Fast CHAM-CTR

정보컴퓨터공학과 권혁동

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

Contents

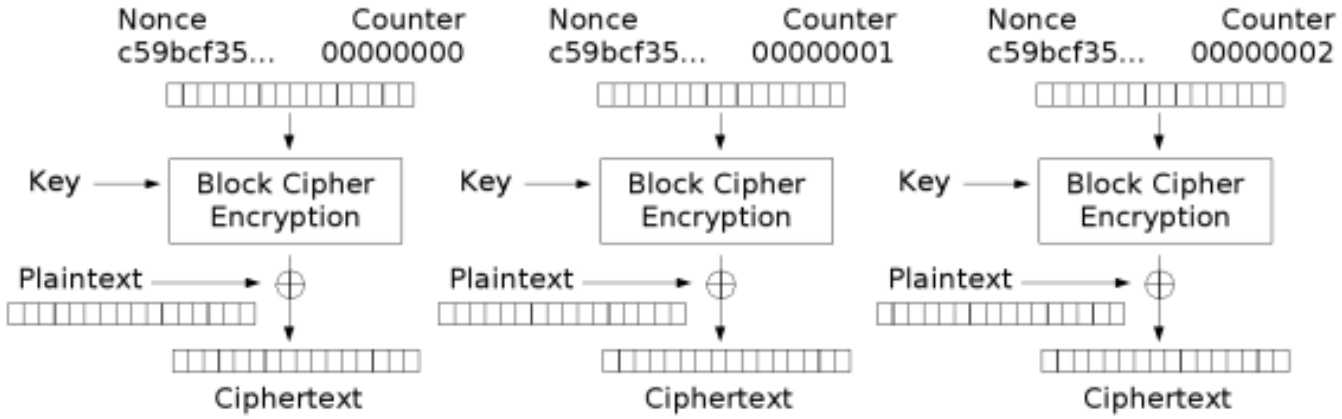
CryptoCraft LAB

카운터 모드

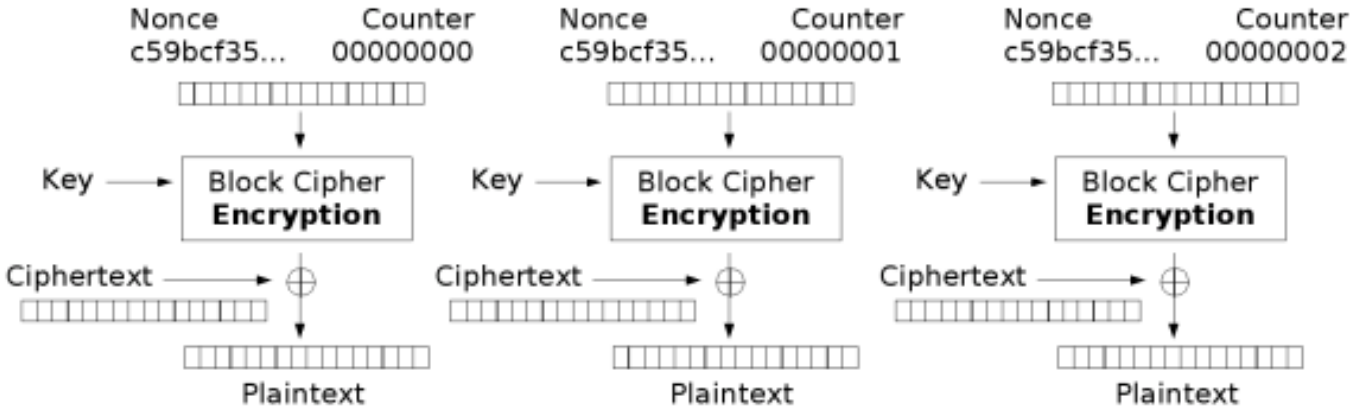
- 블록 암호 운영 모드
  - **하나의 키**를 사용하여 **반복적으로 암호화** 하는 방식
  - 암호화 키가 하나 밖에 없을 때의 위험성에서 안전

- 카운터 모드
  - 논스와 카운터를 사용하는 방식
  - 논스: 고정 난수 값
  - 카운터: 블록의 번호

카운터 모드



Counter (CTR) mode encryption



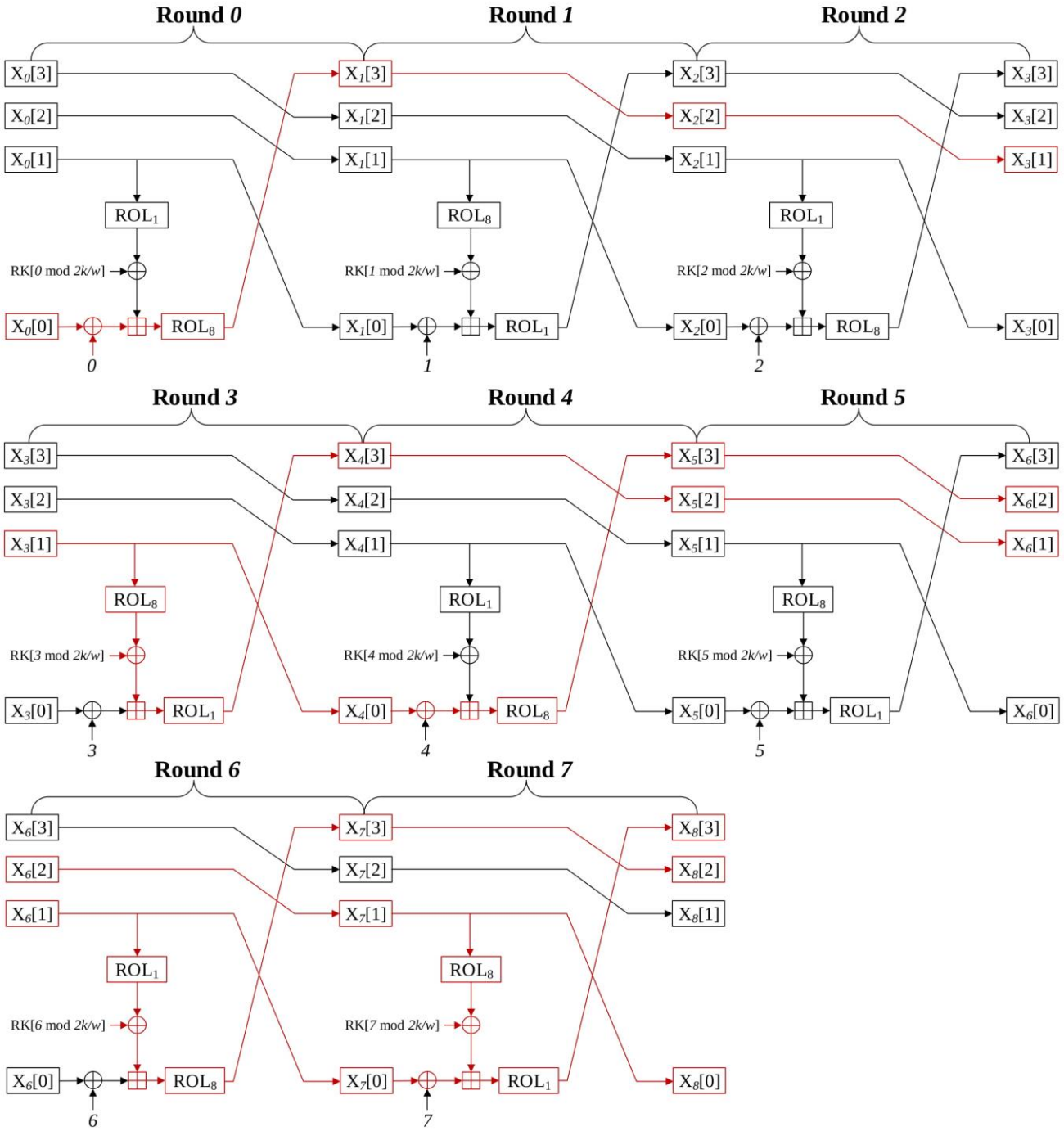
Counter (CTR) mode decryption

CHAM-CTR 최적화

- CHAM의 평문을 카운터 모드의 입력으로 대체
  - 64/128: 16비트 카운터
  - 128/128, 128/256: 32비트 카운터

- CHAM은 평문 블록을 4개로 나누어서 운영
  - 첫 블록: 카운터
  - 나머지 블록: 논스

- **논스는 고정이므로, 논스 값은 카운터와 상관 없이 연산 결과가 동일**

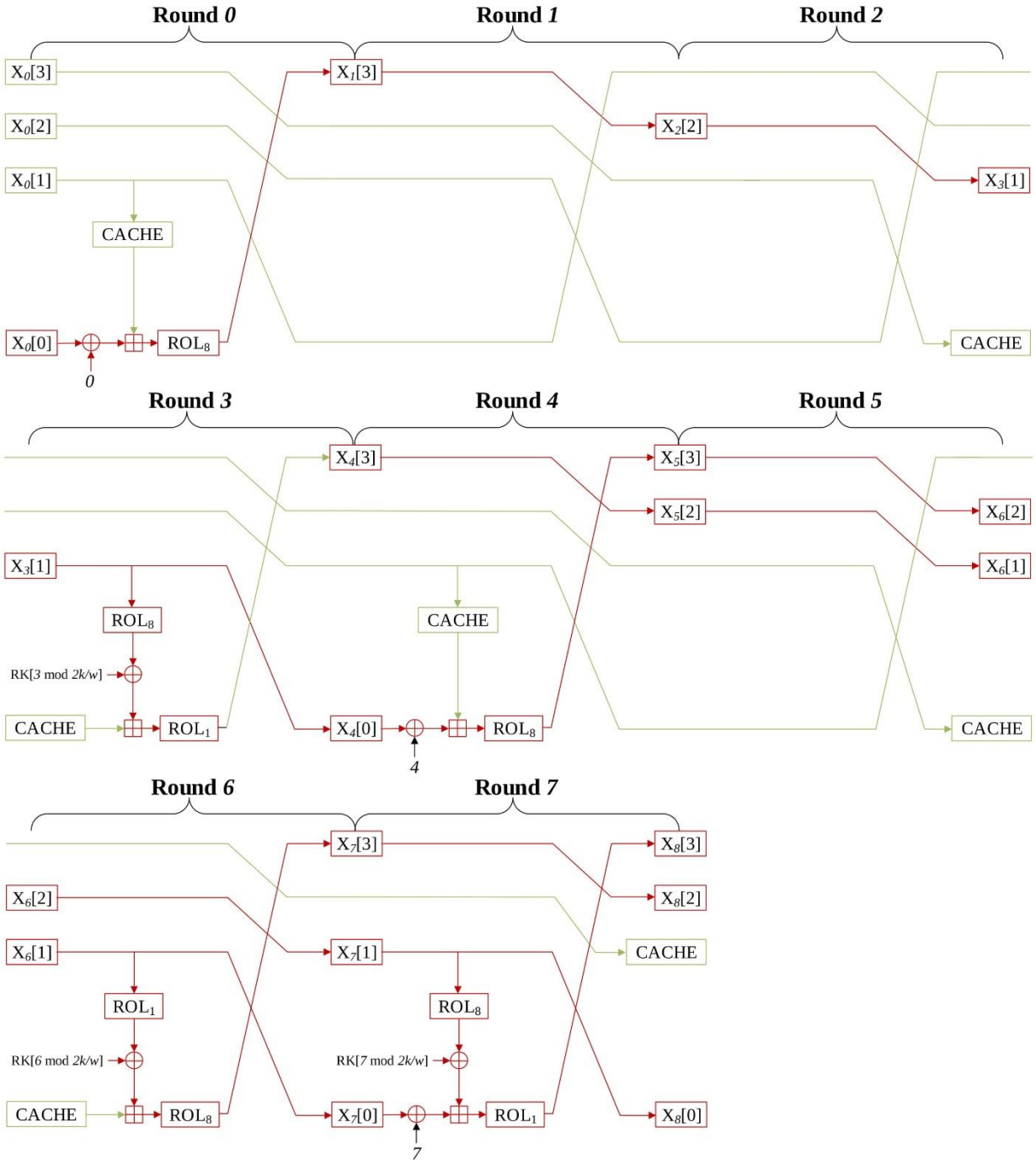


빨간색: 카운터에 영향을 받는 부분

검은색: 카운터에 영향을 받지 않는 부분

검은색 부분은 사전연산 가능

-> 연산 과정 생략 후 결과만 저장



연산 과정을 수정 가능

녹색: 연산이 생략된 부분

CHAM-CTR 최적화

- 실제 구현물
- LDI: 사전 연산 값을 로드
  - LPM명령어 대신 사용하는 것으로 2사이클 감소
- 라운드 키 값을 ROM에 두는 것으로 LPM 명령어 대신 LD 명령어 사용
  - 1사이클 감소
- 라운드 카운터와 라운드 키 주소는 몰아서 연산

```
// i = 0
LDI XT0, 0x45
LDI XT1, 0x65

ADD X00, XT0
ADC X01, XT1

// i = 1, i = 2

// i = 3
ADIW R30, 6

MOVW XT0, X00

LD RK, Z+
EOR XT0, RK
LD RK, Z+
EOR XT1, RK

LDI X30, 0x65
LDI X31, 0x77

ADD X30, XT0
ADC X31, XT1

LSL X30
ROL X31
ADC X30, ZERO

LDI RC, 4
```

```
// i = 4
EOR X01, RC

LDI XT0, 0xDC
LDI XT1, 0xCA

ADD X01, XT0
ADC X00, XT1

// i = 5
LDI X10, 0x02
LDI X11, 0X32

// i = 6
ADIW R30, 4

MOVW XT0, X30

LSL XT0
ROL XT1
ADC XT0, ZERO

LD RK, Z+
EOR XT0, RK
LD RK, Z+
EOR XT1, RK

LDI X20, 0x0B
LDI X21, 0x3D

ADD X21, XT0
ADC X20, XT1

LDI RC, 7
```

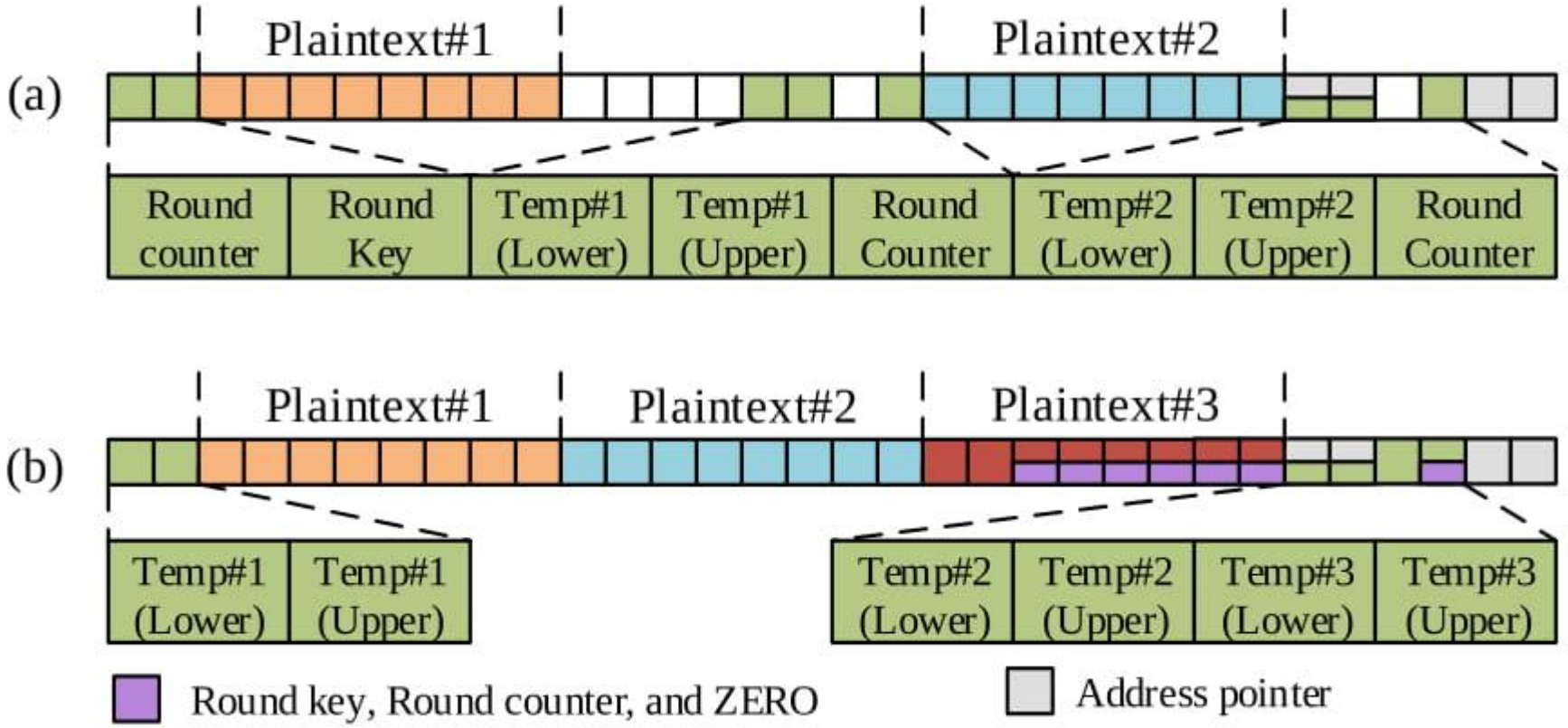
```
// i = 7
MOVW XT0, X00

EOR X30, RC

LD RK, Z+
EOR XT1, RK
LD RK, Z+
EOR XT0, RK

ADD X30, XT1
ADC X31, XT0

LSL X30
ROL X31
ADC X30, ZERO
```

CHAM 병렬 최적화

- CHAM의 가동을 병렬화
  - **한 번의 암호화로 다수의 암호문 생성**
  - 더욱 효율적인 암호 알고리즘 운영 가능
  - 128/128, 128/256은 레지스터의 부족으로 구현이 어려움

- **2병렬, 3병렬** 두 가지 형태로 나눔
- 8비트 32개 레지스터 분배의 필요

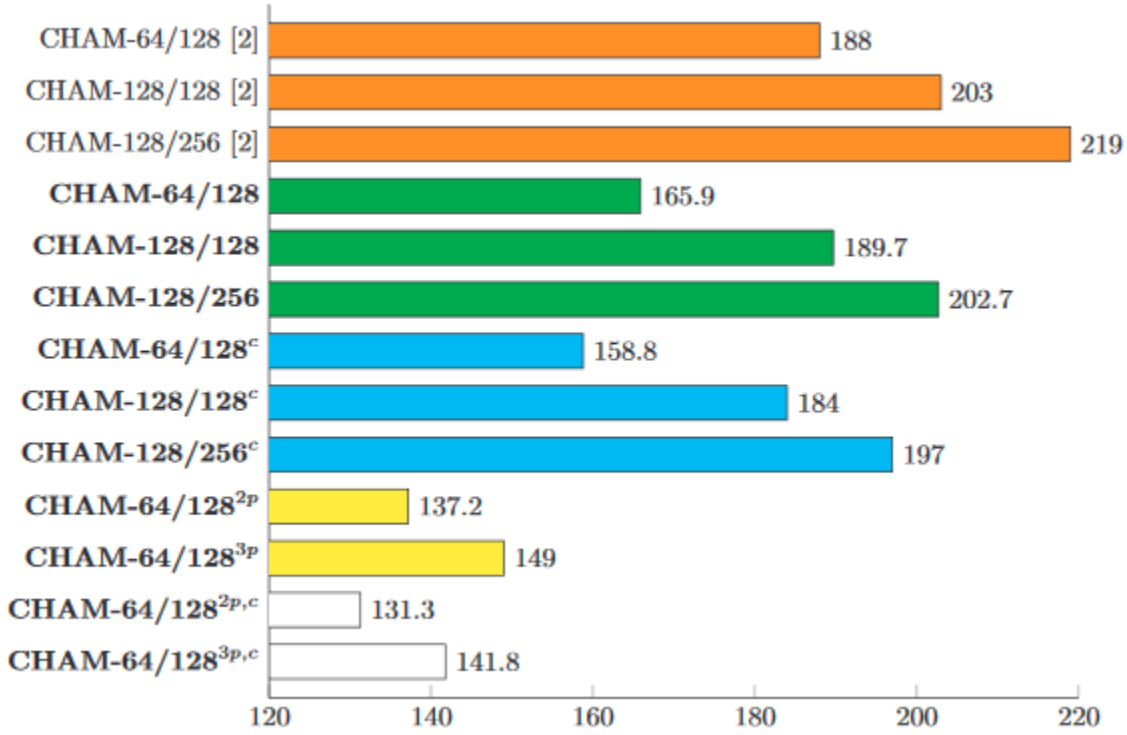
CHAM 병렬 최적화


(a)
Plaintext#1
Plaintext#2
Round counter
Round Key
Temp#1 (Lower)
Temp#1 (Upper)
Round Counter
Temp#2 (Lower)
Temp#2 (Upper)
Round Counter
(b)
Plaintext#1
Plaintext#2
Plaintext#3
Temp#1 (Lower)
Temp#1 (Upper)
Temp#2 (Lower)
Temp#2 (Upper)
Temp#3 (Lower)
Temp#3 (Upper)
Round key, Round counter, and ZERO
Address pointer

성능 평가

- Atmel Studio 7.0, -O2 옵션

구 현 물	비교
CHAM-64/128	vs Revised 11.7%
CHAM-128/128	vs Revised 6.5%
CHAM-128/256	vs Revised 7.4%
CHAM_CTR-64/128	vs CHAM 4.2%
CHAM_CTR-128/128	vs CHAM 3.0%
CHAM_CTR-128/256	vs CHAM 2.8%



결론

- CHAM-CTR 모드는 **카운터 값에 영향 받지 않는 구간**이 존재
  - 해당 구간은 연산 결과가 고정이므로 **사전연산이 가능**
  - Revised CHAM에 비해 훨씬 향상된 성능

- CHAM은 **평문 길이가 짧다**는 점을 이용하여, **병렬 연산 가능**
  - 64/128에 한정
  - 알고리즘을 2회, 3회 가동하는 것보다 높은 효율

- CTR 모드의 표준은 32-bit 카운터
  - 후행 과제로 64/128에 **32-bit 카운터를 사용한 카운터 모드 구현**