CHAPTER 5

RECTANGLE LIGHTWEIGHT BLOCK CIPHER

5.1 Introduction

This chapter highlights the overview of RECTANGLE lightweight block cipher. The structure of the cryptographic algorithm which is divided into the encryption algorithm and key schedule algorithm are technically explained in detail. In addition, the strengths and weaknesses of RECTANGLE are discussed in this chapter to highlight its cryptographic design issues. Lastly, the preliminary study conducted on RECTANGLE block cipher is presented.

5.2 Overview of RECTANGLE

RECTANGLE block cipher was designed for small embedded devices to provide security at a lower cost (Zhang et al., 2015). The cipher has 64 bits block size and accepts 80 or 128 bits key indicated as RECTANGLE-80 and RECTANGLE-128 respectively. RECTANGLE design adopts 25 encryption rounds using SPN structure. This algorithm enables lightweight and fast implementations using bit-slice methods (Tezcan et al., 2016). RECTANGLE provides excellent performance in both the software and hardware environments, which offers sufficient flexibility for multiple application platforms (Omrani et al., 2018b).

5.2.1 Cipher and Subkey States

The cipher state consists of a 64-bit input data for the encryption operation. The reason for naming the cipher to RECTANGLE is that it presents a cipher state in the form of 4 by 16 array of bits. Let $W = w_{63}\|...\|w_1\|w_0$ represent the cipher state. In the first 16 bits, $w_{15}\|...\|w_1\|w_0$ are positioned in Row_0 and the following 16 bits $w_{31}\|...\|w_{17}\|w_{16}$ are positioned in Row_1 and so on. Furthermore, a 64-bit subkey could be employed as 4 by 16 array bits, which is identical to the input data block. For ease of definition, a cipher state (a) is expressed in a two-dimensional manner as shown in Figure 5.1.

w_{15}	w_{14}	...	w_2	w_1	w_0
w_{31}	w_{30}	...	w_{18}	w_{17}	w_{16}
w_{47}	w_{46}	...	w_{34}	w_{33}	w_{32}
w_{63}	w_{62}	...	w_{50}	w_{49}	w_{48}

=

$a_{0,15}$	$a_{0,14}$	...	$a_{0,2}$	$a_{0,1}$	$a_{0,0}$
$a_{1,15}$	$a_{1,14}$	...	$a_{1,2}$	$a_{1,1}$	$a_{1,0}$
$a_{2,15}$	$a_{2,14}$	...	$a_{2,2}$	$a_{2,1}$	$a_{2,0}$
$a_{3,15}$	$a_{3,14}$	...	$a_{3,2}$	$a_{3,1}$	$a_{3,0}$

Figure 5.1: Cipher State

5.2.2 Encryption Algorithm

RECTANGLE operates in a 25-rounds substitution-permutation network. Every round is composed of three steps including *AddRoundKey*, *SubColumn*, and *ShiftRow*. There is another *AddRoundKey* after the last round. The encryption process for RECTANGLE algorithm is described in the following pseudo C code (Feizi et al., 2015).

GenerateRoundKeys (*Key*)
for i = 0 *to* 24
{
    AddRoundKey (*State*, K_i)
    SubColumn (*State*);
    ShiftRow (*State*);
}
AddRoundKey (*State*, K_{25})

i) *AddRoundKey*: Bitwise XOR operation of the cipher state (*a*) and the round subkey (*K*) as shown in Figure 5.2.

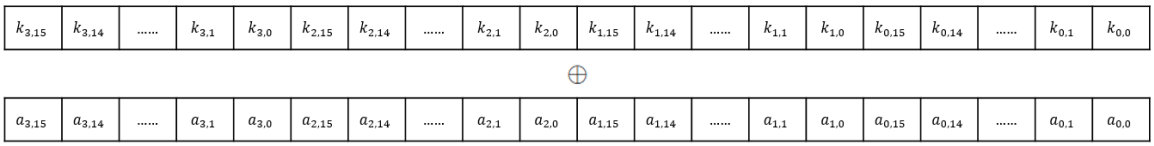

Figure 5.2: Add Round key

ii) *SubColumn*: The process of *SubColumn* is illustrated in Figure 5.3. The input of an S-box is $Col_j = a_{3,j} \| a_{2,j} \| a_{1,j} \| a_{0,j}$ for $0 \leq j \leq 15$, and the output is $(Col_j) = b_{3,j} \| b_{2,j} \| b_{1,j} \| b_{0,j}$. The RECTANGLE S-box operates as a 4-bit to 4-bit S-box as shown in Table 5.1.

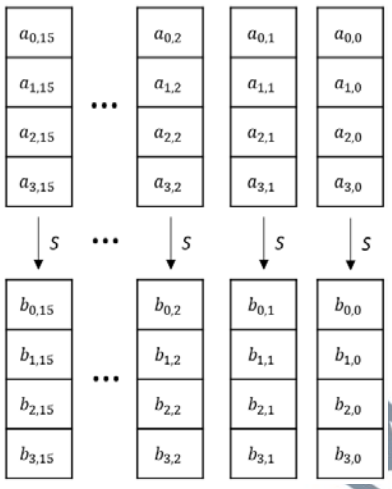

Figure 5.3: Sub Column

Table 5.1: S-box

X (Input)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$ (Output)	6	5	C	A	1	E	7	9	B	0	3	D	8	F	4	2

iii) *ShiftRow*: Every row is left rotated in a specified position. Row_0 remains constant while Row_1, Row_2, and Row_3 are left rotated over 1, 12, and 13 bits respectively as presented in Figure 5.4.

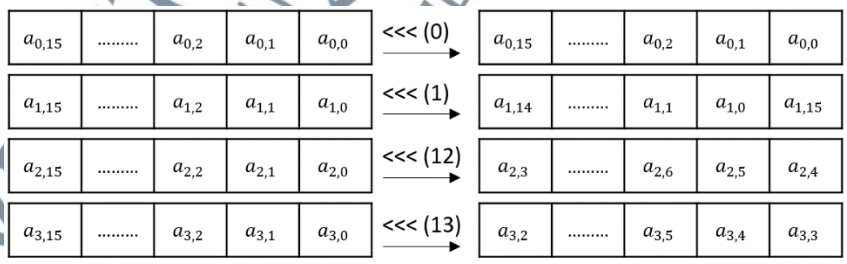

Figure 5.4: Shift Row

5.2.3 Key Schedule Algorithm

RECTANGLE utilizes 80 or 128 bits keys. Let $V = v_{79}\|...\|v_1\|v_0$ define a key as shown in Figure 5.5 (Dahiphale et al., 2019).

v_{15}	v_{14}	...	v_2	v_1	v_0		$k_{0,15}$	$k_{0,14}$	...	$k_{0,2}$	$k_{0,1}$	$k_{0,0}$
v_{31}	v_{30}	...	v_{18}	v_{17}	v_{16}		$k_{1,15}$	$k_{1,14}$	...	$k_{1,2}$	$k_{1,1}$	$k_{1,0}$
v_{47}	v_{46}	...	v_{34}	v_{33}	v_{32}	=	$k_{2,15}$	$k_{2,14}$	...	$k_{2,2}$	$k_{2,1}$	$k_{2,0}$
v_{63}	v_{62}	...	v_{50}	v_{49}	v_{48}		$k_{3,15}$	$k_{3,14}$	...	$k_{3,2}$	$k_{3,1}$	$k_{3,0}$
v_{79}	v_{78}	...	v_{66}	v_{65}	v_{64}		$k_{4,15}$	$k_{4,14}$	...	$k_{4,2}$	$k_{4,1}$	$k_{4,0}$

Figure 5.5: Key State

The 16 rightmost columns of the key are located next to one another to establish the 64-bit of the 64-bit i^{th} subkey K_i at round i for $0 \leq i \leq 25$ as shown in Figure 5.6.

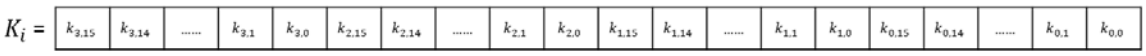

Figure 5.6: Subkey

Upon completion of K_i extraction, the key register values are updated in every round using the following steps:

i) Column$_0$ is rearranged using the S-box, i.e., $k'_{3,0}\|k'_{2,0}\|k'_{1,0}\|k'_{0,0} = S(k_{3,0}\|k_{2,0}\|k_{1,0}\|k_{0,0})$.

ii) Generalized Feistel transformation is applied, i.e., $Row_0 = (Row_0 <<< 8) \oplus Row_1$, $Row_1 = Row_2$, $Row_2 = Row_3$, $Row_3 = (Row_3 <<< 12) \oplus Row_4$, and $Row_4 = Row_0$.

iii) 5-bit key state is XORed with the round constant $Rc[i]$, i.e., $(k'_{4,0}\|k'_{3,0}\|k'_{2,0}\|k'_{1,0}\|k'_{0,0}) = (k_{4,0}\|k_{3,0}\|k_{2,0}\|k_{1,0}\|k_{0,0}) \oplus Rc[i]$ for $0 \leq i \leq 5$. Lastly, K_{25} is derived from the revised key state.

5.3 Strengths of RECTANGLE

The design of RECTANGLE consists of a nonlinear layer that applies a 4-bit S-box to each column of the state, which is represented as a 4 by 16-bit matrix. Meanwhile, the linear layer rotates each row by a different amount. The simple linear layer consists of three rotations of 16-bit words by 1, 12, and 13 bits, which can be efficiently implemented on 8, 16, and 32-bit architectures.

Implementation of bit-sliced S-box is relatively fast in software because it uses only bitwise operations. The bit-sliced design provides efficient implementations regardless of its cipher structure (Dinu, 2017). RECTANGLE has small code and RAM requirements, while being very fast on all multiple platforms.

The study shows that algorithms with a simple structure that is based solely on bitwise logical operations and rotations such as the RECTANGLE block cipher is efficient and can be accommodated in various software implementations.

5.4 Weaknesses of RECTANGLE

RECTANGLE is chosen as the focus of the study because the block cipher acquires a very competitive encryption speed performance compared to the other lightweight algorithms. Although RECTANGLE performed efficiently, its lack of focus on security deserves further investigation. Based on the literature, two weaknesses were addressed in RECTANGLE block cipher which related to the S-box and the key schedule algorithm implemented in the encryption algorithm.

5.4.1 S-box

It is found that RECTANGLE is lacking in confusion property that should be offered by an encryption algorithm (Zhang et al., 2015b). Confusion property can be achieved by introducing different classifications of 4-bit ideal S-boxes. However, the weakness of RECTANGLE S-box is identified from the analysis of undisturbed bits.

Undisturbed bits are invariant output bit differences of S-boxes and can be seen as probability one truncated differentials for S-boxes (Tezcan, 2020). For a fixed input difference, an output bit is called undisturbed if its difference remains invariant.

From the examination of the Difference Distribution Table of RECTANGLE (Senol, 2017), it can be seen that for a fixed input difference at last two bits do not change and these two bits are undisturbed bits as shown in Table 5.2.

Table 5.2: Undisturbed bits of RECTANGLE

Input	Output
1_x	??1?
4_x	??11
5_x	??0?
8_x	???1
C_x	???0

Undisturbed bits can be used to find longer differential characteristics leading to a more effective differential attacks. These bits can be used to find longer differential characteristics leading to more effective differential attack.

5.4.2 Key Schedule Algorithm

Another weakness of RECTANGLE discovered is the poor distribution of RECTANGLE key bits on the diffusion path of the encryption operations which opens it to cryptanalytic attacks (Yan et al., 2019). Diffusion property can be achieved by implementing the key schedule algorithm. However, the weakness of RECTANGLE is identified from the analysis of actual key information (AKI).

AKI depicts the interaction between the key schedule and the round function of block ciphers(Yan et al., 2019)(Yan et al., 2019)(Yan et al., 2019)(Yan et al., 2019). Given the key schedule of a block cipher, the set of all key variables is defined by K and the size of the set K is denoted by $|K|$. Let $K_0, K' \subseteq K$, K' is a reduced set of K_0 if $|K'| \leq |K_0|$ and all key variables in K_0 can be derived from the key variables in K' according to the key schedule which is denoted by $K' \Rightarrow K_0$.

The AKI of a key set K_0 is the minimum size of its reduced sets which is denoted by AKI_{K_0} that can be represented by the following equation (4).

$$\text{AKI}_{K_0} = \min_{K' \Rightarrow K_0} \{|K'|\} \tag{4}$$

AKI is denoted for simplicity if the context specifically indicates the key set K_0.

On the other hand, the theoretical key information (TKI) of the key-guessing set K_0 is defined by the following equation (5).

$$\text{TKI}_{K_0} = \min\{|K_0|, m\} \tag{5}$$

where m is the master key length. If AKI $<$ TKI, the AKI is considered insufficient and indicated as key bits leakage.

Results in Table 5.3 show that there exists key bits leakage in 2, 3, and 4-round forward paths in RECTANGLE-80. Meanwhile, for RECTANGLE-128, there exists key bits leakage in rounds 2, 3, 4, 5, and 6. The interaction between the diffusion of the key schedule and the diffusion of the round function results in AKI are insufficient in 4 consecutive rounds for RECTANGLE-80 and 6 consecutive rounds for RECTANGLE-128 respectively.

Table 5.3: Comparison of the Least AKI and its Theoretical Value TKI

Round	RECTANGLE-80		RECTANGLE-128	
	TKI	AKI	TKI	AKI
1	4	4	4	4
2	20	*18	20	*18
3	56	*42	56	*44
4	80	*73	120	*83
5	80	80	128	*103
6	-	-	128	*120
7	-	-	128	128

* denote the insufficient AKI. The search aborts when the AKI covers the whole key space

Insufficient AKI permits the adversary to get some subkey bits from some other subkey bits for free. The missing of those key bits will further reduce the time complexity of attacks or even lead to more attack rounds. If the AKI of the key-guessing sets is insufficient, then it is possible to lower the attack complexity. With proper data complexity, it is also possible to get more attacked rounds.

5.5 Preliminary Study on RECTANGLE

A preliminary study on RECTANGLE is performed to get a better understanding of the block cipher. Also, the preliminary study verifies the information that has been discussed by the previous researchers on the algorithm, especially those that are related to the weaknesses found in the block cipher. This study helps in narrowing the research scope in order to solve the existing highlighted issues. Three areas of observation were conducted that includes the randomness analysis of RECTANGLE, enhancement of the key schedule algorithm, and adoption of 3D cipher in the algorithm.

5.5.1 Randomness analysis of RECTANGLE

Randomness is an important property of a cryptography algorithm to ensure that no pattern can be identified from the ciphertext output. The randomness testing was performed using the NIST Statistical Test Suite. A total of nine data categories were applied to generate 1,000 input sequences for each algorithm. From the experiments, RECTANGLE-80 and RECTANGLE-128 passed 98.73% and 98.48% of the randomness tests (Zakaria et al., 2020c). The analysis shows that both RECTANGLE variants seem to be non-random based on the 0.1% significance level.

5.5.2 Enhancement of RECTANGLE key schedule algorithm

Key schedule algorithm is one of the core elements that significantly affect the security of an encryption algorithm. However, non-robust round keys generation from RECTANGLE key schedule algorithm seems to be the weakest point of the block cipher. Improvements to the RECTANGLE key schedule algorithm were performed to increase its randomization and confusion properties against high correlation keys as well as the speed and throughput performances (Zakaria et al., 2020b). The experimental results show that the modified designs have produced lower correlation keys by 0.16% to 0.45% improvement, more random ciphertext with an increased of 6.67% to 20.00% passing rate, and better performance that recorded 1.30% to 9.12% faster and increased by 1.33% to 10.06% throughput than the original RECTANGLE.

5.5.3 Adoption of 3D cipher in RECTANGLE

RECTANGLE block cipher has a very efficient encryption speed performance among the existing lightweight algorithms. Although RECTANGLE achieves such high efficiency, the algorithm is short of confusion and diffusion characteristics which are the cryptographic security properties. Therefore, RECTANGLE is modified using a 3D cipher to improve its security features by enhancing the algorithm confusion and diffusion properties (Zakaria et al., 2020a). Security analysis and performance tests were performed to verify the strength of the proposed 3D RECTANGLE. The results show that 3D RECTANGLE performed better than its original version in terms of the correlation between data input and output with an increased of 1.58% for non-linearity results, recorded

approximately 50% bit error rate for sensitiveness against both modifications of plaintext and key, and increased the passing rate in the randomness test by 22.22%.

5.6 Chapter Summary

This chapter identified two technical design issues associated with RECTANGLE lightweight block cipher. The first area that deserves attention is the lack of confusion property offered by the encryption algorithm which pointed to the implemented S-box in RECTANGLE. From the analysis conducted, undisturbed bits are found in the RECTANGLE S-box. These undisturbed bits can be used by an attacker to find longer differential characteristics which lead to effective differential attacks on the algorithm.

The other area to be highlighted is the weak key schedule algorithm of RECTANGLE lightweight block cipher. From the analysis of actual key information (AKI) and theoretical key information (TKI), there exists key bits leakage in both key schedule algorithms of RECTANGLE-80 and RECTANGLE-128. This vulnerability permits the attacker to reduce the time complexity of attacks and leads to more attack rounds on the algorithm.

Besides the weaknesses found in the literature, a preliminary study was conducted to verify the highlighted issues in RECTANGLE design. Observation, modification, and analysis were conducted that includes the randomness analysis of RECTANGLE, enhancement of the key schedule algorithm, and adoption of 3D cipher in the algorithm.

In order to address the highlighted issues of the RECTANGLE lightweight block cipher, a new algorithm is developed in Chapter 6. The development of the new algorithm is focused on providing the confusion and diffusion property that is lacking in the cryptographic design of RECTANGLE lightweight block cipher.