# 블록 암호 알고리즘 RC6의 구현

황재진*, 김용범*, 채현석**, 최명렬*
*한양대학교 전자전기제어계측공학과
**동원대학 모바일컨텐츠과
e-mail : heyjin25@asic.hanyang.ac.kr

# Implementation of RC6 Block Cipher

Jae-Jin Hwang*, Yong-Bum Kim*, Hyen-Seok Chae**,
Myung-Ryul Choi*
*Dept. of EECI, HanYang University
**Dept. of Mobile Contents & Internet, TongWon College

요 약

정보화 물결과 더불어 산업 및 사회 전반에 걸쳐 정보의 중요성은 나날이 증가하고 있다. 특히 인터넷을 기반으로 한 정보 교환과 통신 서비스들이 활발하게 이루어지고 있다. 신뢰성 있는 정보 교환을 위해서 정보 보호 메카니즘은 필수적이다. 많은 암호 알고리즘들이 소프트웨어 방식으로 구현되어 왔는데, 암호화 속도나 해킹에 대한 취약성이 문제되고 있다. 고속의 암호화와 보다 안전한 정보의 관리를 위해서는 암호 알고리즘의 하드웨어 구현이 요구된다. 본 논문에서는 AES의 후보 암호 알고리즘으로 채택되었으며, 우수한 암호 알고리즘으로 평가받고 있는 RC6를 하드웨어로 구현해 보았다.

## 1. 서론

급속한 정보 통신 기술의 발전에 힘입어 현대 사회는 이미 정보화 시대에 접어들었다. 산업 및 사회 전반에 걸쳐 정보의 의존성은 점점 더 증가되고 있다. 특히, 인터넷을 기반으로 한 정보 교환과 통신 서비스가 활발하게 이루어지고 있다. 따라서, 개인 신상 정보 및 금융정보 등과 같은 중요한 정보에 대한 보호는 필수적이며, 이를 위한 정보 보호 메카니즘 역시 필수적이다.

정보 보호를 위한 많은 암호 알고리즘들이 소프트웨어 방식으로 구현되어 왔는데, 이는 암호화 속도의 문제와 해킹에 의한 불법적인 정보 유출의 위험성이 높다. 따라서, 고속의 암호화와 정보의 보다 더 안전한 관리를 위해서는 암호 알고리즘의 하드웨어 구현이 요구된다. 본 논문에서는 새로운 AES (Advanced Encryption Standard)의 최종 다섯 후보에 선정되었던 블록 암호 알고리즘 RC6를 하드웨어로 구현해 보았다.

## 2. RC6 암호 알고리즘

RC6는 보다 우수한 성능과 보안에 대한 요구를 충족시키기 위해 RC5에서 개선된 블록 암호 알고리즘이다. RSA는 새로운 AES의 암호 알고리즘의 후보로 RC6를 제안했다. 그리고, NIST(National Institute of Standards and Technology)는 First AES Candidate Conference(1998년)에서 RC6를 15번째 AES의 후보로 발표하였으며, RC6는 1999년에 MARS, Rijndael, Serpent, Twofish와 함께 최종 다섯 후보로 선정되었다. 비록 AES로 Rijndael이 선택되었지만, RC6는 여전히 우수한 암호 알고리즘으로 평가받고있다[1].

RC6는 RC6-$w/r/b$로 구체화되는데, 각각의 파라메터들은 $w$-bit의 워드, $r$번의 라운딩, 그리고 $b$ bytes의 입력키를 의미한다. RC6는 AES의 요구를 충족시키기 위해, 128 bits의 평문(Plaintext)를 입력받아 128 bits의 암호문(Ciphertext)를 출력한다. 따라서, 4개의 32-bit 레지스터를 사용하고 20번의 라

운드 구성으로 목표되었다. 효과적인 입력키의 길이는 16-, 24-, 32-byte로 권고된다[2]. 블록 암호 알고리즘의 안전성은 대부분의 경우 입력키의 길이와 입력 데이터의 길이 중 작은 쪽에 의존하기 때문에 입력키 길이와 입력 데이터 길이를 동일하게 사용하고 있다. 그러므로 본 논문에서는 RC-*32/20/16*을 구현해 보았다.

### 2.1 RC6- $w/r/b$에 사용되는 기본 연산

RC6- $w/r/b$는 파라메터들에 따라 여러 변형들이 가능하지만, 표 1.에 기술한 6개의 기본 연산들을 사용한다.

표 1. RC6- $w/r/b$의 기본 연산

| | |
|---|---|
| $a + b$ | modulo $2^w$ 가산 |
| $a - b$ | modulo $2^w$ 감산 |
| $a \oplus b$ | Exclusive-OR |
| $a \times b$ | modulo $2^w$ 승산 |
| $a \lll b$ | 왼쪽으로 Rotation |
| $a \ggg b$ | 오른쪽으로 Rotation |

### 2.2 라운드키의 생성

RC6- $w/r/b$의 라운드키 생성 과정은 RC5- $w/r/b$ 와 동일하다. 단지 라운드키가 RC6- $w/r/b$에서 더 많이 유도된다는 점이 다르다.

라운드키 생성의 첫 번째 단계는 입력받은 암호키 $K[0,1,\ldots,b-1]$를 $L[0,1,\ldots,c-1]$에 저장하는 것인데, 이것은 byte에서 words로 암호키를 변환하는 것을 의미한다. 여기서, $c = \lceil b/u \rceil$ 이고 $u = w/8$이다. 두 번째 단계에서, *magic constant* $P_w$와 $Q_w$를 이용하여 $S = [0, 1, \ldots, 2r+3]$ 초기화한다. 그림 1.에 *magic constant*를 정의하였다.

$P_w = Odd((e-2)2^w)$

$Q_w = Odd((\phi-2)2^w)$

여기서,

$e = 2.71828182\ldots$ (자연대수의 밑수)

$\phi = 1.618033988\ldots$ (황금비율)

그림 1. *magic constant*의 정의

마지막으로 세 번째 단계에서, 초기화된 $S = [0, 1, \ldots, 2r+3]$를 다시 혼합(mix)한다. 라운드키 생성 단계를 그림 2.에 상세히 나타내었다.

```
1단계 :
  for i = b-1 down to do {
      L[i/u] = (L[i/u] ⋘ 8 + K[i]) }
2단계 :
  S[0] = P_w
  for i = 1 to 2r+3 do {
      S[i] = S[i-1] + Q_w }
3단계 :
  A = B = i = j = 0
  v = 3×max{c, 2r+4}
  for i = 1 to v do {
      A = S[i] = (S[i]+A+B) ⋘ 3
      B = L[j] = (L[j]+A+B) ⋘ (A+B)
      i = (i+1) mod (2r+4)
      j = (j+1) mod c }
```

그림 2. 라운드키 생성 단계

### 2.3 RC6- $w/r/b$의 암호화 과정

RC6- $w/r/b$의 암호화는 입력되는 평문을 4개의 $w$-bit 레지스터 $A, B, C, D$에 재배열시키는 것에서 시작된다. 평문의 첫 번째 byte는 레지시터 $A$의 최하위 byte에 저장되고, 평문의 마지막 byte는 레지스터 $D$의 최상위 byte에 저장된다. 그 다음, 한 번의 입력 변환과 $r$번의 라운딩을 수행하는데, 각 라운드마다 $(A, B, C, D) = (B, C, D, A)$로 재배열된다. 마지막으로 출력 라운드에서 라운드키로 한 번 더 변환된다. RC6- $w/r/b$의 암호화 과정을 그림 3.에 상세히 기술하였고, 그림 4.에 도식적으로 나타내었다.

### 2.2 RC6- $w/r/b$의 복호화 과정

RC6- $w/r/b$의 복호화 과정은 modulo $2^w$ 감산 연산을 사용하고 오른쪽으로 rotation하는 과정이 추가된 것을 제외하고는 암호화 과정과 거의 동일하다. 그림 5.에 복호화 과정을 상세히 기술하였고 그림 6.에 도식화하였다.

$B = B + S[0]$
$D = D + S[1]$
for $i = 1$ *to* $r$ *do* {
$t = (B \times (2B+1)) \lll \lg w$
$u = (D \times (2D+1)) \lll \lg w$
$A = ((A \oplus t) \lll u) + S[2i]$
$C = ((C \oplus u) \lll t) + S[2i+1]$
$(A, B, C, D) = (B, C, D, A)$
$A = A + S[2r + 2]$
$C = C + S[2r + 3]$

그림 3. RC6- $w/r/b$의 암호화 과정

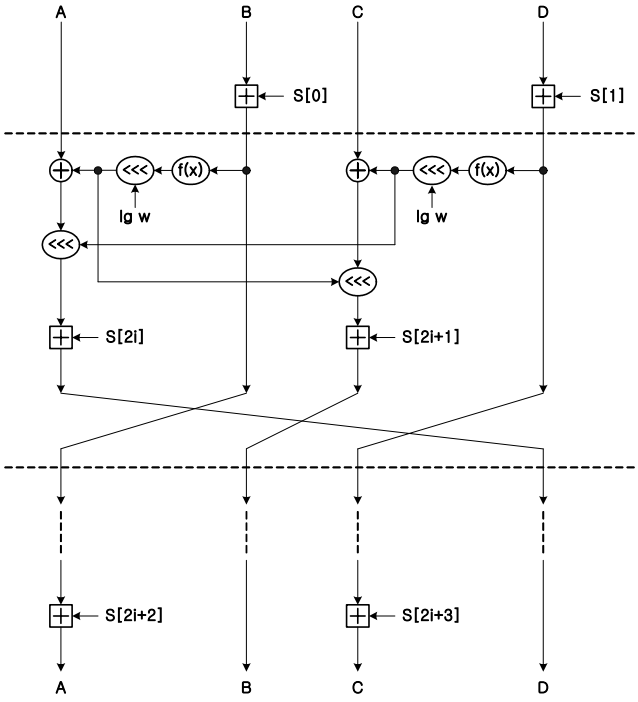


그림 4. RC6- $w/r/b$의 암호화

$C = C - S[2r + 3]$
$A = A - S[2r + 2]$
for $i = r$ *down to* 1 *do* {
$(A, B, C, D) = (D, A, B, C)$
$u = (D \times (2D+1)) \lll \lg w$
$t = (B \times (2B+1)) \lll \lg w$
$C = ((C - S[2i + 1]) \ggg t) \oplus u$
$A = ((A - S[2i]) \ggg u) \oplus t$ }
$D = D - S[1]$
$B = B - S[0]$

그림 5. RC6- $w/r/b$의 복호화 과정

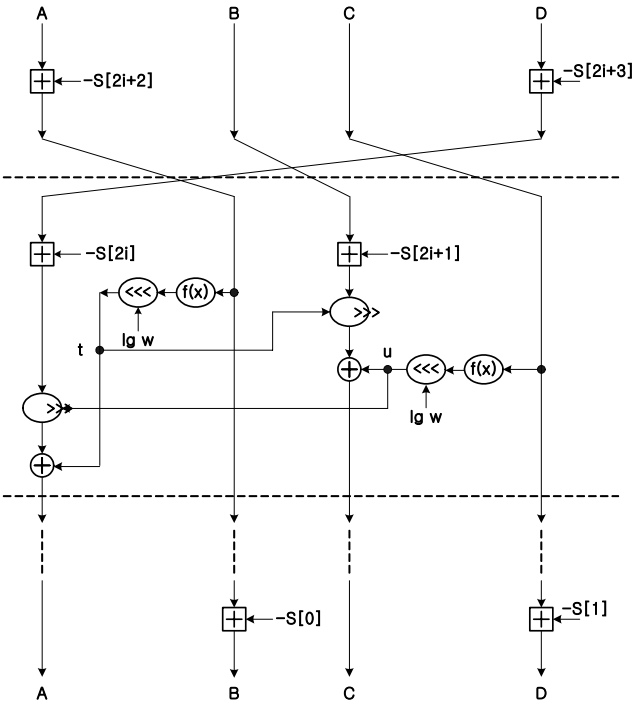


그림 6. RC6- $w/r/b$의 복호화

2.3 구현된 RC6-*32/20/16* 암호 알고리즘

본 논문에서는 Xilinx사의 ISE 5.21i를 사용하여 VHDL로 RC6-*32/20/16*를 설계하였다. 구현된 RC6-*32/20/16*은 128 bits의 평문을 입력받아 44개의 라운드키를 이용하여 1번의 입력 변환, 20번의 라운딩, 그리고 다시 1번의 출력 변환을 수행한 후, 128 bits의 암호문을 출력한다.

그림 7.은 구현한 RC6-*32/20/16*의 구조를 나타낸 것이다. 라운드키를 생성하는 key generator와 암호화 또는 복호화를 수행하는 블록으로 구성된다. ENC_MODE로 암호화 또는 복호화를 선택할 수 있도록 하였다. 암/복호화 과정에서 중복되는 연산들은 멀티플렉서를 사용하여 공유하게 함으로써 하드웨어 리소스를 감소시켰다. key generator가 라운드키를 알리는 EN_KEY 신호를 함께 출력하게 함으로써 controller를 추가로 구현하지 않았다.

## 3. 시뮬레이션 및 성능 분석

그림 8.과 그림 9.는 테스트 벡터를 이용하여 시뮬레이션을 수행한 결과로 암호화 및 복호화가 정상적으로 수행되었음을 확인할 수 있다. 사용한 테스트 벡터[3]는 표 2.에 나타내었다. Mentor Graphics사의 ModelSim을 사용하여 시뮬레이션을 수행하였다. 표 3.은 RC6-*32/20/16*의 성능 분석 결과이다.

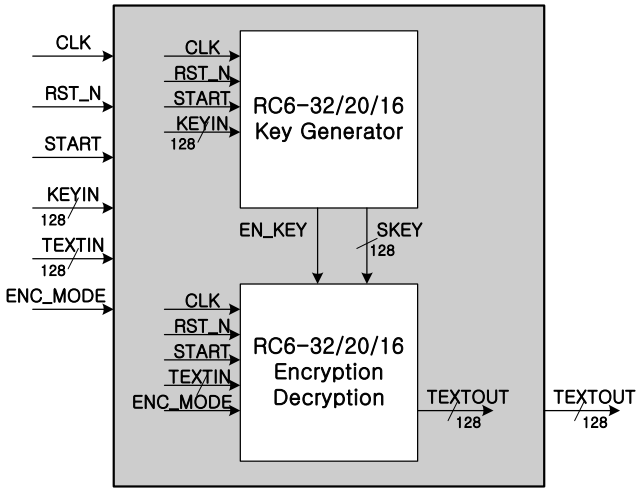


그림 7. RC6-*32/20/16*의 블록다이어그램

표 2. RC6-*32/20/16*의 테스트 벡터

| | |
|---|---|
| 입력키 | 0123456789ABCDEF0112233445566778 |
| 평문 | 02132435465768798A9BACBDCEDFE0F1 |
| 암호문 | 524E192F4715C6231F51F6367EA43F18 |

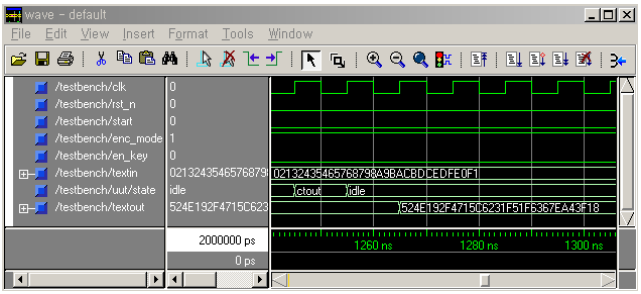


그림 8. RC6-*32/20/16*의 암호화 결과

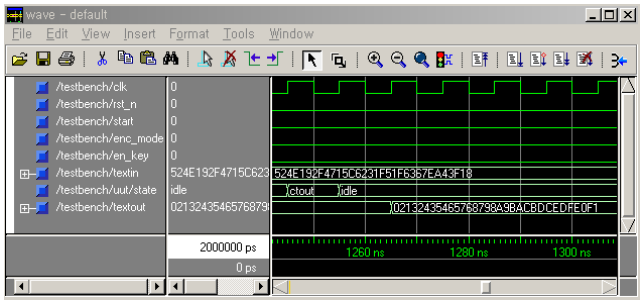


그림 9. RC6-*32/20/16*의 복호화 결과

표 3. RC6-*32/20/16*의 성능 분석

| | |
|---|---|
| Gate 수 | 약 59,325 |
| 최대 동작 주파수 | 53.7 MHz |
| Target Device | Virtex XCV1000 |

## 3. 결론 및 향후 연구 방향

본 논문에서는 소프트웨어로 구현되어왔던 암호 알고리즘의 단점을 보완하기 위해 하드웨어로 암호 알고리즘을 구현하였다. 구현한 RC6-*32/20/16*은 스마트 카드나 소형의 USB 정보 보안 저장 장치 등에 적용가능 하도록 게이트 수를 충분히 작게 설계하였다.

향후 RC6 암호 알고리즘의 성능 개선을 위한 연구를 계속해 나갈 것이며, 테스트 보드 제작을 완료하여 검증을 할 예정이다. 그리고, ECC와 Rijndael 등의 암호 알고리즘도 구현하여 검증할 예정이다.

## 참고문헌

[1] Jean-Luc Beuchat, "FPGA Implementations of the RC6 Block Cipher", P.Y.K. Cheung et al. (Eds.) : FPL 2003, LNCS 2778, pp101-110, 2003.

[2] R.L. Rivest, M.J.B Robshaw, R.Sidney, and Y.L. Yin, "The RC6 Block Cipher", First Advanced Encryption Standards(AES) Conference, 1988.

[3] M.Y. Rhee, "Internet Security : Cryptographic principles, algorithms and protocols", John Wiley, 2003.