Variable-key CHAM 구현

정보컴퓨터공학과 권혁동

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

Contents

기존 기법

제안 기법

성능 평가

결론

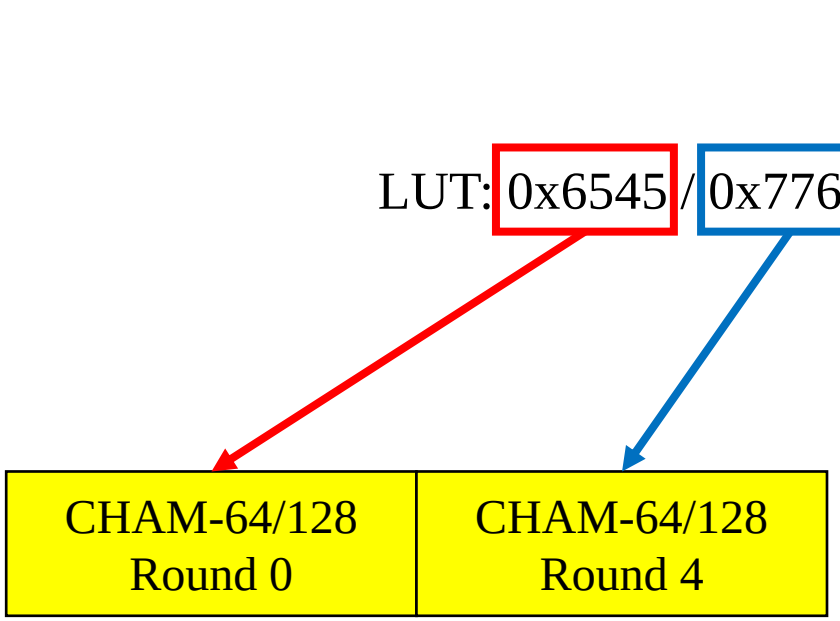
CryptoCraft LAB

기존 기법

- CHAM-CTR 모드 최적화
  - 사전연산을 통해 **일부 연산 구간을 생략**
- **고정키 시나리오**
  - 키가 고정 된 상태로 동작
  - **키 변경 시 동작하지 않는다는 문제점 존재**

기존 기법

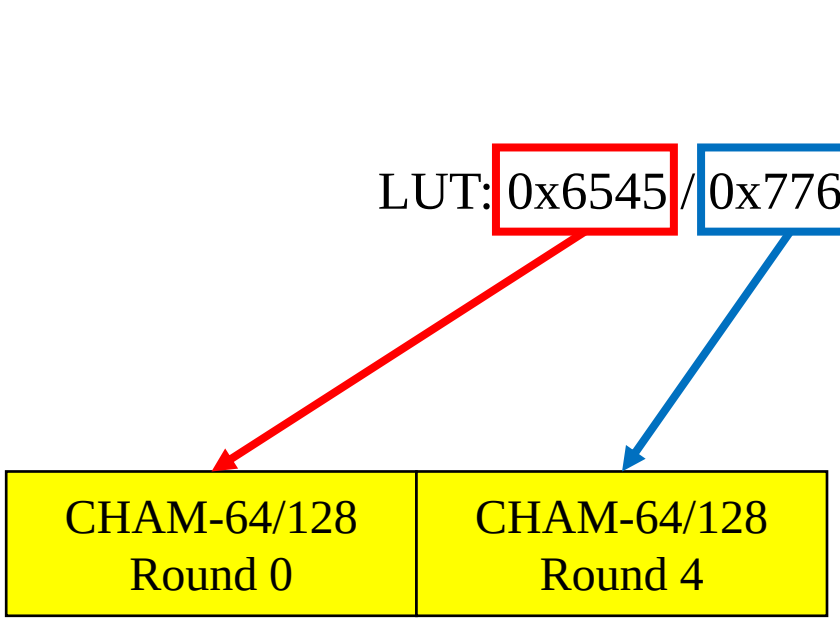
RK: 0x030107050b090f0d131117151b191f1d151e030839322f244d465b50616a777c
PT: 0x1100332255447766



- 필요한 값을 불러와서 사용 가능
- 값이 **고정**이기 때문에, **LDI 명령어 사용 가능**
  - LD: 2clock cycle, LDI: 1clock cycle

기존 기법

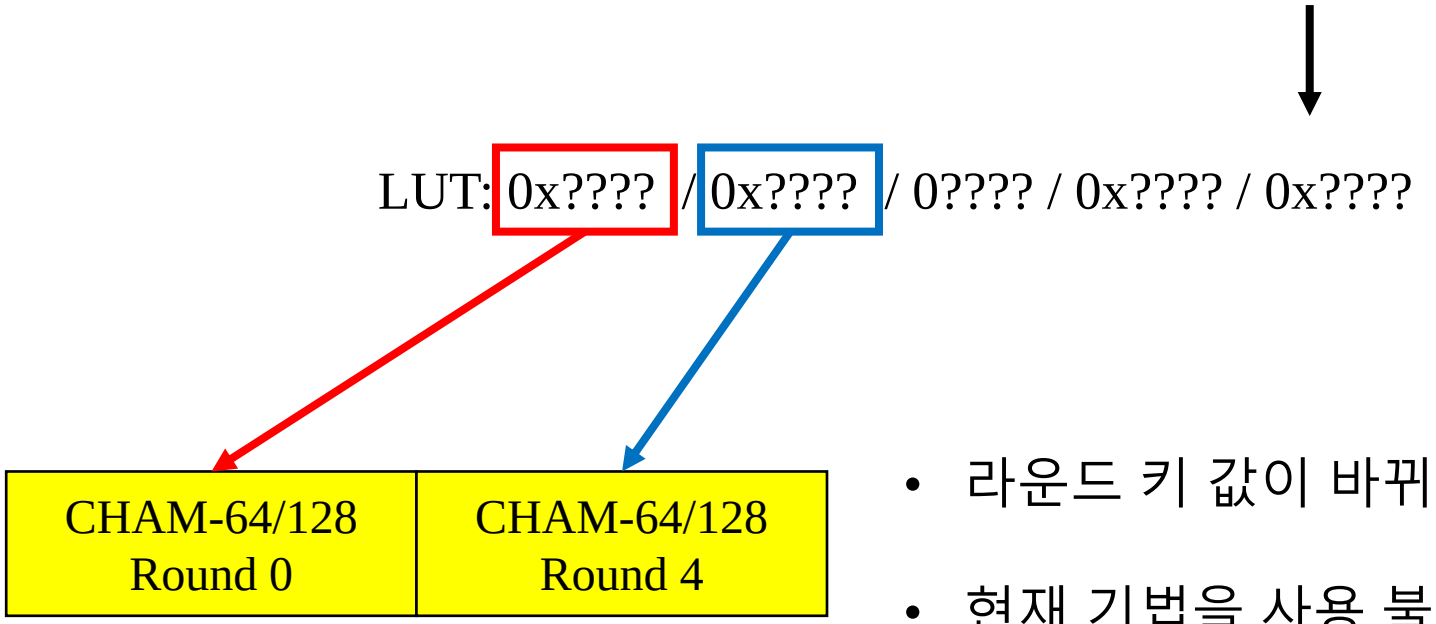
RK: 0x030107050b090f0d131117151b191f1d151e030839322f244d465b50616a777c
PT: 0x1101332255447766



- 카운터 부분은 변화하는 값
  - 값의 예측이 불가능
- 논스 부분은 고정 값
  - 해당 부분의 연산 결과를 저장해서 불러오기

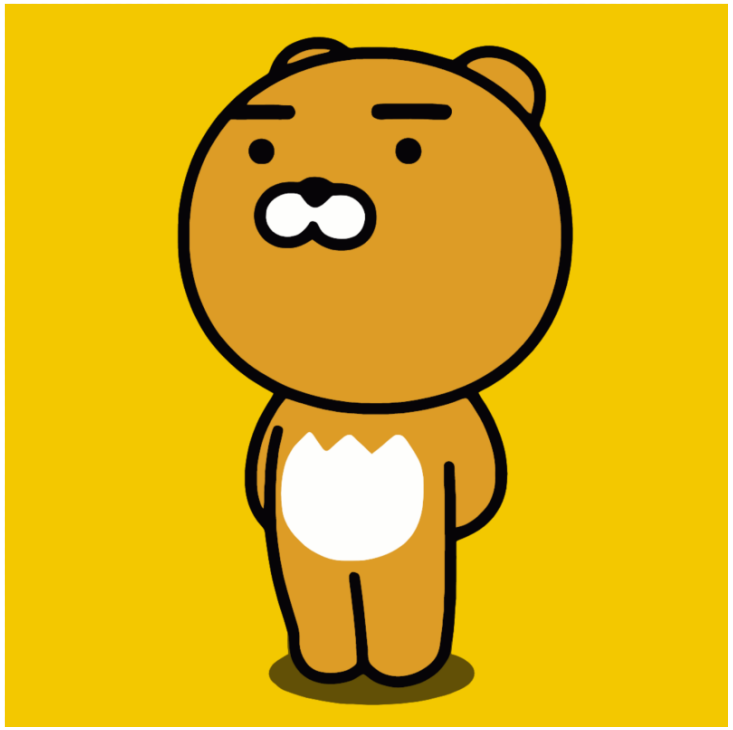
기존 기법

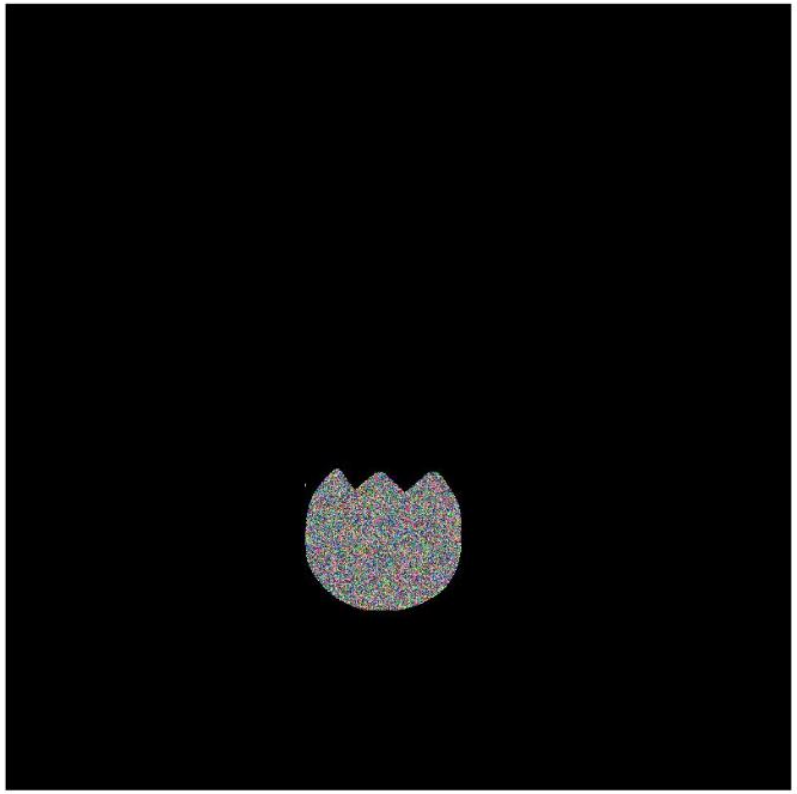
RK: 0x030207050b090f0d131117151b191f1d151e030839322f244d465b50616a777c
PT: 0x1100332255447766



- 라운드 키 값이 바뀌면, 사전 테이블도 변화
- 현재 기법을 사용 불가능

기존 기법



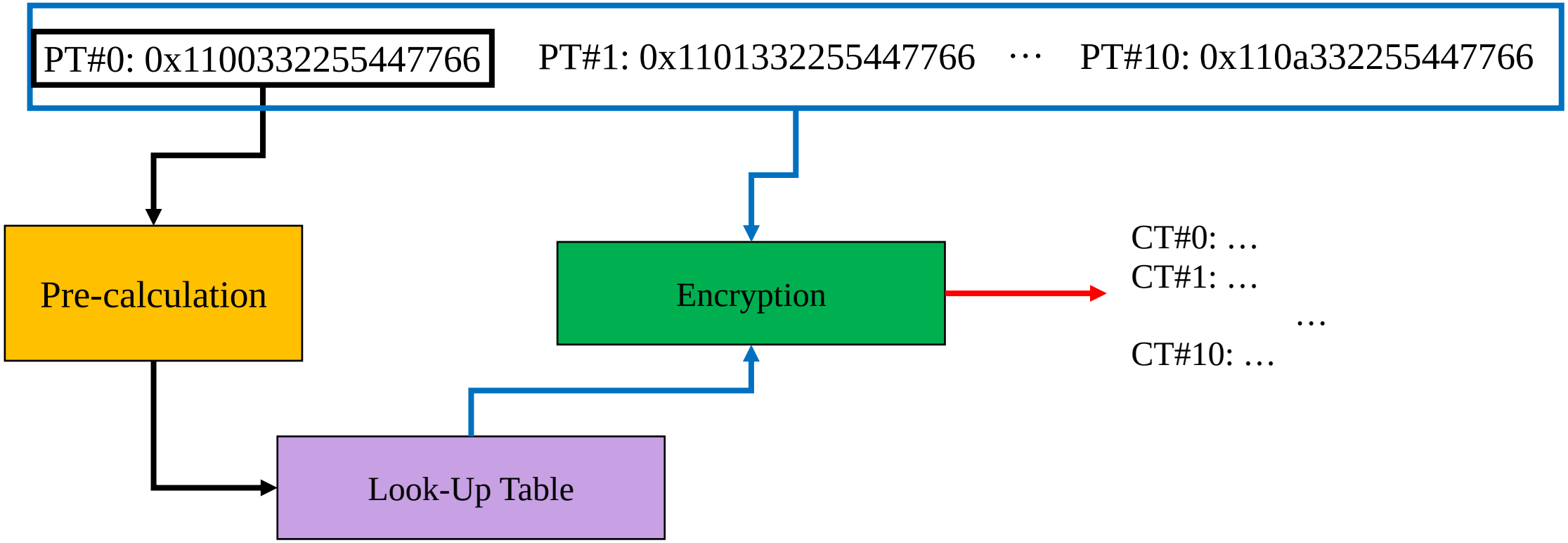


제안 기법

- 가변키(Variable-key) 시나리오를 구현
- **사전 테이블 연산**
- 두 가지 방법이 존재
  - 사전 테이블만 연산 ->분리형 모델(Separated model)
  - 사전 테이블을 연산하며 하나의 블록을 암호화 -> 동시형 모델(in online model)

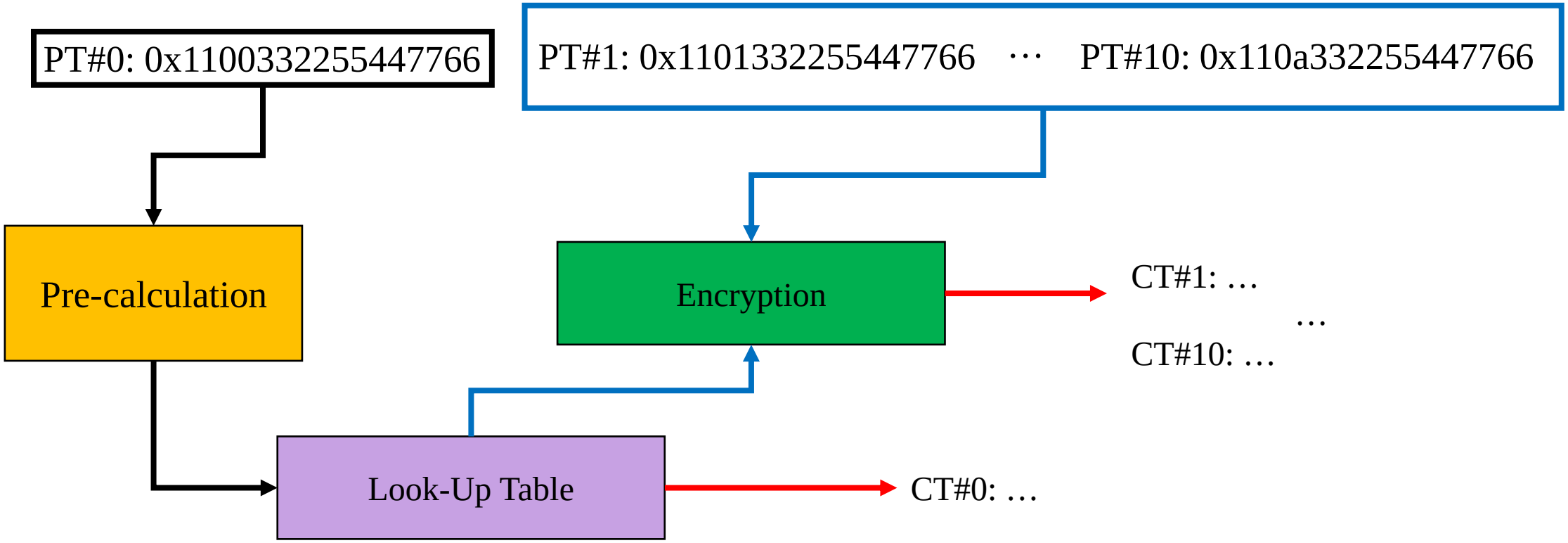
제안 기법

- 분리형 모델
- 첫 블록을 사용하여 **사전 연산만 진행**

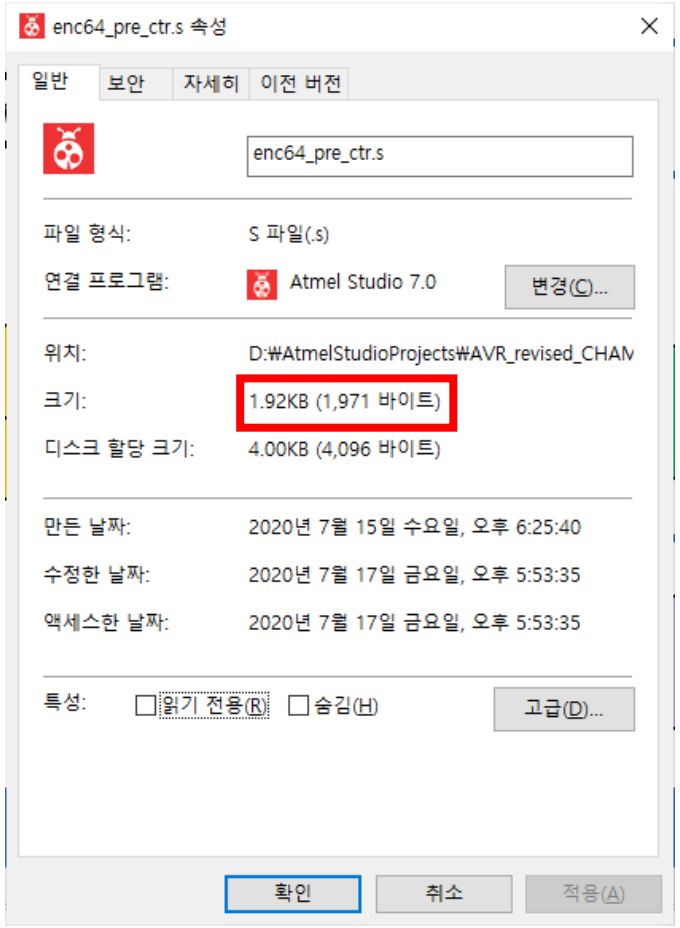


제안 기법

- 동시형 모델
- 첫 블록을 사용하여 사전 연산을 하며, **암호화도 진행**



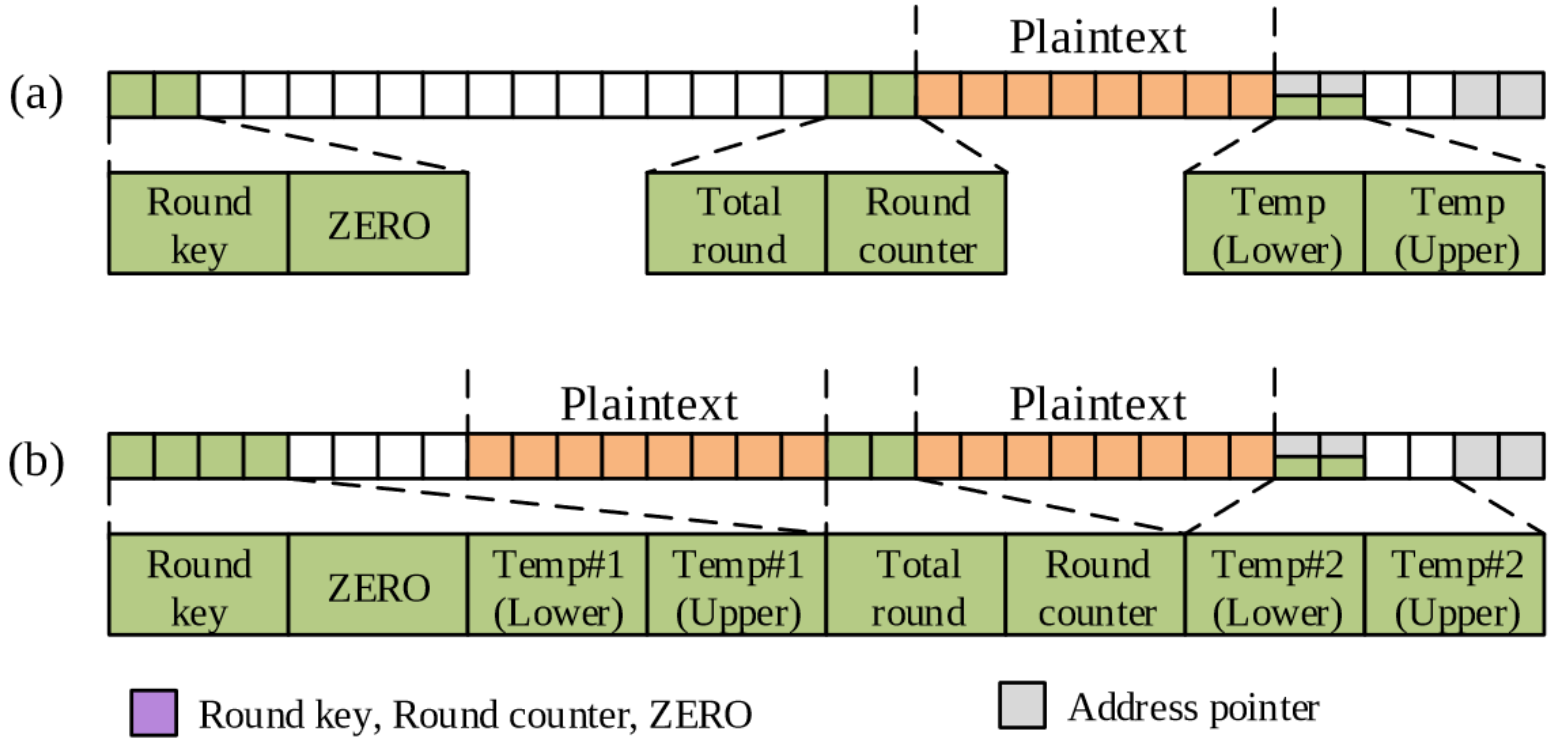
제안 기법



분리형 모델

동시형 모델

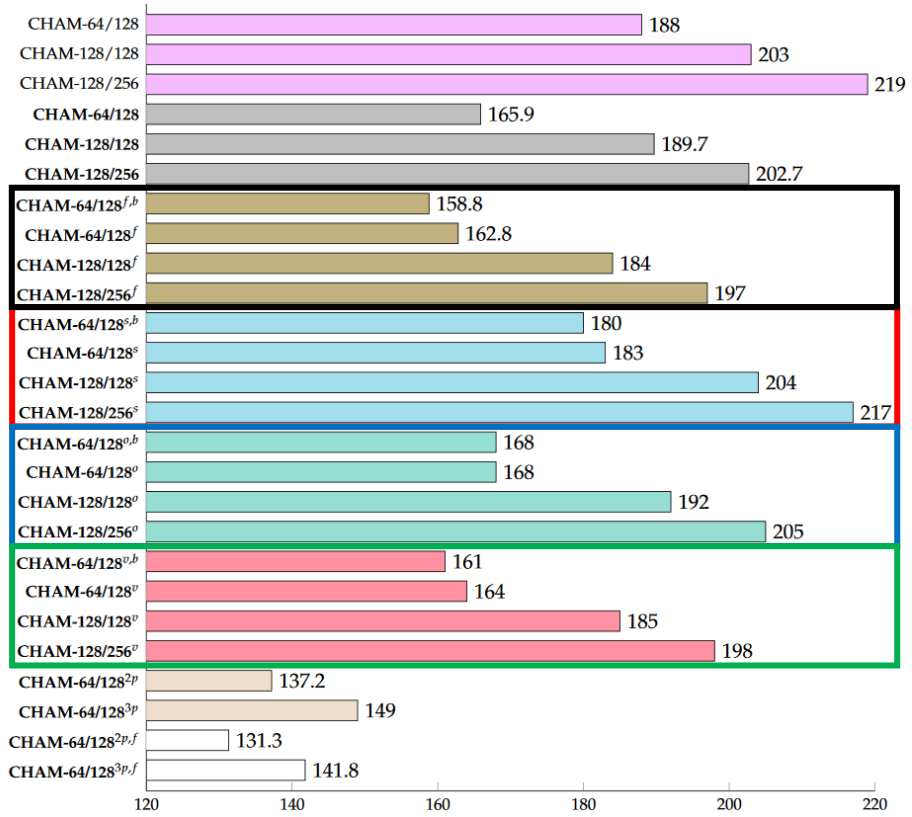
제안 기법



- 효율적인 레지스터 할당을 통해 최대 성능 발휘
- (a): CHAM-64/128 (b): CHAM-128/128, 128/256

성능 평가

- 기존 기법
- 분리형 모델
- 동시형 모델
- 사전 연산 활용



- 기존 기법보다 아주 적게 낮은 성능
  - LDI -> LD로 대체된 것의 영향
- 분리형 모델이 동시형 모델보다 빠름
  - 하지만, 2블록 이상 암호화 할 때는 느림

- 남는 레지스터를 활용한 CHAM-64/128 고속 구현