RoadRunneR: A Small and Fast Bitslice Block Cipher for Low Cost 8-Bit Processors

Adnan Baysal[1,2](✉) and Sühap Şahin[2]

[1] TÜBİTAK BİLGEM, 41470 Gebze, Kocaeli, Turkey
[2] Department of Computer Engineering, Kocaeli University, 41380 Umuttepe Yerleşkesi, Kocaeli, Turkey
`adnan.baysal@tubitak.gov.tr`

Abstract. Designing block ciphers targeting resource constrained 8-bit CPUs is a challenging problem. There are many recent lightweight ciphers designed for better performance in hardware. On the other hand, most software efficient lightweight ciphers either lack a security proof or have a low security margin. To fill the gap, we present RoadRunneR which is an efficient block cipher in 8-bit software, and its security is provable against differential and linear attacks. RoadRunneR has lowest code size in Atmel's ATtiny45, except NSA's design SPECK, which has no security proof. Moreover, we propose a new metric for the fair comparison of block ciphers. This metric, called *ST/A*, is the first metric to use key length as a parameter to rank ciphers of different key length in a fair way. By using *ST/A* and other metrics in the literature, we show that RoadRunneR is competitive among existing ciphers on ATtiny45.

Keywords: Lightweight · Cryptography · Block cipher · Bitslice · 8-bit CPU · Wireless sensor network · ATtiny45

1 Introduction

As the price of small electronic devices decreases, notions like *ubiquitous computing, Internet of things,* and *smart buildings* become more popular each day. RFID tags and low cost 8-bit CPUs are commonly deployed in these applications. Atmel's ATtiny45, one of commonly used 8-bit CPUs, costs less than $1 [1]. This availability and programmable nature make these CPUs a good choice for many applications such as wireless sensor networks (WSNs).

One of the main problems in such applications is the security and privacy of information shared between devices. In many applications, data is shared between the nodes and the server over the air. Hence, an attacker can possibly get private information, or even change it for her/his benefit. For this reason, it is required to use cryptographic algorithms in these applications. Since the nodes are resource constrained in terms of memory, frequency and energy, use of lightweight cryptography becomes the best option in these applications.

Block ciphers are one of the main primitives for cryptographic applications. Therefore, the design of lightweight block ciphers has attracted many researchers'

T. Güneysu et al. (Eds.): LightSec 2015, LNCS 9542, pp. 58–76, 2016.
DOI: 10.1007/978-3-319-29078-2_4

attention, especially in the last 10 years. There are many designs, some with innovative ideas, such as LBlock [45], LED [26], PRESENT [10], PRINCE [12], PRINTcipher [32], SEA [41], TEA [44], SIMON and SPECK [5], ITUbee [28], PRIDE [3], and RECTANGLE [47]. Most of these algorithms used building blocks to optimize hardware implementations. For this reason, many of these algorithms are not a good choice for software applications in 8-bit CPUs. On the other hand, some more recent designs such as ITUbee, SPECK, PRIDE, and RECTANGLE are focused on performance in software to be an alternative in low cost CPUs. Some of these recent ciphers use bitslice substitution layers (S-boxes) where Boolean operations on CPU words are used to describe the S-box. By this approach, look-up-tables can be avoided which saves code size and CPU clock cycles. Moreover, since bitslice ciphers use small S-boxes, their hardware areas are small.

Another problem in block cipher design is the comparison of efficiencies of different ciphers for an application, and sometimes for academic purposes. Each platform and application has its own constraints and a simple comparison of area or throughput values are neither enough nor fair. Formulas for ranking block ciphers using the area-speed characteristics are needed, since implementation methods affect both values.

$\frac{Throughput}{area}$ is one metric offered in [11] to make a fair comparison of block ciphers in different hardware implementation methods (serial, parallel, pipelined, *etc.*). Badel *et al.* [4] expanded this definition by considering the possibility to trade-off throughput for power in energy-critical applications. Their formula is called *Figure Of Merit* (FOM) and defined as $\frac{Throughput}{Area^2}$. This formula is further improved by Khoo *et al.* [29] by calculating throughput at the minimum round number that the cipher is secure according to a security metric, and called this comparison metric as *Figure Of Adversarial Merit* (FOAM). In their paper, this security metric is the number of active S-boxes in differential and linear trails.

In [20], a new definition of FOM for software implementations was given. In that paper, authors suggested summing each performance indicator (code size, ram size, cycle counts) divided by the minimum of that value in the compared ciphers in a weighted manner, i.e., by multiplying each indicator with its corresponding weight. In this approach, hardest part is to find reasonable and useful weights, and they selected all weights as 1. None of the metrics above use key size in their formula. Therefore, there is no fair way of comparing ciphers of different key sizes using the metrics in the literature.

1.1 Our Contribution

We designed a new lightweight block cipher, RoadRunneR, with the goal of efficiency (especially in 8-bit low cost CPUs) and provable security in terms of minimum number of active S-boxes in differential and linear trails. The cipher is especially designed to have a very low code size, while having high throughput. Simulation results showed that on ATtiny45, our cipher have the least code size among other compared lightweight ciphers, except NSA's design SPECK which has no provable security properties to determine the round number. Our preliminary cryptanalysis showed that RoadRunneR have a relatively high security margin in contrast to most lightweight ciphers. RoadRunneR has variable area-time-security trade-off characteristics with different implementation methods so that it can fit the needs of specific application it may be used in.

Moreover, we defined a new efficiency comparison metric for block ciphers which (to the best of our knowledge for the first time) takes into account the key size of the cipher. Using this metric, we could compare block ciphers with different key sizes in a fair way. To compare RoadRunneR with existing lightweight ciphers, we used this metric and the classical ones. We gave comparison results for both the original round numbers and in the round numbers as suggested in FOAM approach. We used ATtiny45 for benchmarking, since it is one of the lowest cost 8-bit CPUs and there are many recent ciphers implemented in this device in the literature.

The organization of the rest of the paper is as follows: In Sect. 2, we define our new cipher RoadRunneR and give the design criteria of it. Preliminary cryptanalysis of RoadRunneR against known attacks is presented in Sect. 3. Our new comparison metric for block ciphers is given in Sect. 4. We give performance results of RoadRunneR and compare it with existing block ciphers using our new metric and other known metrics in Sect. 5. Section 6 concludes the paper.

2 Definition and Design Rationale of RoadRunneR

In the design of RoadRunneR, our main objectives were the following:

1. Implementation efficiency in 8-bit CPUs,
2. No table and SRAM usage,
3. Low decryption overhead,
4. Provable security like in *wide trail design strategy* [17].

We could achieve these objectives as shown in the rest of the paper. Our main focus was on reducing memory. This is because low cost 8-bit CPUs have program memory of only a few kilobytes (KB). In most applications this memory is shared by some other algorithms (such as interrupt service routines) and possibly a real time operating system. So reducing memory footprint is beneficial in our target platform.

In [39], it is stated that a hardware implementation of a lightweight block cipher targeting RFID tags and WSNs should cost less than 2000 gate equivalent (GE). For software implementations there is no stated bound, but we believe that 1KB memory should not be exceeded for a lightweight block cipher implementation.

2.1 General Structure

RoadRunneR is a Feistel-type block cipher, shown in Fig. 1, with 64-bit block size and 80-bit or 128-bit key lengths. 80-bit key requires 10 rounds and 128-bit version is 12 rounds. Initial and final round whitening is used which XOR's the whitening keys (WK_0 and WK_1) to the left part of the state. There is no swap operation in the final round. Decryption uses the same round function where the order of whitening keys, round keys and constants are reversed.

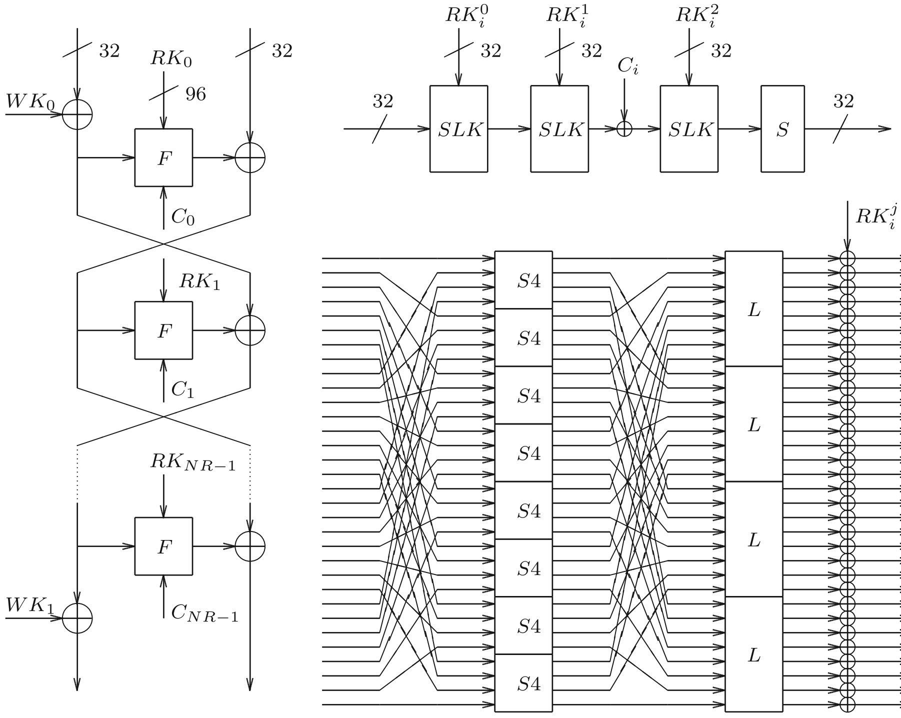

Fig. 1. Figures of functions in RoadRunneR. Feistel structure on left, F function on top right, and SLK function on bottom right.

If the state is shown as $x_0\|x_1\|\ldots\|x_7$, each round function F takes most significant (leftmost) 4 bytes of the state, that is $x_0\|x_1\|x_2\|x_3$, as input data, 1-byte C_i as constant, and 96-bit round key. Output of F is XORed to $x_4\|x_5\|x_6\|x_7$. F is a 4 round substitution-permutation-network (SPN) type function as shown on top right of Fig. 1. In that figure, SLK is the consecutive application of S-box layer (S), diffusion layer (L), and key addition (K), as shown on the bottom right. The last function S is the same S-box layer in SLK. After the second SLK function, round constant is XORed to the least significant byte (rightmost byte, i.e., x_3) of the state. For round $i = 0, 1, \ldots, NR-1$, the round constant is $C_i = NR - i$, where NR is the number of rounds, and C_i is represented as 8-bit little endian integer, that is 12 = `00001100`, 11 = `00001011`, etc. Round constants prevent simple slide attacks [8] and makes the round function to be different for different rounds. 4-round SPN-like structure ensures high number of active S-boxes for an active F.

In the *SLK* function, first a bitslice S-box layer is applied on 4 words of 8 bits. This layer can be seen as parallel application of 4-bit S-boxes to the i^{th} bit of each word, and writing the outputs to the same location (i^{th} bit) again. This is the best S-box in terms of security and efficiency (on CPUs) found in [42]. S-box layer is described in Sect. 2.2. After the S-box layer, a diffusion matrix L is applied to each byte independently to ensure diffusion inside the bits of a byte. L is designed such that it provides good diffusion and can be efficiently implemented even on the simplest CPUs. Definition and design criteria of L will be described in Sect. 2.3.

2.2 S-Box Layer

Using bitslice S-boxes has become more popular in the last 10 years, especially with the recent advances in lightweight cryptography. Block ciphers such as NOEKEON [16], SEA [41], PRIDE [3], and RECTANGLE [47] use bitslice S-boxes with different S-box layer design strategies. Bitslice S-box structure has advantage in both hardware and software implementations. In software, the permutations before and after the S-boxes disappear. In hardware, on the other hand, they can be implemented by a simple wiring which consumes no extra area. Since S-boxes of large bit size and high non-linearity have a complicated circuit representation, 3-bit and 4-bit S-boxes are used in bitslice ciphers.

In RoadRunneR, an efficient bitslice S-box is used so that it can be implemented in a small number of bit-wise operations on CPU words. The table of S-box is given below:

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	0	8	6	D	5	F	7	C	4	E	2	3	9	1	B	A

This S-box was found in [42] by a brute force search on possible assembly code combinations. The search space was restricted to 4 input words, a single temporary word, and following instructions: `AND`, `OR`, `XOR`, `NOT`, and `MOV`. For the best 4-bit S-boxes, maximal correlation and differential probabilities are 2^{-2}. They experimentally found that the minimum number of instructions to generate such a bitslice S-box layer is 9. The selected S-box in RoadRunneR satisfies this property. The assembly code of the S-box for Atmel's 8-bit CPU's is as follows (`X0` is the most significant byte entering the S-box layer):

```
; S-box layer
mov T0,X3 ; State words : X0,X1,X2,X3
and X3,X2 ; Temporary word : T0
eor X3,X1
or  X1,X2
eor X1,X0
and X0,X3
```

```
eor X0,T0
and T0,X1
eor X2,T0
```

2.3 Diffusion Layer

After the bitslice S-box layer, we used a linear function on each byte of the state to provide diffusion inside 8-bit words. So we needed an efficient linear function operating on bytes. One classical solution for such a linear function for CPUs is using XOR of shifted and rotated values of the input word.

On ATtiny45 (and on most low cost 8-bit CPUs), there is no parametric shift or and no rotation instruction. So, to shift and/or rotate a byte for parametric values multiple cycles are necessary which consumes program memory and clock cycles. 1-bit left rotation can be done in 2 instructions if `ADC` instruction is used, whereas 1-bit right rotation can be done in 3 cycles using `BST` and `BLD` instructions. There is another instruction that swaps halves of a byte, which results in a 4-bit rotation.

Using these instructions, we try to build linear functions of the form $L(x) = (x \lll i) \oplus (x \lll j) \oplus (x \lll k)$ to use in RoadRunneR, where $x \lll i$ represents i-bit rotation of the CPU word x to the left. Linear layers of this form are guaranteed to be invertible and all have branch number 4. Branch number of a matrix L is defined as follows:

$$BN(L) = min_{x \neq 0}\{hw(x) + hw(L(x))\}$$

where $hw(x)$ denotes the Hamming weight of a binary vector x. This number gives the minimum number of active S-boxes in two consecutive rounds. Besides the branch number, we calculated the minimum number of active S-boxes in 4 round SPN structure of F with each L matrix candidate. Table 1 shows the best linear functions (less then 15 instructions for two matrix multiplications) found in our search:

From the Table 1, we have chosen L_1 as our diffusion layer matrix since it provides good diffusion and performance. The minimum number of differentially active S-boxes in an active F using the above linear layers is calculated in a

Table 1. Best L matrices under given constraints.

Matrix	i, j, k	# of instructions (for two matrix mult.)	Minimum # of active S-boxes in F
L_1	0,1,2	13	10
L_2	0,1,4	11	8
L_3	0,1,5	11	8
L_4	0,4,5	11	8
L_5	1,4,5	11	8

truncated manner, i.e., it is independent of the selected S-box. We could analyze this using an observation in [25] which gives:

$$\alpha = x_0 \vee x_1 \vee x_2 \vee x_3 \rightsquigarrow L[x_0] \vee L[x_1] \vee L[x_2] \vee L[x_3] = \beta \quad (1)$$

where x_i's are 8-bit words, set bits in α and β gives the active S-box positions entering into the S-box layer and after the linear layer respectively. Since there are multiple choices for x_i's to produce the same α, some input truncated active S-box pattern will have multiple possible outputs, which we call transitions. $\alpha \rightsquigarrow \beta$ means that there is a transition from α to β. Since word size is 8 bits and there are 4 words, we could search for all possible transitions of active S-box positions while passing an SLK using an exhaustive search of 2^{32} complexity. Here we tried all possible values of $x_0\|x_1\|x_2\|x_3$ to generate $\alpha \rightsquigarrow \beta$ transitions.

Using the truncated transitions, we generated a directed graph of 256 vertices. In this graph, vertices are 8-bit numbers representing active S-box positions. If vertices α and β satisfy $\alpha \rightsquigarrow \beta$, this is shown as a directed edge from the vertex α to vertex β. Using that graph, starting from all possible vertices (except 0), we tried all possible directed paths of 4 vertices, summing the Hamming weights as the number of active S-boxes. Minimum weight in these paths give the minimum number of active S-boxes in an active F. Linear characteristics follow very similar patterns because of the definition of matrices and F function's symmetric structure.

The selected matrix have single non-trivial fixed point which is `FF` in hexadecimal notation. In truncated active S-box transition notation, we have the following fixed points: `77,7F,BB,BF,DD,DF,EE,EF,F7,FB,FD,FE,FF`. So we can say that there are at least 6 active S-boxes whenever an analysis require to use the same active S-box positions before and after SLK.

The AVR assembly code for 2 matrix multiplication is given below. The rationale behind using two matrix multiplication is to use single cycle 16-bit copy operation `MOVW` on inputs of our matrix (`lsb` and `msb` denotes least and most significant bits respectively).

```
; State registers     : X0,X1
; Temporary registers : T0,T1,ZERO (value in ZERO is 0)
movw T0,X0   ; T0,T1 <- X0,X1
lsl  T0      ; msb of T0 is moved to carry flag
adc  T0,ZERO ; Since ZERO is 0, this moves msb in carry to lsb
eor  T0,X0
lsl  T0
adc  T0,ZERO
eor  X0,T0
lsl  T1
adc  T1,ZERO
eor  T1,X1
lsl  T1
adc  T1,ZERO
eor  X1,T1
```

2.4 Key Schedules

RoadRunneR can take 80-bit or 128-bit keys. RoadRunneR-80 and RoadRunneR-128 denotes 80-bit and 128-bit key versions respectively. For both key sizes, key schedules have the same simple description: Start from the beginning of the master key, whenever a new 32-bits of key material is required, continue on the master key in a circular way.

For example, if master key is 128-bit, it will be divided into 4 words of 32-bit as $A\|B\|C\|D$. Initial whitening key is A, first round key is B-C-D, second round key is A-B-C, etc. Same key words appear in the same place in a period of 4 rounds. For 80-bit key schedule, master key is divided into 5 words of 16-bit : $A\|B\|C\|D\|E$. In this nomenclature, whitening key is $A\|B$; first round key is $(C\|D)$-$(E\|A)$-$(B\|C)$; second round key is $(D\|E)$-$(A\|B)$-$(C\|D)$, etc. There is a 5-round period in this schedule, so round number is chosen as a multiple of 5. Complete list of master key words used in rounds is given in Table 2.

Table 2. Key Schedules of RoadRunneR.

80-bit key schedule Master key = $A\Vert B\Vert C\Vert D\Vert E$		128-bit key schedule Master key = $A\Vert B\Vert C\Vert D$	
Initial whitening : $A\Vert B$		Initial whitening : A	
Rounds	Key words	Rounds	Key words
0, 5	$(C\Vert D)$-$(E\Vert A)$-$(B\Vert C)$	0, 4, 8	B-C-D
1, 6	$(D\Vert E)$-$(A\Vert B)$-$(C\Vert D)$	1, 5, 9	A-B-C
2, 7	$(E\Vert A)$-$(B\Vert C)$-$(D\Vert E)$	2, 6, 10	D-A-B
3, 8	$(A\Vert B)$-$(C\Vert D)$-$(E\Vert A)$	3, 7, 11	C-D-A
4, 9	$(B\Vert C)$-$(D\Vert E)$-$(A\Vert B)$		
Final whitening : $C\Vert D$		Final whitening : B	

3 Security of RoadRunneR

For the attacks on reduced round numbers, we assume that there is an initial and final whitening in reduced versions on the left hand side of the cipher, and no swap is applied in the last round. Final whitening is the next key word in the key schedule.

Whitening is only applied to left sides to prevent attackers from using known bits of the cipher's intermediate round values. There is no need to XOR whitening key on right halves to provide this property. In most of the attacks, extending the attack by even one round is not possible since this requires to search for all 128-bit keys (in 128-bit key version) because of the whitening. So, we can say that whitening keys play a crucial security role in RoadRunneR, and cannot be omitted in the attacks.

Since the key schedule uses the master key without any change while using in the rounds, we have no security claim against related key attacks. Related key attacks are easier to defend on the protocol level, and some of the lightweight ciphers do not consider this attack as well, such as PRINCE and PRIDE. Hence we used this assumption in RoadRunneR. In fact, each F can be passed with only two active S-boxes in a related key attack, with total of 24 active S-boxes. This total number may be further reduced in a more detailed analysis.

3.1 Differential and Linear Attacks

Differential attack [6] and linear attack [37] are two most successful attacks on block ciphers. RoadRunneR has provable bounds on minimum number of differentially active S-boxes, which shows that there is no useful differential characteristic for 5 rounds or more. Since the transpose of our diffusion layer have similar properties and F function have a symmetric structure, differential and linear properties (in a truncated manner) of RoadRunneR are the same. So whenever we mention active S-boxes, we mean both differentially and linearly active S-boxes.

As we have mentioned in Sect. 2.3, the minimum number of active S-boxes in an active F is 10. Note that this number is better than the value suggested by branch number, which is 8 for 4 rounds (since branch number is 4). There are 8 truncated trails of 10 and 11 active S-boxes, and all of them start with 1 active S-boxes. Moreover, the characteristics starting with the same input pattern follow the same path for round 2 and 3, that is only the last round is different for minimal weight paths. These paths are given below.

```
Weight 10:                       Weight 11:
01 --> 07 --> 1B --> 41          01 --> 07 --> 1B --> 49
02 --> 0E --> 36 --> 82          02 --> 0E --> 36 --> 92
04 --> 1C --> 6C --> 05          04 --> 1C --> 6C --> 25
08 --> 38 --> D8 --> 0A          08 --> 38 --> D8 --> 4A
10 --> 70 --> B1 --> 14          10 --> 70 --> B1 --> 94
20 --> E0 --> 63 --> 28          20 --> E0 --> 63 --> 29
40 --> C1 --> C6 --> 50          40 --> C1 --> C6 --> 52
80 --> 83 --> 8D --> A0          80 --> 83 --> 8D --> A4
```

We experimentally checked some high probability differential characteristics of F starting with one active S-box to see if the probability of characteristics and differentials are close or not. In our experiments, we did not see any significant increase in differential probability, from the theoretically calculated characteristic probability. So we assumed that each active S-box multiplies the probability with 2^{-2}, and an active F has approximately 2^{-20} probability.

We also calculated the minimum number of active S-boxes in r-round RoadRunneR for $4 \leq r \leq 6$, again in a truncated manner. This is done by an exhaustive search, thanks to the graph we mentioned in Sect. 2.3. Utilizing that graph, we could generate all possible truncated active S-box transitions on F function, together with their minimum number of active S-boxes. In our search of minimum number of active S-boxes in r-rounds RoadRunneR, we try all possible truncated input difference patterns to the cipher and follow an r-round path using branching over F and Feistel XORs, counting number of active S-boxes in all F functions. Whenever two truncated difference meet in an XOR of Feistel scheme, we tried both cases with a difference and without difference. Table 3 shows minimum number of active S-boxes for rounds from 4 up to 6, together with the percentage of active S-boxes.

Table 3. Minimum number of truncated active S-boxes for rounds.

Round	4	5	6
# of Active S-boxes	26	36	48
Percentage	20.3 %	22.5 %	25 %

Note that these bounds are better than the classical bounds on Feistel ciphers with invertible F function, which gives 2, 3 and 4 active F functions in 4, 5, and 6 consecutive rounds respectively. In that classical approach, since an active F has at least 10 active S-boxes, the bound is 20, 30 and 40 active S-boxes for 4, 5, and 6 rounds, whereas we found 26, 36, and 48 active S-boxes for these round numbers in our search. We believe that the active S-box percentage values are very good for such a lightweight linear layer. Table 3 proves that there is no useful differential characteristic (or linear trail) in 5 or more rounds of RoadRunneR, since the probability is at least 2^{-72}.

We listed all paths with minimum number of S-boxes in our search. By observing the trails, we saw that there were no clustering in best trails, i.e. no paths starting and ending with the same active S-box positions in 5 rounds. This gives confidence that characteristic and differential probabilities are very close in the whole cipher. Hence, we believe that 5 round RoadRunneR is secure against classical differential and linear attacks.

There are many attacks derived from differential attack and linear attack. Some examples are: higher order differential attack [34], boomerang attack [43], multidimensional linear attack [13], differential-linear attack [35], etc. In general, these extension attacks do not give better results then classical differential and linear attacks. We think that the same is true for RoadRunneR. Since all key material is used in the first and last rounds with the use of whitening keys, it is hard to apply 1R and 2R attacks.

3.2 Impossible Differential Attack

In impossible differential attack [7], truncated differentials with probability 1 are used to find a difference contradiction in middle rounds. This contradiction is then used to eliminate wrong keys in the extra rounds added before and after the characteristic. For Feistel ciphers, there is a generic 5 round impossible differential characteristic [31] as follows:

$$(0, \varDelta\alpha) \overset{5}{\nrightarrow} (0, \varDelta\alpha)$$

Since the round function of RoadRunneR has a four round SPN structure, we could not find impossible differential characteristic for more than 5 rounds. On the other hand, all key material is used in the first and last round when whitening keys are considered, so we believe that impossible differential attack cannot be applied to more than 6 rounds of RoadRunneR.

3.3 Integral Attack

The integral attack (or square attack) was demonstrated in [14] as a custom attack to SQUARE block cipher. It was also applied to Rijndael which become AES, and many other ciphers. In this attack, all possible values are given to specific bit blocks in the plaintext, and other bits are kept constant. After some rounds of encryption, fixed sum (generally zero) in specific ciphertext bit locations are expected.

Because of the 4 round SPN structure in F function, giving all possible values to a single S-box do not give too many rounds in an integral attack. The best attack can be achieved by 32-bit active block on the right half of the plaintexts as in the following:

$$(0, A) \rightarrow (A, 0) \rightarrow (A, A) \rightarrow (B, A) \rightarrow (?, B)$$

Here, A denotes an active 32-bit block where all possible values are seen. B is a balanced block, that is XOR sum of values are zero. An undetermined block is represented by a ? mark. This characteristic cannot be extended to more rounds. Therefore, we do not think that square attack threatens RoadRunneR for more than 6 rounds.

3.4 MITM-type Attacks

All state bits are affected by all key bits after 3 rounds of RoadRunneR encryption. Moreover, when the matching variable in a Meet-In-The-Middle (MITM) attack is selected in the right half of the state at the output of round 3, it is not possible to add even 2 rounds because of the fact that all key bits affect that variable in the decryption direction after 2 rounds. Same ideas apply for 2 rounds at the beginning and 3 rounds at the end case due to the Feistel structure. Hence, MITM attack cannot be applied to more than 4 rounds.

There are some extensions of MITM attack such as multidimensional MITM [48], Demirci-Selçuk attack [18], and MITM attacks with tabulation and differential enumeration techniques [22]. These attacks generally uses truncated differential characteristics with high probability over multiple rounds. In the case of RoadRunneR, since round function F is a 4-round SPN, we believe that these attacks are not more effective than basic MITM attack.

3.5 Side-Channel Attacks

Lightweight ciphers are vulnerable to side-channel attacks. The attacker can access low cost devices that have the secret key, and can measure encryption time, power dissipation, radiation, etc. Therefore, mechanisms to protect the cipher against such attacks may be necessary in some applications.

It is shown in [15] that ciphers with bitslice nonlinear layer are easier to defend against side-channel attacks such as differential power analysis (DPA) [33]. Since RoadRunneR has a bitslice non-linear layer, we can say that the additional overhead caused by DPA protecting mechanisms is low for our cipher.

4 A New Efficiency Metric for Block Ciphers : *ST/A*

In this section, we propose a new metric called ST/A, which we read as *Security times Throughput over Area*. In this new metric, we insert key size to the efficiency metric formulae since there is no fair way to compare block ciphers of different key length in the literature. We extend $\frac{Throughput}{Area}$ metric by multiplying it with the key size, so we have:

$$ST/A = \frac{KeySize \times Throughput}{Area}$$

where $KeySize$ is the bit size of key used in the cipher, *Throughput* is given in bit-per-second, and *Area* is gate equivalent (GE) in hardware or memory usage in software.

We inserted the key size by multiplication, hence increase in the key size increases the efficiency of a cipher. Moreover, other parameters affect the metric in a multiplicative manner. So another mathematical operation, such as addition, would have less meaning. In our metric, algorithms with equal round function and round number for different key sizes, such as PRESENT, will have better efficiency in higher key size. On the other hand, existing metrics in the literature do not differentiate these key sizes. This is an other evident that our metric makes more fair comparison even in this specific case.

In [20], the metric is calculated in an additive manner. On the other hand, all previous methods and ST/A are multiplicative, that is all performance value are multiplied. We think that multiplying is a more useful technique. For example, let E_1 and E_2 be two ciphers which will be compared. Also assume that all performance indicators are the same for both cipher except area values, where E_1 has area A and E_2 has area $2 \times A$. By the multiplication method, we can say that E_1 is two times better than E_2. In the summation case, however, even if the weights are equal, we would not have this ratio in efficiency values. Therefore, we multiply each indicator as in the classical $\frac{Throughput}{Area}$ formula (here throughput is multiplied by $\frac{1}{Area}$) in ST/A.

We believe that the throughput should be defined as in $FOAM$, but we leave this to the user of this metric. Weighting approach as in [20] can be used in ST/A by taking weights as powers of area, speed and key size values, since we use a multiplicative approach. Again, this is left to the user, and we use all powers as 1.

In Sect. 5, we use our metric for the comparison of the efficiency of some lightweight block ciphers and our new design RoadRunneR.

5 Performance Analysis

We have simulated RoadRunneR for ATtiny45 processor using AVR assembly language in Atmel Studio 6.2. Implementations are encryption only, where master key and plaintext are read from SRAM, plaintext is encrypted, and written back to the same place again. There is no SRAM usage besides master key and plaintext, so only code size is given as the area performance. Loading of key and plaintext, and storing back ciphertext is included in the single block encryption time. Various optimization methods are applied. In Table 4, we give the performance result of RoadRunneR block cipher.

Table 4. Performance values of RoadRunneR-80 and RoadRunneR-128 for different optimization methods. Compact-1 and Compact-2 are in between methods, shown as signs of possible trade-offs.

Key Size	Optimization	Code Size (Byte)	Time (cycle)
80	Area	202	3279
80	Speed	386	2091
128	Area	196	3819
128	Compact-1	228	2461
128	Compact-2	402	2171
128	Speed	502	2025

Area optimized 80-bit key version has slightly more area than 128-bit key version. This is because of the more complex key schedule in 80-bit key. Optimization column in Table 4 shows the various implementation methods we apply. Area optimization method gives the smallest code size we could achieve. This is done by extensive use of subroutines which saves program memory. Speed optimization, on the other hand, use no subroutines to avoid extra cycles required by branching to subroutines. Compact methods are described below:

- Compact-1 : This is derived from area optimized version. Some subroutines are removed and repeating codes are written for them.
- Compact-2 : Derived from speed optimized version. Key selection part in speed optimized version changed to a subroutine.

There are more trade-offs with different code size/clock cycle properties but we did not include them in the paper. From this and Table 4, we see that RoadRunneR have good throughput/area/security trade-off properties. If we start from the speed optimized version, we can reduce the area more than half and pay a speed penalty of less than half. Fastest implementation is still relatively small, and the smallest implementation is not that slow.

Comparison of RoadRunneR with some other ciphers is given in Table 5. We show four comparison metric values for each cipher. Metrics are explained below:

– T/A is the classical $\frac{Throughput}{Area}$ metric.
– T/A-FOAM is the same metric with the throughput definition in FOAM.
– ST/A is calculated by multiplying the T/A by the key size.
– ST/A-FOAM is calculated by multiplying the T/A-FOAM by the key size.

Instead of using $\frac{Throughput}{Area}$, we chose to use $Area \times Time$ product (time to produce 1 byte, i.e., Cycle/Byte) as in the comparisons in [28], which gives the same order. Here, in contrast to $\frac{Throughput}{Area}$, small values are better. We also normalize each comparison metric column for better understanding. For the normalization, we divide all numbers in the column with the smallest value in that column.

In the calculation of FOAM values, we searched for the best attack on each cipher in terms of round number, and used that as round number to calculate encryption time. This calculation done by multiplying the original encryption clock count by NR^*/NR, where NR is the original round number and NR^* is the round numbers calculated by the above idea. We do not exclude any initial setup since we do not know each implementation in detail. We also excluded related-key attacks since we have no security claim for this type of attack. For NOEKEON and SEA, we use the bounds found by the designers because of the lack of cryptanalysis in the literature on these ciphers.

In Table 5, (A) and (S) stands for area optimized and speed optimized implementations, respectively. (C1) and (C2) are compact implementations as defined previously. We write RRR as an abbreviation of RoadRunneR. We did not include SERPENT-128 and CLEFIA-128 in the list since they were far behind any of the other ciphers in the table in terms of efficiency metrics.

Table 5 shows that, the best cipher in terms of our metrics and classical metrics is SPECK family. But this family follows the Addition-Rotation-XOR (ARX) design principle and lacks the provable security properties. So, the round number selection of SPECK have no scientific rationale. Moreover, in an attack paper on Simon [38], authors claim that truncated differential characteristics to be found in the future may extend their 26 round attack to more rounds on the cipher. Therefore, if we exclude SIMON and SPECK, we have the following picture among remaining implementations:

RoadRunneR is the best algorithm in terms of code size (except speed optimized and C2 implementations) and security margin. When FOAM approach is not considered, i.e., in T/A and ST/A metrics, PRIDE outperforms all others, RoadRunneR implementations follow PRIDE. When we take into account security margins, T/A-FOAM and ST/A-FOAM metrics show that (A) and (C1) implementations have the highest rank, PRIDE and other implementations of RoadRunneR follow them. Throughput of RoadRunneR is not the best in any implementation but the fastest implementation of it has the rank 3 among 8

Table 5. Comparison of some block ciphers implemented on ATtiny45. RRR stands for RoadRunneR, (A), (S), (C1), and (C2) are implementations methods. Comparison metrics are normalized where small values are better.

Cipher	Block size	Key size	Attacked rounds	Mem. [byte]	Enc. Clks	Cyc./ Byte	T/A	T/A FOAM	ST/A	ST/A FOAM
AES [24]	128	128	7/10 [19]	1570	3159	197	11.83	12.35	11.11	12.38
PRESENT [24]	64	128	26/31 [9]	660	10792	1349	33.97	42.51	31.91	42.58
SEA [3]	96	96	72/93 [41]	386	17745	1479	21.78	25.43	27.26	33.96
NOEKEON [23]	128	128	9/12 [16]	364	23517	1470	20.41	22.84	19.17	22.88
PRINCE [3]	64	128	12/12 [27]	1108	3614	451	19.01	28.49	19.94	28.54
ITUBEE [28]	80	80	**10/20** [40]	716	2607	261	7.13	5.32	10.72	8.53
PRIDE [3]	64	128	19/20 [46]	266	1514	**189**	**1.92**	2.72	**1.80**	2.73
RRR-80 (A)	64	80	6/10	**202**	3279	410	3.16	2.83	4.75	4.53
RRR-80 (S)	64	80	6/10	386	2091	261	3.85	3.45	5.78	5.53
RRR-128 (A)	64	128	**6/12**	**196**	3819	477	3.57	**2.66**	3.35	**2.66**
RRR-128 (C1)	64	128	**6/12**	**228**	2461	308	2.68	**2.00**	2.51	**2.00**
RRR-128 (C2)	64	128	**6/12**	402	2171	271	4.16	3.10	3.91	3.11
RRR-128 (S)	64	128	**6/12**	502	2025	253	4.85	3.62	4.55	3.62
SIMON [5]	64	128	26/42 [2]	282	2000	250	2.69	2.37	2.53	2.37
SPECK [5]	64	96	18/26 [21]	182	1152	144	1.00	1.03	1.28	1.38
SPECK [5]	64	128	17/27 [21]	186	1200	150	1.06	1.00	1.00	1.00

ciphers. We think that RoadRunneR is fast enough for most applications with low cost 8-bit CPUs. Bold numbers show the best values in their column except SIMON and SPECK family. Multiple values in RoadRunneR implementations are written bold if they are better then all previous results.

6 Conclusion and Future Work

A very efficient Feistel type bitslice block cipher, RoadRunneR, with 64-bit block size and 80-bit or 128-bit key length is presented. RoadRunneR is a perfect choice for devices with very restricted memory resources and for applications requiring reasonable throughput expectations. Our cipher has a high security margin in contrast to most of other lightweight block ciphers. We simulated RoadRunneR on ATtiny45 devices by using Atmel Studio 6.2, for which there are implementation results of recent lightweight ciphers in the literature.

To compare our cipher and other ciphers with different key lengths, we proposed a new comparison metric which considers throughput, area and key size. When two ciphers of similar area and throughput values are achieved, the one with larger key size will have a higher rank in this metric. Our metric is the first one to use key length in the literature.

Implementation results show that RoadRunneR is a competitive candidate in all metrics in the literature and in our new metric. In our comparisons, only SPECK and PRIDE performed better than RoadRunneR in some metrics, but SPECK lacks a security proof and there is a 19 out of 20 round differential attack on PRIDE. In this sense, we think that RoadRunneR is a good alternative to current lightweight block ciphers.

Future Work : Methods for counting minimum number of active S-boxes in an r-round ($r > 2$) bitslice SPN cipher (like PRIDE and RECTANGLE) for larger than 8-bit word size is a challenge. If an efficient method can be found, this may be used to generate and evaluate binary matrices used in bitslice ciphers, together with their efficiency. Moreover, general frameworks for determining power weights for area, throughput, and key size (security) in ST/A for various implementation platforms is necessary. In the current state, we take all powers as 1, but some implementations may require very constrained area or time characteristics. How to find most useful powers is an open problem. We also leave efficient hardware implementations of RoadRunneR as a future work.

A Test Vectors for 80-Bit Key Length

Plaintext	Key	Ciphertext
0000_0000_0000_0000	0000_0000_0000_0000_0000	7F0B_3486_640D_2F5E
0000_0000_0000_0002	8000_0000_0000_0000	4FA2_5EF2_64CE_C6E4
FEDC_BA98_7654_3210	0123_4567_89AB_CDEF_0123	328C_798A_0EB2_5A3B

B Test Vectors for 128-Bit Key Length

Plaintext	Key	Ciphertext
0000_0000_0000_0000	0000_0000_0000_0000	3B07_DE72_9642_54AC
	0000_0000_0000_0000	
0000_0000_0000_0002	8000_0000_0000_0000	C168_C69A_C195_845E
	0000_0000_0000_0000	
FEDC_BA98_7654_3210	0123_4567_89AB_CDEF	D9DF_068F_5993_8882
	0123_4567_89AB_CDEF	

References

1. ATtiny45. http://www.atmel.com/devices/attiny45.aspx. Accessed 18 June 2015
2. Abed, F., List, E., Lucks, S., Wenzel, J.: Differential and linear cryptanalysis of reduced-round simon. Technical report, Citeseer (2013)
3. Albrecht, M.R., Driessen, B., Kavun, E.B., Leander, G., Paar, C., Yalçın, T.: Block ciphers – focus on the linear layer (feat. PRIDE). In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 57–76. Springer, Heidelberg (2014)

4. Badel, S., Dağtekin, N., Nakahara Jr., J., Ouafi, K., Reffé, N., Sepehrdad, P., Sušil, P., Vaudenay, S.: ARMADILLO: a multi-purpose cryptographic primitive dedicated to hardware. In: Mangard and Standaert [36], pp. 398–412
5. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The simon and speck families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404. http://eprint.iacr.org/
6. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 2–21. Springer, Heidelberg (1991)
7. Biryukov, A.: Impossible differential attack. In: van Tilborg, H.C.A. (ed.) Encyclopedia of Cryptography and Security. Springer, USA (2005)
8. Biryukov, A., Wagner, D.: Slide attacks. In: Knudsen [30], pp. 245–259
9. Blondeau, C., Nyberg, K.: Links between truncated differential and multidimensional linear properties of block ciphers and underlying attack complexities. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 165–182. Springer, Heidelberg (2014)
10. Bogdanov, A.A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
11. Bogdanov, A., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y.: Hash functions and RFID tags: mind the gap. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 283–299. Springer, Heidelberg (2008)
12. Borghoff, J., et al.: PRINCE – a low-latency block cipher for pervasive computing applications. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer, Heidelberg (2012)
13. Cho, J.Y., Hermelin, M., Nyberg, K.: A new technique for multidimensional linear cryptanalysis with applications on reduced round serpent. In: Lee, P.J., Cheon, J.H. (eds.) ICISC 2008. LNCS, vol. 5461, pp. 383–398. Springer, Heidelberg (2009)
14. Daemen, J., Knudsen, L.R., Rijmen, V.: The block cipher SQUARE. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 149–165. Springer, Heidelberg (1997)
15. Daemen, J., Peeters, M., Van Assche, G.: Bitslice ciphers and power analysis attacks. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 134–149. Springer, Heidelberg (2001)
16. Daemen, J., Peeters, M., Van Assche, G., Rijmen, V.: Nessie proposal: Noekeon (2000)
17. Daemen, J., Rijmen, V.: The wide trail design strategy. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 222–238. Springer, Heidelberg (2001)
18. Demirci, H., Selçuk, A.A.: A meet-in-the-middle attack on 8-round AES. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 116–126. Springer, Heidelberg (2008)
19. Derbez, P., Fouque, P.-A.: Exhausting demirci-selçuk meet-in-the-middle attacks against reduced-round AES. In: Moriai, S. (ed.) FSE 2013. LNCS, vol. 8424, pp. 541–560. Springer, Heidelberg (2014)
20. Dinu, D., Corre, Y.L., Khovratovich, D., Perrin, L., Großschädl, J., Biryukov, A.: Triathlon of lightweight block ciphers for the internet of things. IACR Cryptology ePrint Archive, 2015:209 (2015)
21. Dinur, I.: Improved differential cryptanalysis of round-reduced speck. Cryptology ePrint Archive, Report 2014/320 (2014). http://eprint.iacr.org/

22. Dunkelman, O., Keller, N., Shamir, A.: Improved single-key attacks on 8-round AES-192 and AES-256. J. Cryptology **28**(3), 397–422 (2015)
23. Eisenbarth, T., et al.: Compact implementation and performance evaluation of block ciphers in ATtiny devices. In: Mitrokotsa, A., Vaudenay, S. (eds.) AFRICACRYPT 2012. LNCS, vol. 7374, pp. 172–187. Springer, Heidelberg (2012)
24. Engels, S., Kavun, E.B., Paar, C., Yalçin, T., Mihajloska, H.: A non-linear/linear instruction set extension for lightweightciphers. In: Nannarelli, A., Seidel, P.-M., Tang, P.T.P. (eds.) 21st IEEE Symposium on Computer Arithmetic, ARITH 2013, Austin, TX, USA, 7–10 April 2013, p. 67–75. IEEE ComputerSociety (2013)
25. Grosso, V., Leurent, G., Standaert, F.-X., Varıcı, K.: LS-designs: bitslice encryption for efficient masked software implementations. In: Cid, C., Rechberger, C. (eds.) FSE 2014. LNCS, vol. 8540, pp. 18–37. Springer, Heidelberg (2015)
26. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.J.B.: The LED block cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Heidelberg (2011)
27. Jean, J., Nikolic, I., Peyrin, T., Wang, L., Wu, S.: Security analysis of prince. Cryptology ePrint Archive, Report 2015/372 (2015). http://eprint.iacr.org/
28. Karakoç, F., Demirci, H., Karakoç, A.E.: ITUbee: a software oriented lightweight block cipher. In: Avoine, G., Kara, O. (eds.) LightSec 2013. LNCS, vol. 8162, pp. 16–27. Springer, Heidelberg (2013)
29. Khoo, K., Peyrin, T., Poschmann, A., Yap, H.: FOAM: searching for hardware-optimal SPN structures and components with a fair comparison. IACR Cryptology ePrint Archive, 2014:530 (2014)
30. Knudsen, L.R. (ed.): FSE 1999. LNCS, vol. 1636. Springer, Heidelberg (1999)
31. Knudsen, L.R.: The security of feistel ciphers with six rounds or less. J. Cryptology **15**(3), 207–222 (2002)
32. Knudsen, L.R., Leander, G., Poschmann, A., Robshaw, M.J.B.: PRINTcipher: a block cipher for IC-printing. In: Mangard and Standaert [36], pp. 16–32
33. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
34. Lai, X.: Higher order derivatives and differential cryptanalysis. In: Blahut, R.E., Costello Jr., D.J., Maurer, U., Mittelholzer, T. (eds.) Communications and Cryptography. The Springer International Series in Engineering and Computer Science, vol. 276, pp. 227–233. Springer, US (1994)
35. Langford, S.K., Hellman, M.E.: Differential-linear cryptanalysis. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 17–25. Springer, Heidelberg (1994)
36. Mangard, S., Standaert, F.-X. (eds.): CHES 2010. LNCS, vol. 6225. Springer, Heidelberg (2010)
37. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)
38. Mourouzis, T., Song, G., Courtois, N., Christofii, M.: Advanced differential cryptanalysis of reduced-round simon64/128 using large-round statistical distinguishers. Cryptology ePrint Archive, Report 2015/481 (2015). http://eprint.iacr.org/
39. Saarinen, M.-J.O., Engels, D.W.: A do-it-all-cipher for RFID: design requirements (extendedabstract).IACR Cryptology ePrint Archive, 2012:317 (2012)
40. Soleimany, H.: Self-similarity cryptanalysis of the block cipher itubee. IET Inf. Secur. **9**(3), 179–184 (2014)
41. Standaert, F.-X., Piret, G., Gershenfeld, N., Quisquater, J.-J.: SEA: a scalable encryption algorithm for small embedded applications. In: Domingo-Ferrer, J., Posegga, J., Schreckling, D. (eds.) CARDIS 2006. LNCS, vol. 3928, pp. 222–236. Springer, Heidelberg (2006)

42. Ullrich, M., De Canniere, C., Indesteege, S., Küçük, Ö., Mouha, N., Preneel, B.: Finding optimal bitsliced implementations of 4× 4-bit s-boxes. In: SKEW Symmetric Key Encryption Workshop, Copenhagen, Denmark, pp. 16–17 (2011)
43. Wagner, D.: The boomerang attack. In: Knudsen [30], pp. 156–170
44. Wheeler, D.J., Needham, R.M.: TEA, a tiny encryption algorithm. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008. Springer, Heidelberg (1995)
45. Wu, W., Zhang, L.: LBlock: a lightweight block cipher. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 327–344. Springer, Heidelberg (2011)
46. Yang, Q., Hu, L., Sun, S., Qiao, K., Song, L., Shan, J., Ma, X.: Improved differential analysis of block cipher PRIDE. In: Lopez, J., Wu, Y. (eds.) ISPEC 2015. LNCS, vol. 9065, pp. 209–219. Springer, Heidelberg (2015)
47. Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., Verbauwhede, I.: RECTANGLE: A bit-slice ultra-lightweight block cipher suitable for multiple platforms. IACR Cryptology ePrint Archive, 2014:84 (2014)
48. Zhu, B., Gong, G.: Multidimensional meet-in-the-middle attack and its applications to KATAN32/48/64. Crypt. Commun. **6**(4), 313–333 (2014)