Low Power RC5 Cipher for ZigBee Portable Biomedical Systems

Yain-Reu Lin, Chia-Hao Hsu, *Student Member, IEEE*, R. Rieger, and Chua-Chin Wang, *Senior Member, IEEE*

***Abstract* – This paper presents a half-run RC5 cipher architecture with low power dissipation for transmission security of biomedical systems. The proposed architecture uses a resource-sharing approach utilizing only one adder/subtractor, one bi-directional barrel shifter, and one XOR with 32-bit bus width. Therefore, two data paths are switched through four multiplexers in the encryption/decryption procedure. A prototype chip is fabricated by a standard 0.18 μm CMOS technology. The size is 704*697 μm^2, where a total of 1.64k gates are used. The proposed architecture consumes 5.87 mW@50 MHz system clock.**

Key word: RC5, low power, cipher, biomedical system

I. INTRODUCTION

RC5, a fast block cipher, was proposed in 1994 [1], which exploits data rotation to achieve high level nonlinearity. It is commonly used in communication applications such as Wireless Transport Layer Security (WTLS), which is the security layer of Wireless Application Protocol (WAP). Moreover, the ZigBee protocol, which is very popular on WSAN (Wireless Sensor Area Network), recently also adopted the RC5 algorithm to protect the user's privacy as shown in Fig 1. Through prior works [2] and [3] have reported their designs to carry out RC5, the low-power and low-cost requirements were not seriously considered. A novel RC5 architecture is proposed to reduce power consumption and cost for WSAN applications in this study. We have firstly used Altera Quartus II Synthesis to prove its functionality. The proposed RC5 cipher is then carried out with merely 1.64 k gate count using a standard 0.18 μm CMOS technology. The proposed RC5 design is proven on silicon only consumes 5.87 mW @ 50 MHz which is more power-efficient than other known designs.

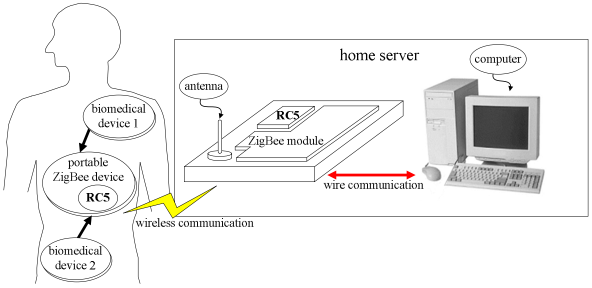


Fig. 1 A WSAN system

II. RC5 ALGORITHM

RC5 cipher mainly includes three operations: key expansion, encryption, and decryption. Encryption/decryption typically starts operating after finishing key expansion since they need subkeys generated by the key expansion. The operation of RC5 cipher only needs three logic functions. The first one is bitwise XOR, denoted by $\oplus$. The second one is modulo-2^w addition, denoted by "+", while the dual subtraction is denoted as "-". The last one is rotation of words, denoted by $x<<<y$, which means x rotated left by y bits. The reverse rotation is denoted $x>>>y$. RC5 algorithm has three parameters, denoted as $w/r/b$. w is the word size in bits, which can be one of the 3 options, 16, 32, and 64. Each of the plaintext and the ciphertext is $2w$ long. r is the number of rounds, which is from 0 to 255. b stands for the length of key in bytes. The key will generate $2r$+2 subkeys by a defined complex key expansion [1]. The pseudo-code of the key expansion is expressed as followed:

```
$i=j=X=Y=0$;
  do 3*max(t.c) times:{
    $X=S[i]=(S[i]+X+Y)<<<3$;
    $Y=L[i]=(L[i]+X+Y)<<<(X+Y)$;
    $i=(i+1)\mathrm{mod}(t)$;
    $j=(j+1)\mathrm{mod}(c)$; }
```

where i and j are counters. X and Y are registers used to store the computing temporary results of S and L. S is the key table which is mixed with L. L is filled with a secret key K[0, …, b-1] in words at the beginning, and then mixed with S.

In the encryption /decryption process of RC5 algorithm, we use a half-run algorithm. The following pseudo code is the procedure of encryption and decryption.

Encryption:

```
$A=A_{plain}+S[0]$;
$B=B_{plain}+S[1]$;
For $i$ = 1 to $r$ do {
$A=((A\oplus B)<<<B)+S[2*i]$;
$B=((B\oplus A)<<<A)+S[2*i+1]$; }
$A_{cipher}=A$;
$B_{cipher}=B$;
```

Decryption:

```
$A= A_{cipher}$;
$B= B_{cipher}$;
For $i$ = $r$ downto 1 do {
 $B=((B-S[2*i+1])>>>A)\oplus A$;
 $A=((A-S[2*i])>>>B)\oplus B$; }
$A_{plain}=B-S[1]$;
$B_{plain}=A-S[0]$;
```

where A and B are registers. $\{A_{plain}, B_{plain}\}$ are 2-w plaintext. $\{A_{cipher}, B_{cipher}\}$ are 2-w ciphertext. Thus, we can get $\{A_{cipher}, B_{cipher}\}$ from $\{A_{plain}, B_{plain}\}$ in the encryption process and $\{A_{plain}, B_{plain}\}$ from $\{A_{cipher}, B_{cipher}\}$ in the decryption process

This research was partially supported by the National Science Council under grant NSC96-2628-E-110-019-MY3 andNational Health Research Institute under grant NHRI-EX99-9732EI. It is also partially supported by Ministry of Economic Affairs, Taiwan, under grant 98-EC-17-A01-S1-133 and 98-EC-17-A-07-S2-0010. C.-H. Hsu, Y.-R. Lin, R. Rieger, and C.-C. Wang are with Department of Electrical Engineering, National Sun Yat-Sen University, 80424, Taiwan. (email: ccwang@ee.nsysu.edu.tw)

978-1-4244-8712-7/11/$26.00©2011 IEEE

III. Proposed RC5 Architecture

Fig. 2 shows the proposed encryption/decryption design, comprises 2 registers, one Circular shifter, one XOR block, and an add/sub to execute addition or subtraction. Notably, this architecture combines encryption and decryption into one core by sharing the same circuit instead of two cores in conventional architecture [1]. Besides, we replace one add/sub with two multiplexers to reduce area and power.

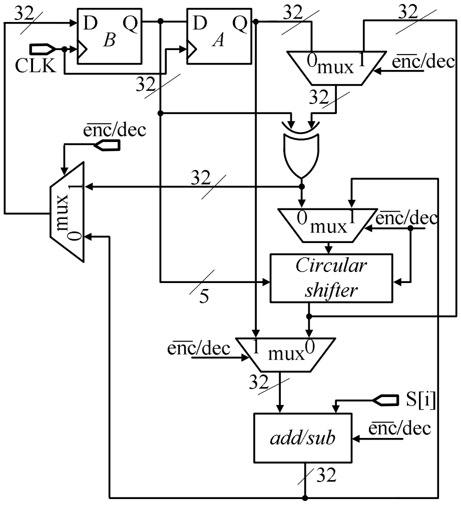

32
D
Q
B
CLK
D
Q
A
32
32
32
0 mux 1
$\overline{\text{enc}}$/dec
32
$\overline{\text{enc}}$/dec
32
mux
0
0 mux 1
$\overline{\text{enc}}$/dec
Circular shifter
5
$\overline{\text{enc}}$/dec
1 mux 0
32
S[i]
$\overline{\text{enc}}$/dec
add/sub
32

Fig. 2 Proposed encryption/decryption design

A. Encryption/Decryption process

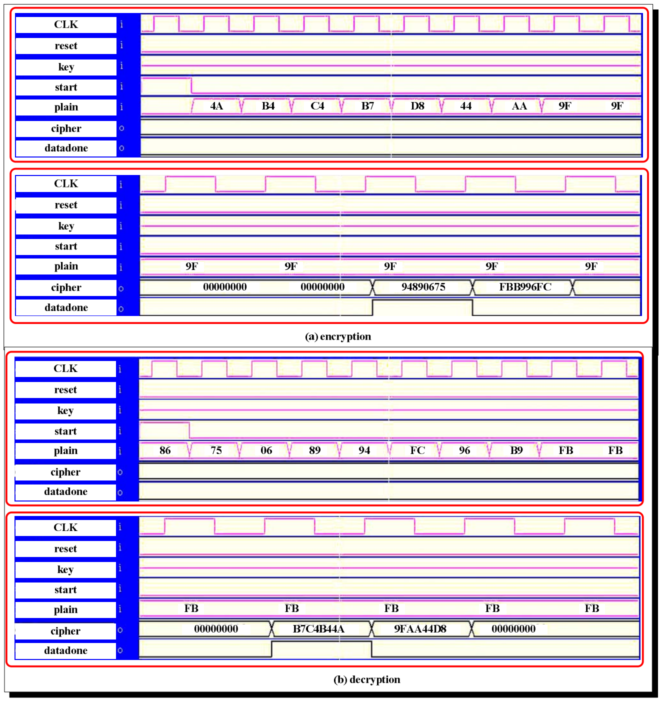
In the proposed architecture, when the selection signal, $\overline{\text{enc}}$/dec , is 0, the process of encryption is activated. On the contrary, when $\overline{\text{enc}}$/dec becomes 1, it will turn to decryption process. It needs two clock cycles to complete one round operation. Take encryption process for example. In the first clock cycle, the data stored in register *A* is XORed with *B*, and the data stored in register *B* is shifted to *A*. At this moment, only *A*, half plaintext, is encrypted. The other half plaintext, *B*, which has already been shifted to *A*, will operate in the second clock cycle.

B. Circular shifter

The bi-directional Circular shifter is composed of 32 32-to-1 multiplexers with a 5-bit selection signal. The rotational direction is determined by $\overline{\text{enc}}$/dec. When it is 0, Circular shifter executes the left rotation. When it is 1, Circular shifter executes the right rotation. Because shifting right by *N* bits is equal to shifting left by (*w*-*N*) bits, the equivalent bits of left rotation can be calculated by the 5-bit subtractor in advance to accelerate the operational speed and save more area.

IV. Implementation and Measurement

The on-silicon measurement result of the encryption and decryption processes by an Agilent 93000 SOC Test System is shown in Fig. 3.



(a) encryption

(b) decryption

Fig. 3 Measurement result of proposed RC5

Table I compares the proposed work with prior works. 1 CLBs = 0.5 LEs, where CLBs and LEs are logical elements of Xilinx and Altera, respectively. Obviously, our work is most power-efficient and area-efficient compared with others. In [5], it adopted AES (advanced encryption standard) in WSAN, but its throughput is much lower, and the gate count is four times higher than ours.

TABLE I
COMPARISON WITH PRIOR WORKS

Works	Process	Throughput (Mbps)	Maximum frequency (MHz)	Area(without memory)			Power (mW)	year
				LEs	CLBs	Gate count		
[2]	FPGA	2100	71	N/A	998	N/A	N/A	2003
[3]	FPGA	133	50	N/A	499	N/A	N/A	2005
[4]	FPGA	179.8	42	4913	2456	N/A	138.3	2008
[5]	0.25 μm	1.6	1	N/A	N/A	4k	N/A	2008
proposed	0.18 μm	133	50	N/A	N/A	**1.64k**	**5.87**	**2010**
proposed	FPGA	111.1	42	**655**	**328**	N/A	51.6	**2010**

1 CLBs = 0.5 LEs

References

[1] R. L. Rivest, "The RC5 encryption algorithm," in Proc. *1994 Leuven Workshop on Fast Software Encryption*, vol. 1008, pp. 86-96, Springer-Verlag, 1995.

[2] N. Sklavos, C. Machs, and O. Koufopavlou, "Area optimized architecture and VLSI implementation of RC5 encryption algorithm," in Proc. *10th IEEE International Conference on Electronics, Circuits and Systems*, vol. 1, pp. 172-175, Dec. 2003.

[3] L. Hua, L. Jianzhou, and Y. Jing, "An efficient and reconfigurable architecture for RC5," Canadian Conference on Electrical and Computer Engineering, pp. 1648-1651, May 2005.

[4] O. Elkeelany and S. Nimmagadda, "Effect of loop-unrolling in hardware reconfigurable implementations of RC5-192 encryption algorithm," IEEE Region 5 Conference, pp. 1-4 , Apr. 2008.

[5] K. Hyejung, K. Yongsang, and Y. Hoi-Jun, "A low energy bio sensor node processor for continuous healthcare monitoring system," IEEE Asian Solid-State Circuits Conference, pp. 317-320, Nov. 2008.