).
2. Encode the differential characteristic as the starting points of the differential search
3. Encode the intermediate differences of the characteristic into the differential search (Sec. 3.2)
4. Find a new differential characteristic corresponding to the initial differences specified in Step 2.
5. Repeat steps 3 and 4 until no more characteristics can be found.
6. Increment the target weight by 1.
7. Repeat steps 3 to 6 until the desired differential probability has been achieved.

3.4. Performance Comparison and Verification of Correctness

All experiments were performed on a computer with an Intel Core i7-9700K 3.60GHz CPU and 32GB RAM. For the best differential characteristic with a weight of w_{best}, we find all other characteristics up to w_{best} + 5 for verification purposes. Several examples of differentials found using the proposed approach are illustrated in Table 3. Three of the best characteristics from each differential (with the variations in intermediate differences highlighted in bold) are shown in Tables 4, 5, 6, and 7. We were unable to find a valid 9-round differential even when increasing the search space to w_{best}+20. 9-round differential characteristics with probabilities ranging between 2^{-82} and 2^{-90} were found but barely contribute to improving the overall differential probability. This finding supports BORON's designer's claims that its linear layer is effective in maximizing the number of active S-boxes in fewer rounds as compared to a conventional bitwise permutation.

To verify the correctness of the differentials that were found, we implemented the same search using Matsui's algorithm. We adapted an enhanced version from [20] to BORON's structure, which includes a meet-in-the-middle time-memory trade-off to optimize efficiency. To further speed up the search, we reduce the number of operations required by simplifying BORON's linear layer. This was performed by combining the block shuffle and bitwise rotation operations into a single bitwise permutation. The simplification of the linear layer is as shown in Figure 2. We compare both Matsui and SMT approaches in terms of computational time required to find the same 9-round differential described in Table 3. Matsui's algorithm and the SMT approach required approximately 186243.599s and 1589.071s respectively to find all 8 characteristics of the target differential. The SMT approach was found to be faster than Matsui's algorithm by a factor of approximately 117.2x when applied to 9 rounds of BORON.

4. New Key Recovery Attacks on BORON

In this section, we provide a detailed key recovery framework with a more accurate estimate of the attack complexity. We first take a quick look at the previous attack on BORON before describing our improved attacks on 9 rounds of BORON-80 and 10-rounds of BORON-128 using differentials.

4.1. Previous 9-round Attack on BORON-80

In [32], the authors described an attack on BORON-80 using an 8-round differential characteristic, 0000 0800 0010 0000 $\rightarrow$ 0041 0041 0008 0009 that holds with a probability of 2^{-62}. They omit the final linear layer of the 9th round and recover 24 bits of RK_9 corresponding to the active S-boxes of their output difference, including $rk_{9,0}$, $rk_{9,1}$, $rk_{9,2}$, $rk_{9,3}$, $rk_{9,16}$, $rk_{9,17}$, $rk_{9,18}$, $rk_{9,19}$, $rk_{9,32}$, $rk_{9,33}$, $rk_{9,34}$, $rk_{9,35}$, $rk_{9,36}$, $rk_{9,37}$, $rk_{9,38}$, $rk_{9,39}$, $rk_{9,48}$, $rk_{9,49}$, $rk_{9,50}$, $rk_{9,51}$, $rk_{9,52}$, $rk_{9,53}$, $rk_{9,54}$, and $rk_{9,55}$. If the final linear layer is taken into consideration, just guessing these 24 bits will not be sufficient to recover the expected output difference of their distinguisher due to the bitwise mixing that occurs due to the block XOR operation. In fact, a total of 55 bits are actually involved in the recovery of the expected output difference as shown in Figure 3. However, the 24-bit key is still an equivalent key that provides 24 bits of key information. We will adopt a similar assumption in our 9-round attack

Table 3: Best Differentials for BORON limited to a weight of $w_{best} + 5$

Rounds	Input Difference	Output Difference	Probability	Number of Characteristics
7	00 0b 00 0a 06 30 00 00	09 03 3d 02 34 c1 34 c3	$2^{-50.42}$	2
8	00 00 30 00 00 0b 00 06	a0 00 a0 00 00 00 a0 00	$2^{-62.415}$	8
8	00 00 00 08 a0 00 00 00	8e 03 8e 03 70 00 7e 00	$2^{-58.74}$	36
9	00 00 30 00 00 0B 00 06	A1 E0 21 E0 80 00 81 E0	$2^{-68.42}$	8

Table 4: Differential characteristics from the 7-round $2^{-50.42}$ differential

Round	Characteristic 1	Characteristic 2
0	00 0b 00 0a 06 30 00 00	00 0b 00 0a 06 30 00 00
1	00 00 00 02 00 **e0** 00 **e0**	00 00 00 02 00 **c0** 00 **c0**
2	80 00 80 00 00 01 00 00	80 00 80 00 00 01 00 00
3	00 00 60 00 00 00 00 00	00 00 60 00 00 00 00 00
4	50 00 50 00 00 00 00 00	50 00 50 00 00 00 00 00
5	00 00 60 00 60 00 60 00	00 00 60 00 60 00 60 00
6	41 00 41 00 08 00 09 00	41 00 41 00 08 00 09 00
7	09 03 3d 02 34 c1 34 c3	09 03 3d 02 34 c1 34 c3
Probability	2^{-51}	2^{-52}

Table 5: Differential characteristics from the 8-round $2^{-62.415}$ differential

Round	Characteristic 1	Characteristic 2	Characteristic 3
0	00 00 30 00 00 0b 00 06	00 00 30 00 00 0b 00 06	00 00 30 00 00 0b 00 06
1	00 00 00 00 10 00 00 00	00 00 00 00 10 00 00 00	00 00 00 00 10 00 00 00
2	00 00 00 00 **0a** 00 **0a** 00	00 00 00 00 **06** 00 **06** 00	00 00 00 00 **0e** 00 **0e** 00
3	00 08 00 08 00 e0 00 e8	00 08 00 08 00 e0 00 e8	00 08 00 08 00 70 00 78
4	0c 00 0c 06 00 00 0c 01	0c 00 0c 06 00 00 0c 01	0c 00 0c 06 00 00 0c 01
5	00 80 0c 80 0c 00 00 02	00 80 0c 80 0c 00 00 02	00 80 0c 80 0c 00 00 02
6	00 00 00 60 00 00 06 00	00 00 00 60 00 00 06 00	00 00 00 60 00 00 06 00
7	00 00 00 00 00 00 00 10	00 00 00 00 00 00 00 10	00 00 00 00 00 00 00 10
8	a0 00 a0 00 00 00 a0 00	a0 00 a0 00 00 00 a0 00	a0 00 a0 00 00 00 a0 00
Probability	2^{-64}	2^{-65}	2^{-66}

Table 6: Differential characteristics from the 8-round $2^{-58.74}$ differential

Round	Characteristic 1	Characteristic 2	Characteristic 3
0	00 00 00 08 a0 00 00 00	00 00 00 08 a0 00 00 00	00 00 00 08 a0 00 00 00
1	00 06 00 06 0f 00 0f 00	00 06 00 06 0f 00 0f 00	00 06 00 06 0f 00 0f 00
2	00 **07** 00 **17** 00 **60** 00 **70**	00 **07** 00 **17** 00 **f0** 00 **e0**	00 **04** 00 **14** 00 **f0** 00 **e0**
3	00 70 00 78 00 00 80 00	00 70 00 78 00 00 80 00	00 70 00 7e 00 00 80 00
4	00 0e 00 ce 00 c0 00 00	00 0e 00 ce 00 c0 00 00	00 0c 00 cc 00 c0 00 00
5	00 00 00 0c 00 00 00 00	00 00 00 0c 00 00 00 00	00 00 00 0c 00 00 00 00
6	00 06 00 06 00 00 00 00	00 06 00 06 00 00 00 00	00 06 00 06 00 00 00 00
7	00 00 00 04 00 04 00 04	00 00 00 04 00 04 00 04	00 00 00 04 00 04 00 04
8	8e 03 8e 03 70 00 7e 00	8e 03 8e 03 70 00 7e 00	8e 03 8e 03 70 00 7e 00
Probability	2^{-62}	2^{-62}	2^{-62}

Algorithm 1 Finding differential characteristics with corresponding to $\Delta X \xrightarrow{r} \Delta Y$.

r: Number of rounds
ch: Number of characteristics being excluded
for $c \leftarrow 0$ to $ch - 1$ **do**
    for $i \leftarrow 0$ to $r - 1$ **do**
        for $j \leftarrow 0$ to 15 **do**
            $mod\_c\_i\_j \leftarrow before\_sbox\_difference\_i\_j \oplus inter\_c\_i\_j$
        end for
    end for
end for
for $c \leftarrow 0$ to ch-1 **do**
    for $i \leftarrow 0$ to $r - 1$ **do**
        for $j \leftarrow 0$ to 15 **do**
            $total\_mod\_c \leftarrow total\_mod\_c + mod\_c\_i\_j$
        end for
    end for
end for
if $total\_mod\_c = 0$ for all values of c **then**
    No new differential characteristic has been found
else
    Output solution as a new differential characteristic
end if

Table 7: Differential characteristics from the 9-round $2^{-68.42}$ differential

Round	Characteristic 1	Characteristic 2	Characteristic 3
0	00 00 30 00 00 0b 00 06	00 00 30 00 00 0b 00 06	00 00 30 00 00 0b 00 06
1	00 00 00 00 10 00 00 00	00 00 00 00 10 00 00 00	00 00 00 00 10 00 00 00
2	00 00 00 00 **0a** 00 **0a** 00	00 00 00 00 **06** 00 **06** 00	00 00 00 00 **0e** 00 **0e** 00
3	00 08 00 08 00 e0 00 e8	00 08 00 08 00 e0 00 e8	00 08 00 08 00 70 00 78
4	0c 00 0c 06 00 00 0c 01	0c 00 0c 06 00 00 0c 01	0c 00 0c 06 00 00 0c 01
5	00 80 0c 80 0c 00 00 02	00 80 0c 80 0c 00 00 02	00 80 0c 80 0c 00 00 02
6	00 00 00 60 00 00 06 00	00 00 00 60 00 00 06 00	00 00 00 60 00 00 06 00
7	00 00 00 00 00 00 00 10	00 00 00 00 00 00 00 10	00 00 00 00 00 00 00 10
8	a0 00 a0 00 00 00 a0 00	a0 00 a0 00 00 00 a0 00	a0 00 a0 00 00 00 a0 00
9	a1 e0 21 e0 80 00 81 e0	a1 e0 21 e0 80 00 81 e0	a1 e0 21 e0 80 00 81 e0
Probability	2^{-70}	2^{-71}	2^{-72}

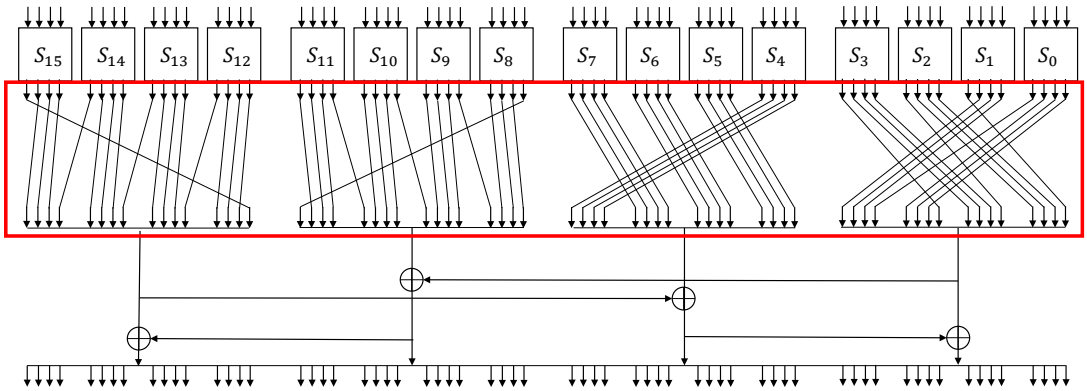

Figure 2: Simplified linear layer of BORON.

against BORON. Additionally, we notice that the previous attack has omitted the time complexity of the data preparation phase which we will estimate more accurately in the following section.

4.2. 9-round Attack on BORON-80 with an Improved Differential

The best 8-round differential characteristic used in [32], 0000 0800 0010 0000 → 0041 0041 0008 0009 holds with a probability of 2^{-62}. For the same characteristic, we found a total of 111 other characteristics that conform to the differential. This improves the overall differential probability from 2^{-62} to $2^{-58.156}$. We encrypt $2^{58.156}$ pairs and expect 1 right pair to follow the given differential. We encrypt all pairs and keep the ones that have an output difference of 00** 00** 000* 000* after the S-box layer, where * represents all possible output differences of the S-box. Based on BORON's DDT, input differences of 1, 4, 8, and 9 all have 6 possible values for *. Thus, each "00**" block can take on 6^2 possible 16-bit values whereas each "000*" block can take on 6 possible 16-bit values. After applying these filters, the number of valid pairs are reduced to $2^{58.16} \times (\frac{6^2}{16^4})^2 \times (\frac{6}{16^4})^2 \approx 2^{9.67}$.

We guess the same 24 (equivalent) subkey bits corresponding to each unknown nibble and partially decrypt each pair. We check if the result of the partial decryption matches the output of the distinguisher. If it does, a counter corresponding to the subkey guess is incremented. A total of 2^{24} counters are required for the attack. The candidate subkey with the highest counter value is taken as the right subkey. The overall time complexity calculation is as follows:

- To produce the required plaintext-ciphertext pairs, we require around $2^{59.156}$ 8-round BORON encryptions and a partial BORON encryption for the 9th round to obtain the corresponding $2^{59.156}$ ciphertexts. Taking the conservative estimate that each substitution, shift and XOR operation are equivalent, the 9th round partial encryption is equivalent to $\frac{16}{16+4+4+1} = \frac{16}{25}$ 1-round BORON encryptions. Thus, the time complexity for data preparation is $2^{59.156} \times (\frac{8}{9} + \frac{16}{25} \times \frac{1}{9}) \approx 2^{59.01}$ 9-round BORON encryptions.

- When guessing each subkey, we perform a partial decryption for each pair of ciphertexts, which has a complexity of $\frac{16}{25}$ 1-round BORON encryptions. This is performed for every valid pair and candidate key, which results in a time complexity of $2 \times 2^{9.67} \times \frac{16}{25} \times \frac{1}{9} \times 2^{24} \approx 2^{30.86}$.

- To recover the entire 80-bit master key, the remaining 25 bits of the key register can be brute-forced, which incurs a time complexity of 2^{56}.

The resulting time complexity of the attack is $2^{59.01} + 2^{30.86} + 2^{56} \approx 2^{59.18}$. The memory and data complexity are 2^{24} and $2^{59.156}$ respectively. To calculate the expected success probability for the attack, we adopt the same approach as [35], which estimates the probability of success as a function of the differential probability and total number of candidate pairs. The expected success probability for the attack on BORON is approximately $1 - (1 - 2^{-58.156})^{2^{58.156}} \approx 1 - e^{-1} \approx 0.632$. Inclusive of the data preparation phase, the attack proposed in [32] would require $2^{63} \times (\frac{8}{9} + \frac{16}{25} \times \frac{1}{9}) \approx 2^{62.94}$ BORON encryptions which dominates the overall attack complexity. Thus, use of a differential rather than a single characteristic results in an attack that is 13.55 times faster and requires 14.36 times less data.

4.3. 10-round Attack on BORON-128

To attack 10 rounds of BORON-128, we search for a valid 8-round differential that has a low number of active S-boxes in the output difference. We found an 8-round differential characteristic, 0000 3000 000b 0006 → a000 a000 0000 a000 that holds with a probability of 2^{-64}. A total of 8 differential characteristics conform to the given differential, improving the overall differential probability from 2^{-64} to $2^{-62.415}$. After the substitution layer and block shuffle of the 9th round, the expected output difference is 00*0 00*0 0000 00*0 as shown in Figure 4. We encrypt this expected output difference until after the substitution layer of the 10th round, resulting in an expected output difference of **** **0* 0**0 ****. Based on the unknown bits of each nibble and BORON's DDT, each nibble can take on $(6^2 \times 16^2)$, $(6^2 \times 16)$, (6×16), and $(6 \times 13 \times 16^2)$ possible 16-bit values respectively. This will be used as a filter to eliminate wrong pairs.

We encrypt $2^{62.415}$ pairs and expect 1 right pair to follow the given differential. We encrypt all pairs the full 10 rounds but only keep the ones that have an output difference of **** **0* 0**0 **** after the 10th S-box layer. We expect $2^{62.415} \times \frac{6^2 \times 16^2}{16^4} \times \frac{6^2 \times 16}{16^4} \times \frac{6 \times 16}{16^4} \times \frac{6 \times 13 \times 16^2}{16^4} \approx 2^{41.6}$ pairs to remain.

For the full 10 rounds, we guess all 64 bits of RK_{10}. This then allows to derive 21 out of 28 bits of RK_9 subkey bits which are needed to check if the given pair fulfills the 00*0 00*0 0000 00*0 round-9 difference as shown in Figure 4. We only need to guess 7 of these 28 RK_9 bits as the rest can be derived based on the key schedule. Thus, a total of 71 subkey bits need to be guessed. This includes all 64 bits of RK_{10}, $rk_{9,63}$, $rk_{9,62}$, $rk_{9,61}$, $rk_{9,56}$, $rk_{9,55}$, $rk_{9,54}$, and $rk_{9,53}$. We check if the result of the partial decryption matches the output of the distinguisher. If it does, a counter corresponding to the subkey guess is incremented. A total of 2^{71} counters are required for the attack. The candidate subkey with the highest counter value is taken as the right subkey. The overall time complexity calculation is as follows:

- To produce the required plaintext-ciphertext pairs, we require $2^{63.415}$ 10-round BORON encryptions to obtain the corresponding $2^{63.415}$ ciphertexts. One round of decryption is required to filter pairs, which has a time complexity of $\frac{1}{10} \times 2^{63.415} = 2^{60.09}$ 10-round BORON encryptions.

- When guessing each subkey, we perform two rounds of decryption for each pair of ciphertexts, which has a complexity of $\frac{2}{10} = 2^{-2.3219}$ 10-round BORON encryptions. This is performed for every valid pair and candidate key, which results in time complexity of $2 \times 2^{41.6} \times 2^{-2.3219} \times 2^{71} = 2^{111.28}$.

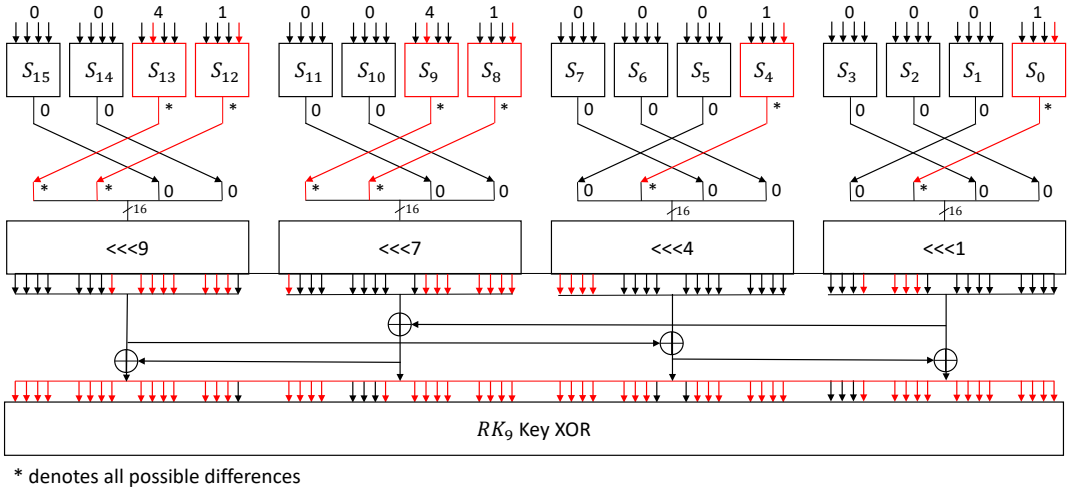

Figure 3: Key propagation for round 9 of BORON (ΔY = 0041 0041 0008 0009)

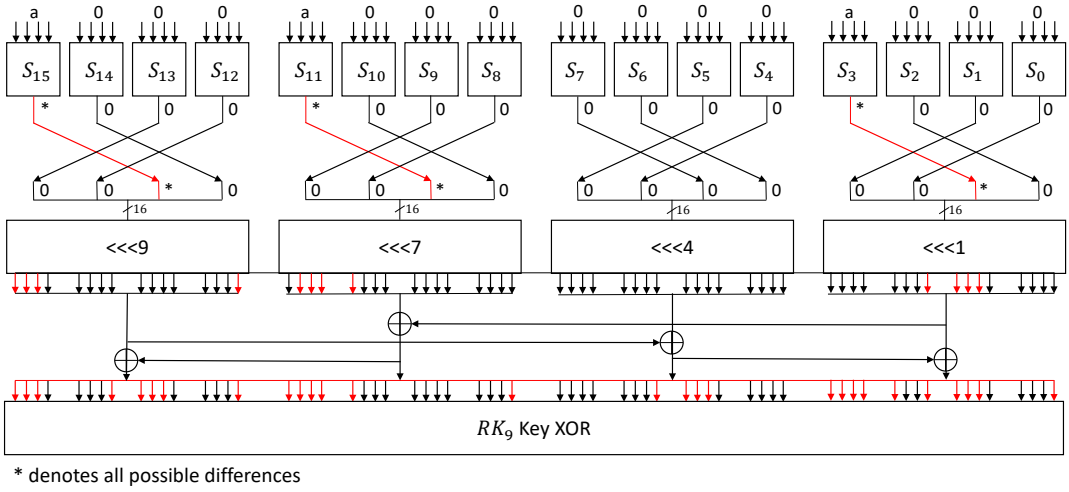

Figure 4: Key propagation for round 9 of BORON (ΔY = a000 a000 0000 a000)

- To simplify calculations, we assume that each operation (S-box, rotation, XOR) in the key schedule is equivalent in terms of computational cost. In total, there are 4 operations performed for each iteration of the key schedule. Each BORON round has 25 operations in total. Thus, the computational cost of one key schedule iteration is approximately $\frac{4}{25\times10} = 2^{-5.9658}$ 10-round BORON encryptions. This is performed for every valid pair and candidate key, which results in time complexity of $2^{41.6-5.9658+71} = 2^{106.63}$.

- To recover the entire 128-bit master key, the remaining 57 bits can be brute-forced, which incurs a time complexity of 2^{57}.

The resulting time complexity of the attack is $2^{60.09} + 2^{111.28} + 2^{106.63} + 2^{57} \approx 2^{111.34}$ 10-round BORON encryptions. The memory and data complexity are 2^{71} and $2^{63.415}$ respectively. The expected success probability for the attack on BORON-128 is approximately $1 - (1 - 2^{-62.415})^{2^{62.415}} \approx 1 - e^{-1} \approx 0.632$.

5. Conclusion

In this paper, we examined the security of the lightweight block cipher BORON against differential cryptanalysis in the single-key model. To obtain a more accurate estimation of its security margin against differential cryptanalysis, we use differentials rather than a single optimal differential characteristic when attacking the cipher. This allowed us to improve the overall distinguishing probability by finding multiple differential characteristics that conform to the same input and output differences. Notably, we found 8-round differentials of BORON with probabilities of $2^{-58.156}$ and $2^{-62.415}$ that were used to perform key recovery attacks against 9 rounds of BORON-80 and 10 rounds of BORON-128 respectively. The attack against 9 rounds of BORON-80 has a time/data/memory complexity of $2^{59.18}/2^{59.16}/2^{24}$ while the attack against 10 rounds of BORON-128 has a time/data/memory complexity of $2^{111.34}/2^{63.42}/2^{71}$. These are the best differential attacks to date against BORON in the single-key model. We also found that there is no valid

9-round differential (with a probability $> 2^{-64}$) for BORON even when multiple characteristics are taken into consideration. This supports the designers' claims that BORON's linear layer is effective in activating many S-boxes in fewer rounds. Although our attacks do not yet threaten the security of the full 25 rounds of BORON, it provides a detailed look at the cipher's security against differential cryptanalysis. Future work includes improving the time complexity of the key recovery attack against BORON-128 by adopting a divide-and-conquer strategy when guessing the final round subkeys. The differential search can also be applied to related-key or boomerang attacks against BORON to potentially attack more rounds of the cipher.

CRediT Authorship Contribution Statement

Je Sen Teh: Conceptualization, Conceptualization, Methodology, Validation, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review and Editing, Visualization, Supervision; **Li Jing Tham:** Methodology,Data Curation Validation, Investigation, Writing - Original Draft; **Norziana Jamil:** Resources, Writing - Review and Editing, Project Administration, Funding Acquisition; **Wun-She Yap:** Validation, Investigation, Writing - Review and Editing;

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the Ministry of Higher Education Malaysia through the Fundamental Research Grant Scheme with Project Code: FRGS/1/2019/ICT05/USM/02/1 and the Uniten BOLD2025 Research Fund entitled 'A Deep Learning Approach to Block Cipher Security Evaluation, Project Code J510050002/2021052. The final authenticated version of the manuscript has been published in the Journal of Information Security and Applications and is available at https://doi.org/10.1016/j.jisa.2022.103129.

References

[1] E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.3, Tech. Rep. RFC8446 (Aug. 2018). doi:10.17487/RFC8446.
URL https://www.rfc-editor.org/info/rfc8446

[2] D. Shaw, The Camellia Cipher in OpenPGP, Tech. Rep. RFC5581 (Jun. 2009). doi:10.17487/rfc5581.
URL https://www.rfc-editor.org/info/rfc5581

[3] T. Ylonen, C. Lonvick, The Secure Shell (SSH) Transport Layer Protocol, Tech. Rep. RFC4253 (Jan. 2006). doi:10.17487/rfc4253.
URL https://www.rfc-editor.org/info/rfc4253

[4] M. J. Dworkin, Recommendation for block cipher modes of operation: The CMAC Mode for Authentication, Tech. rep., National Institute of Standards and Technology (2016). doi:10.6028/nist.sp.800-38b.

[5] J. Black, P. Rogaway, CBC macs for arbitrary-length messages: The three-key constructions, J. Cryptol. 18 (2) (2005) 111–131. doi:10.1007/s00145-004-0016-3.

[6] M. Dworkin, Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC, Tech. rep., National Institute of Standards and Technology (2007). doi:10.6028/nist.sp.800-38d.

[7] A. Bogdanov, F. Mendel, F. Regazzoni, V. Rijmen, E. Tischhauser, ALE: aes-based lightweight authenticated encryption, in: S. Moriai (Ed.), Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers, Vol. 8424 of Lecture Notes in Computer Science, Springer, 2013, pp. 447–466. doi:10.1007/978-3-662-43933-3_23.

[8] D. J. Bernstein, J. Buchmann, E. Dahmen (Eds.), Post-Quantum Cryptography, Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-540-88702-7.

[9] NIST, Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process (Aug. 2018).
URL https://csrc.nist.gov/projects/lightweight-cryptography

[10] L. Sliman, T. Omrani, Z. Tari, A. E. Samhat, R. Rhouma, Towards an ultra lightweight block ciphers for internet of things, J. Inf. Secur. Appl. 61 (2021) 102897. doi:10.1016/j.jisa.2021.102897.

[11] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, C. Vikkelsoe, PRESENT: an ultra-lightweight block cipher, in: P. Paillier, I. Verbauwhede (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings, Vol. 4727 of Lecture Notes in Computer Science, Springer, 2007, pp. 450–466. doi:10.1007/978-3-540-74735-2_31.

[12] T. Suzaki, K. Minematsu, S. Morioka, E. Kobayashi, TWINE : A lightweight block cipher for multiple platforms, in: L. R. Knudsen, H. Wu (Eds.), Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers, Vol. 7707 of Lecture Notes in Computer Science, Springer, 2012, pp. 339–354. doi:10.1007/978-3-642-35999-6_22.

[13] W. Wu, L. Zhang, Lblock: A lightweight block cipher, in: J. López, G. Tsudik (Eds.), Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings, Vol. 6715 of Lecture Notes in Computer Science, 2011, pp. 327–344. doi:10.1007/978-3-642-21554-4_19.
URL https://doi.org/10.1007/978-3-642-21554-4_19

[14] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, The SIMON and SPECK lightweight block ciphers, in: Proceedings of the 52nd Annual Design Automation Conference, San Francisco, CA, USA, June 7-11, 2015, ACM, 2015, pp. 175:1–175:6. doi:10.1145/2744769.2747946.

[15] C. D. Cannière, O. Dunkelman, M. Knezevic, KATAN and KTANTAN - A family of small and efficient hardware-oriented block ciphers, in: C. Clavier, K. Gaj (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings, Vol. 5747 of Lecture Notes in Computer Science, Springer, 2009, pp. 272–288. doi:10.1007/978-3-642-04138-9_20.

[16] G. Bansod, N. Pisharoty, A. Patil, BORON: an ultra-lightweight and low power encryption design for pervasive computing, Frontiers Inf. Technol. Electron. Eng. 18 (3) (2017) 317–331. doi:10.1631/FITEE.1500415.

[17] E. Biham, A. Shamir, Differential cryptanalysis of des-like cryptosystems, J. Cryptol. 4 (1) (1991) 3–72. doi:10.1007/BF00630563.
URL https://doi.org/10.1007/BF00630563

[18] M. Matsui, On correlation between the order of s-boxes and the strength of DES, in: A. D. Santis (Ed.), Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings, Vol. 950 of Lecture Notes in Computer Science, Springer, 1994, pp. 366–375. doi:10.1007/BFb0053451.

[19] A. Biryukov, V. Velichkov, Automatic search for differential trails in ARX ciphers, in: J. Benaloh (Ed.), Topics in Cryptology - CT-RSA 2014 - The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings, Vol. 8366 of Lecture Notes in Computer Science, Springer, 2014, pp. 227–250. doi:10.1007/978-3-319-04852-9_12.

[20] J. Chen, J. Teh, Z. Liu, C. Su, A. Samsudin, Y. Xiang, Towards accurate statistical analysis of security margins: New searching strategies for differential attacks, IEEE Trans. Computers 66 (10) (2017) 1763–1777.

doi:10.1109/TC.2017.2699190.

[21] Z. Chen, J. Chen, W. Meng, J. S. Teh, P. Li, B. Ren, Analysis of differential distribution of lightweight block cipher based on parallel processing on GPU, J. Inf. Secur. Appl. 55 (2020) 102565. doi:10.1016/j.jisa.2020.102565.

[22] W. Yeoh, J. S. Teh, J. Chen, Automated search for block cipher differentials: A gpu-accelerated branch-and-bound algorithm, in: J. K. Liu, H. Cui (Eds.), Information Security and Privacy - 25th Australasian Conference, ACISP 2020, Perth, WA, Australia, November 30 - December 2, 2020, Proceedings, Vol. 12248 of Lecture Notes in Computer Science, Springer, 2020, pp. 160–179. doi:10.1007/978-3-030-55304-3_9.

[23] N. Mouha, Q. Wang, D. Gu, B. Preneel, Differential and linear cryptanalysis using mixed-integer linear programming, in: C. Wu, M. Yung, D. Lin (Eds.), Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers, Vol. 7537 of Lecture Notes in Computer Science, Springer, 2011, pp. 57–76. doi:10.1007/978-3-642-34704-7_5.

[24] S. Sun, L. Hu, M. Wang, P. Wang, K. Qiao, X. Ma, D. Shi, L. Song, K. Fu, Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties, Cryptology ePrint Archive, Report 2014/747, https://eprint.iacr.org/2014/747 (2014).

[25] A. Baksi, New insights on differential and linear bounds using mixed integer linear programming, in: D. Maimut, A. Oprina, D. Sauveron (Eds.), Innovative Security Solutions for Information Technology and Communications - 13th International Conference, SecITC 2020, Bucharest, Romania, November 19-20, 2020, Revised Selected Papers, Vol. 12596 of Lecture Notes in Computer Science, Springer, 2020, pp. 41–54. doi:10.1007/978-3-030-69255-1_4.

[26] H. Guo, S. Sun, D. Shi, L. Sun, Y. Sun, L. Hu, M. Wang, Differential attacks on CRAFT exploiting the involutory s-boxes and tweak additions, IACR Trans. Symmetric Cryptol. 2020 (3) (2020) 119–151. doi:10.13154/tosc.v2020.i3.119-151.

[27] C. Zhou, W. Zhang, T. Ding, Z. Xiang, Improving the milp-based security evaluation algorithm against differential/linear cryptanalysis using A divide-and-conquer approach, IACR Trans. Symmetric Cryptol. 2019 (4) (2019) 438–469. doi:10.13154/tosc.v2019.i4.438-469.

[28] N. Mouha, B. Preneel, Towards finding optimal differential characteristics for ARX: Application to Salsa20, Cryptology ePrint Archive, Report 2013/328, https://eprint.iacr.org/2013/328 (2013).

[29] M. Soos, K. Nohl, C. Castelluccia, Extending SAT solvers to cryptographic problems, in: O. Kullmann (Ed.), Theory and Applications of Satisfiability Testing - SAT 2009, 12th International Conference, SAT 2009, Swansea, UK, June 30 - July 3, 2009. Proceedings, Vol. 5584 of Lecture Notes in Computer Science, Springer, 2009, pp. 244–257. doi:10.1007/978-3-642-02777-2_24.

[30] R. Ankele, S. Kölbl, Mind the gap - A closer look at the security of block ciphers against differential cryptanalysis, in: C. Cid, M. J. J. Jr. (Eds.), Selected Areas in Cryptography - SAC 2018 - 25th International Conference, Calgary, AB, Canada, August 15-17, 2018, Revised Selected Papers, Vol. 11349 of Lecture Notes in Computer Science, Springer, 2018, pp. 163–190. doi:10.1007/978-3-030-10970-7_8.

[31] L. Sun, W. Wang, M. Wang, Accelerating the search of differential and linear characteristics with the SAT method, IACR Trans. Symmetric Cryptol. 2021 (1) (2021) 269–315. doi:10.46586/tosc.v2021.i1.269-315.

[32] L. Sun, W. Wang, M. Wang, Accelerating the search of differential and linear characteristics with the SAT method, IACR Trans. Symmetric Cryptol. 2021 (1) (2021) 269–315. doi:10.46586/tosc.v2021.i1.269-315.

[33] X. Lai, J. L. Massey, S. Murphy, Markov ciphers and differential cryptanalysis, in: D. W. Davies (Ed.), Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings, Vol. 547 of Lecture Notes in Computer Science, Springer, 1991, pp. 17–38. doi:10.1007/3-540-46416-6_2.

[34] V. Ganesh, D. L. Dill, A decision procedure for bit-vectors and arrays, in: W. Damm, H. Hermanns (Eds.), Computer Aided Verification, 19th International Conference, CAV 2007, Berlin, Germany, July 3-7, 2007, Proceedings, Vol. 4590 of Lecture Notes in Computer Science, Springer, 2007, pp. 519–531. doi:10.1007/978-3-540-73368-3_52.

[35] J. Lu, W. Yap, M. Henricksen, S. Heng, Differential attack on nine rounds of the SEED block cipher, Inf. Process. Lett. 114 (3) (2014) 116–123. doi:10.1016/j.ipl.2013.11.004.