Fault attack resistance
오류 주입 공격 방어

https://youtu.be/Rc-WeW-yoFQ

Fault attack - 오류 주입 공격

- 계산 오류 – Data를 변경 시켜 오류를 일으키는 공격

Computation Faults: These faults cause errors in the data that is processed by a program. There is a trade-off between the accuracy by which an adversary can control the fault injection, and the required sophistication of a fault countermeasure that thwarts it. Therefore, we assume four computational fault models:

```
round key[0] = 003a0fd4 02497010 194f7db1 02497010 090d0883 02497010
round key[1] = 11fdcbb1 9e98e0c8 18b570cf 9e98e0c8 9dc53a79 9e98e0c8
round key[2] = f30f7bb5 6d6628db b74e5dad [illegible]
round key[3] = 74120631 dac9bd17 cd1ecf34 dac9bd17 540f76f1 dac9bd17
round key[4] = 662147db c637c47a 46518932 c637c47a 23269260 c637c47a
round key[5] = e4dd5047 f694285e e1c2951d f694285e 8ca5242c f694285e
round key[6] = baf8e5ca 3e936cd7 0fc7e5b1 3e936cd7 f1c8fa8c 3e936cd7
round key[7] = 5522b80c ee22ca78 8a6fa8b3 ee22ca78 65637b74 ee22ca78
round key[8] = 8a19279e 6fb40ffe 85c5f092 6fb40ffe 92cc9f25 6fb40ffe
round key[9] = 9dde584c cb00c87f 4780ad66 cb00c87f e61b5dcb cb00c87f
round key[10] = 4fa10466 f728e276 d255411b f728e276 656839ad f728e276
```

```
round key[0] = 003a0fd4 02497010 194f7db1 02497010 090d0883 02497010
round key[1] = 11fdcbb1 9e98e0c8 18b570cf 9e98e0c8 9dc53a79 9e98e0c8
round key[2] = [illegible] 23269260 6d6628db a65e46d0 6d6628db
round key[3] = 74120631 dac9bd17 cd1ecf34 dac9bd17 540f76f1 dac9bd17
round key[4] = 662147db c637c47a 46518932 c637c47a 23269260 c637c47a
round key[5] = e4dd5047 f694285e e1c2951d f694285e 8ca5242c f694285e
round key[6] = baf8e5ca 3e936cd7 0fc7e5b1 3e936cd7 f1c8fa8c 3e936cd7
round key[7] = 5522b80c ee22ca78 8a6fa8b3 ee22ca78 65637b74 ee22ca78
round key[8] = 8a19279e 6fb40ffe 85c5f092 6fb40ffe 92cc9f25 6fb40ffe
round key[9] = 9dde584c cb00c87f 4780ad66 cb00c87f e61b5dcb cb00c87f
round key[10] = 4fa10466 f728e276 d255411b f728e276 656839ad f728e276
```

Fault attack - 오류 주입 공격

- 계산 오류 – Data를 변경 시켜 오류를 일으키는 공격
  - Word / Byte / Bit / Bit Pair

  - **Random Word:** The adversary can target a specific word in a program and change its value into a random value unknown to the adversary.
  - **Random Byte:** The adversary can target a specific word in a program and change a single byte of it into a random value unknown to the adversary.
  - **Random Bit:** The adversary can target a specific word in a program and change a single bit of it into a random value unknown to the adversary.
  - **Chosen Bit Pair:** The adversary can target a chosen bit pair of a specific word in a program, and change it into a random value unknown to the adversary.

Fault attack - 오류 주입 공격

- 명령어 오류 – 명령어를 바꾸거나 스킵하는 오류 공격

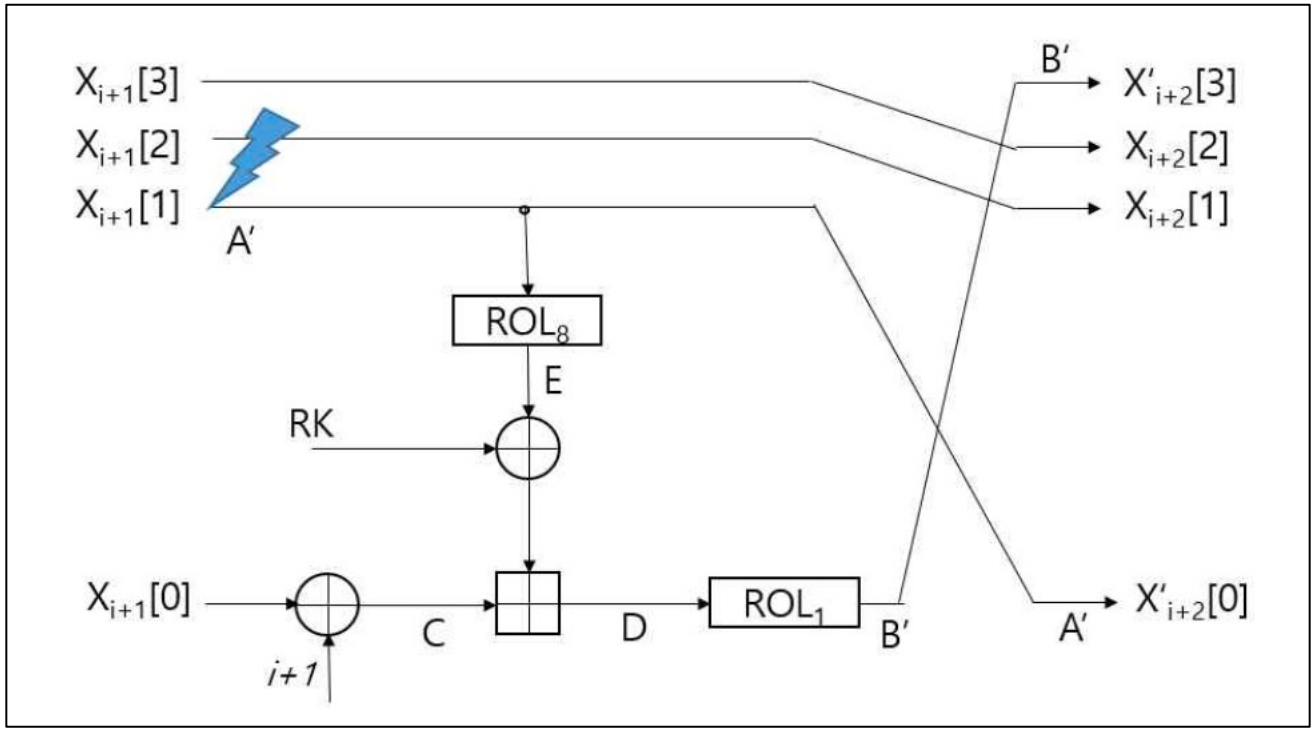
Instruction Faults: This fault model assumes that an attacker can change the opcode of an instruction by fault injection. A very common model is the *Instruction Skip* fault model, which replaces the opcode of an instruction with a **nop** instruction. Using this model, an attacker can skip the execution of a specific instruction in the program.

```
.macro Addroundkey
    lw          t2, 0(a1)
    lw          t3, 4(a1)
    addi        a1, a1, 8

    xor         a2, a2, t2
    srli        t2, t2, 8
    xor         a3, a3, t2
    srli        t2, t2, 8
    xor         a4, a4, t2
    srli        t2, t2, 8
    xor         a5, a5, t2

    xor         a6, a6, t3
    srli        t3, t3, 8
    xor         a7, a7, t3
    srli        t3, t3, 8
    xor         t0, t0, t3
    srli        t3, t3, 8
    xor         t1, t1, t3
.endm
```

```
.macro Addroundkey
    lw          t2, 0(a1)
    lw          t3, 4(a1)
    addi        a1, a1, 8

    xor         a2, a2, t2
    srli        t2, t2, 8
    xor         a3, a3, t2
    srli        t2, t2, 8
    xor         a4, a4, t2




    xor         a6, a6, t3
    srli        t3, t3, 8
    xor         a7, a7, t3
    srli        t3, t3, 8
    xor         t0, t0, t3
    srli        t3, t3, 8
    xor         t1, t1, t3
.endm
```

Fault attack – Computation Fault

- CHAM 알고리즘에 대한 오류 주입 공격



$$B = ROL_1(ROL_8(A) \oplus RK + C)$$

$$D = ROR_1(B) = ROL_8(A) \oplus RK + C$$
$$= E \oplus RK + C$$

$$D - C = E \oplus RK$$

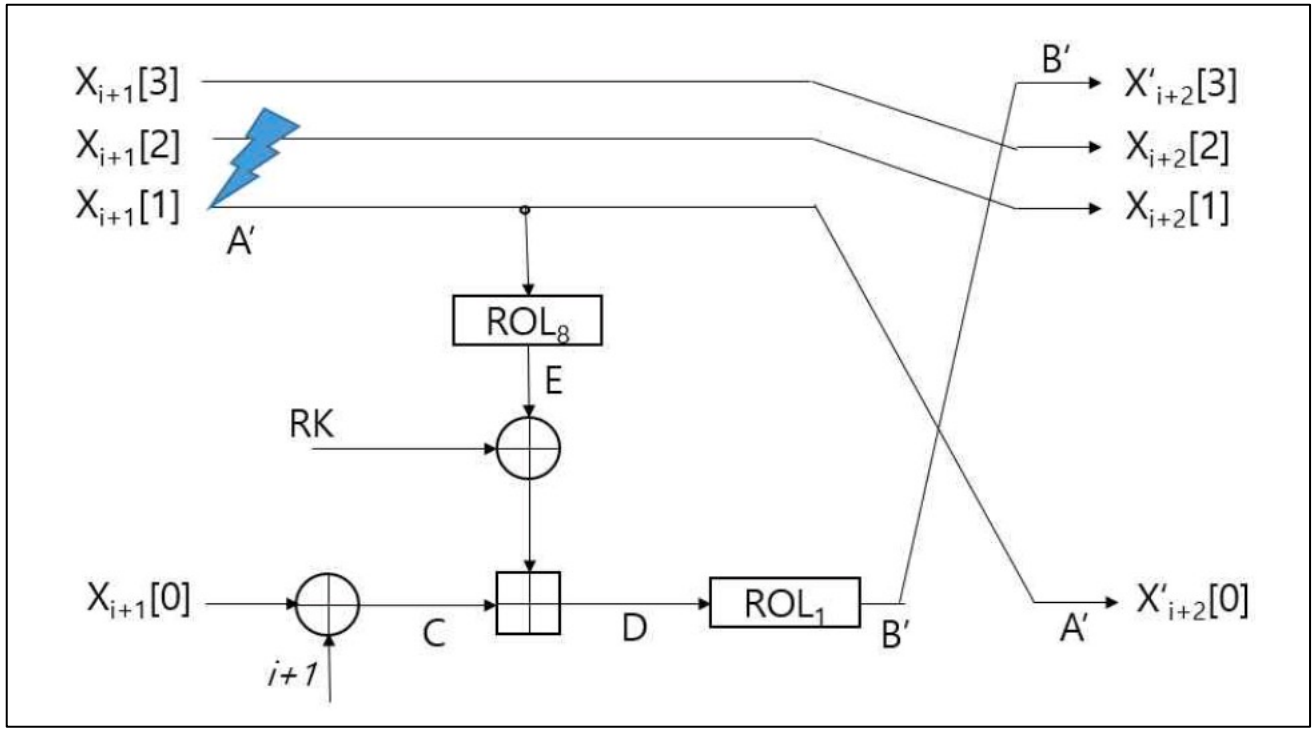
$$B' = ROL_1(ROL_8(A') \oplus RK + C)$$

$$D' = ROR_1(B') = ROL_8(A') \oplus RK + C$$
$$= E' \oplus RK + C$$

$$D' - C = E' \oplus RK \quad (2)$$

Fault attack – Computation Fault

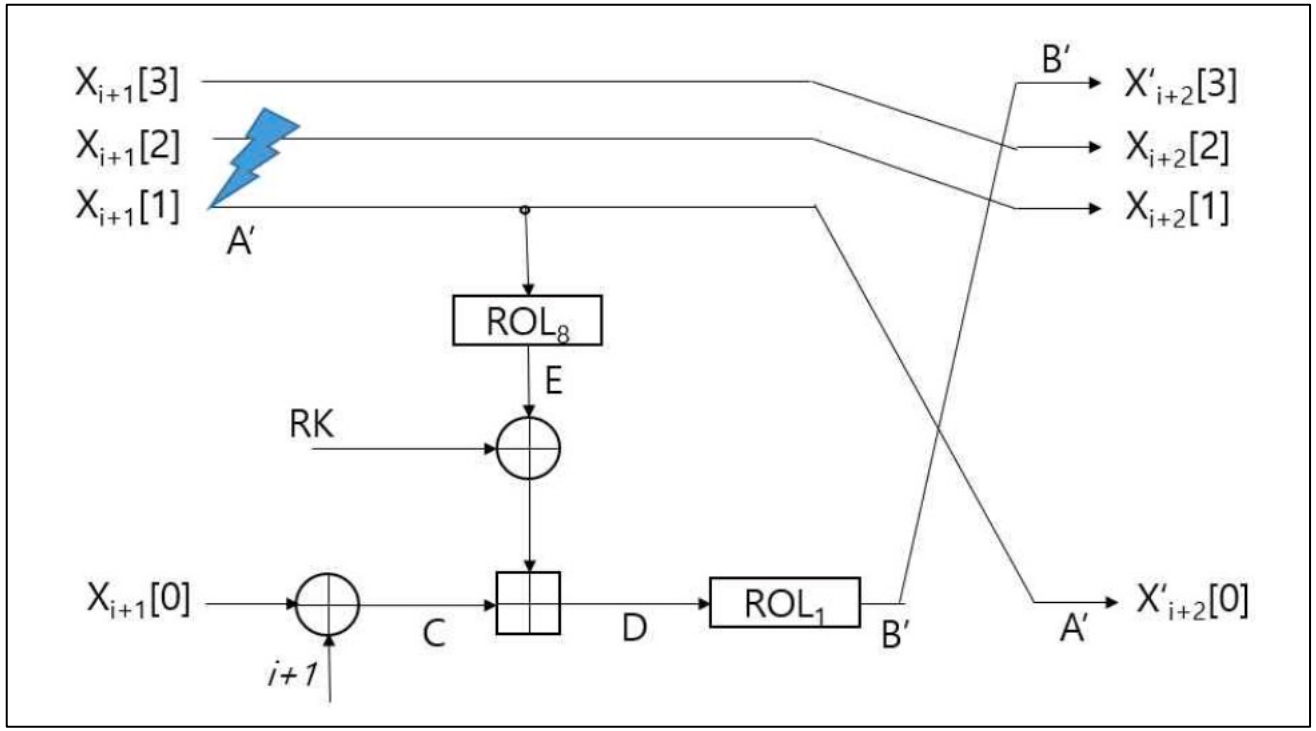
- CHAM 알고리즘에 대한 오류 주입 공격



$$K = E \oplus E' = (D - C) \oplus (D' - C)$$

Fault attack – Computation Fault

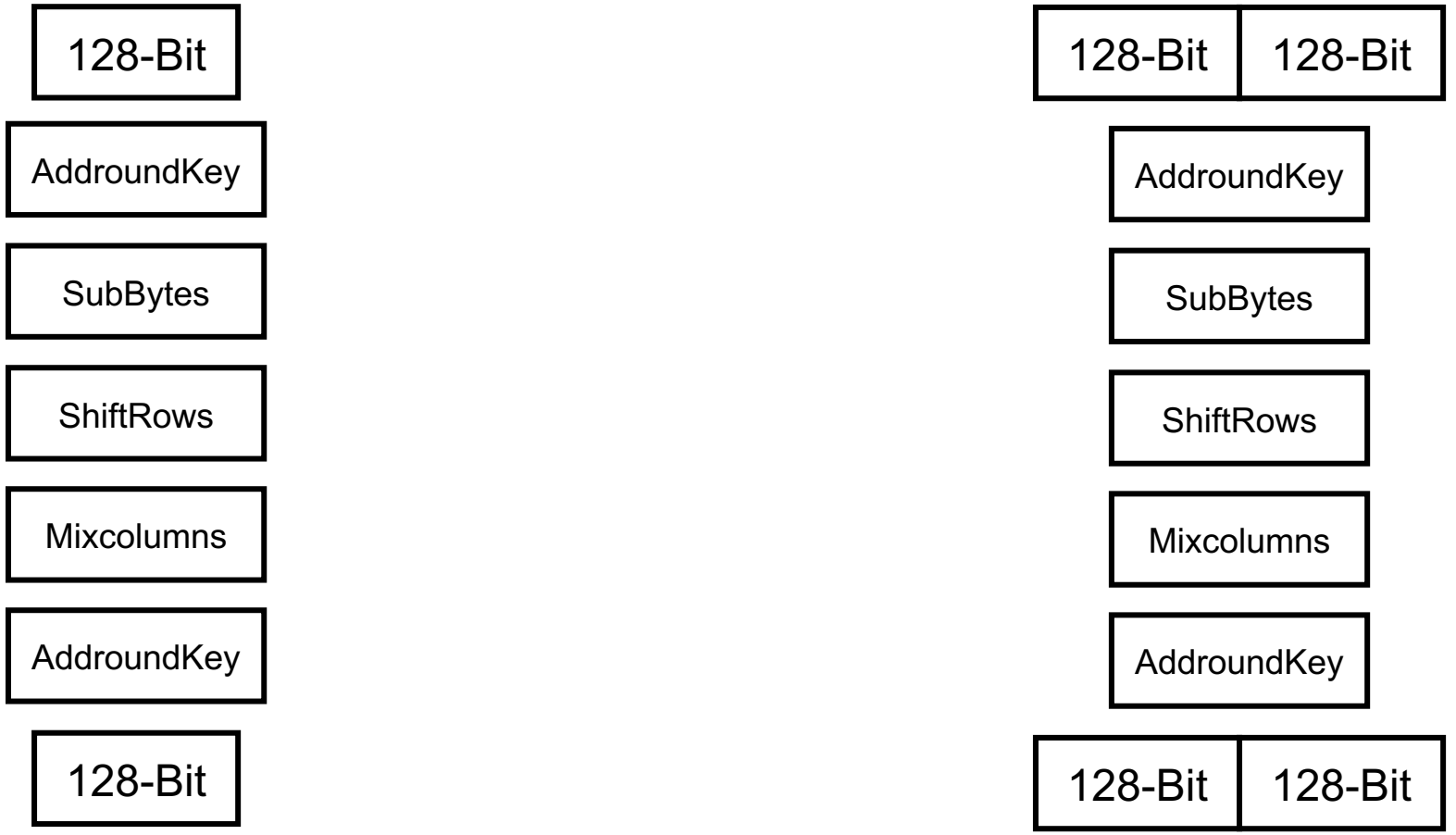
- CHAM 알고리즘에 대한 오류 주입 공격



$$RK = E \oplus (D - C)$$
$$= ROL_8(A) \oplus (ROR_1(B) - C)$$

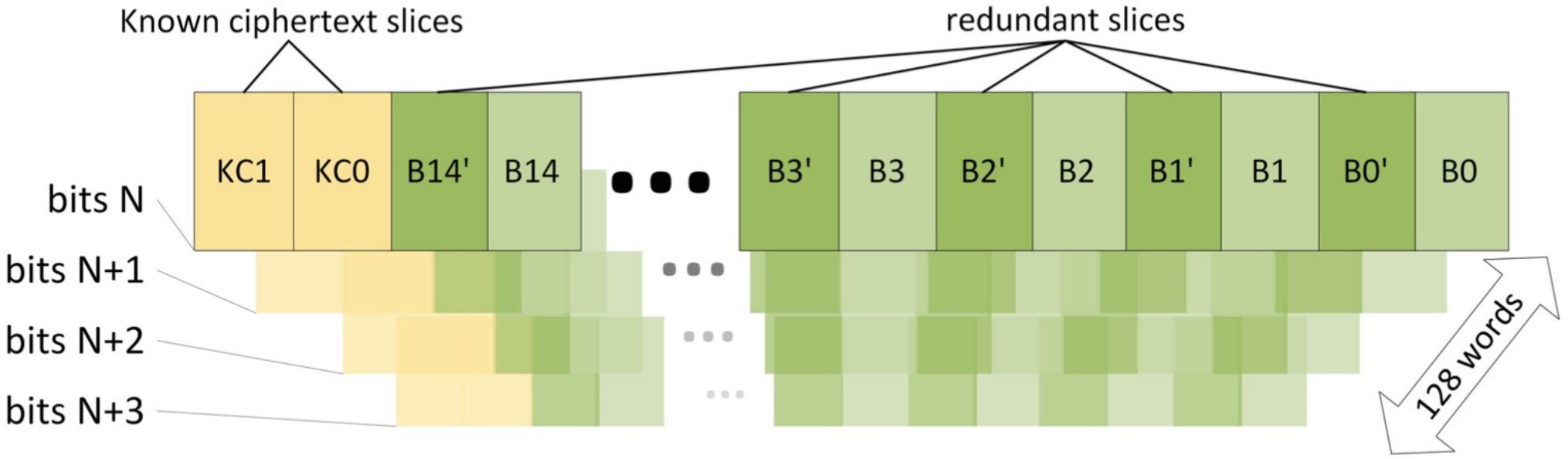
Fault attack resistance

- 오류 주입 공격을 탐지 하는 방법
  - 데이터를 복사하여 오류가 발생한 것을 탐지하는 방법



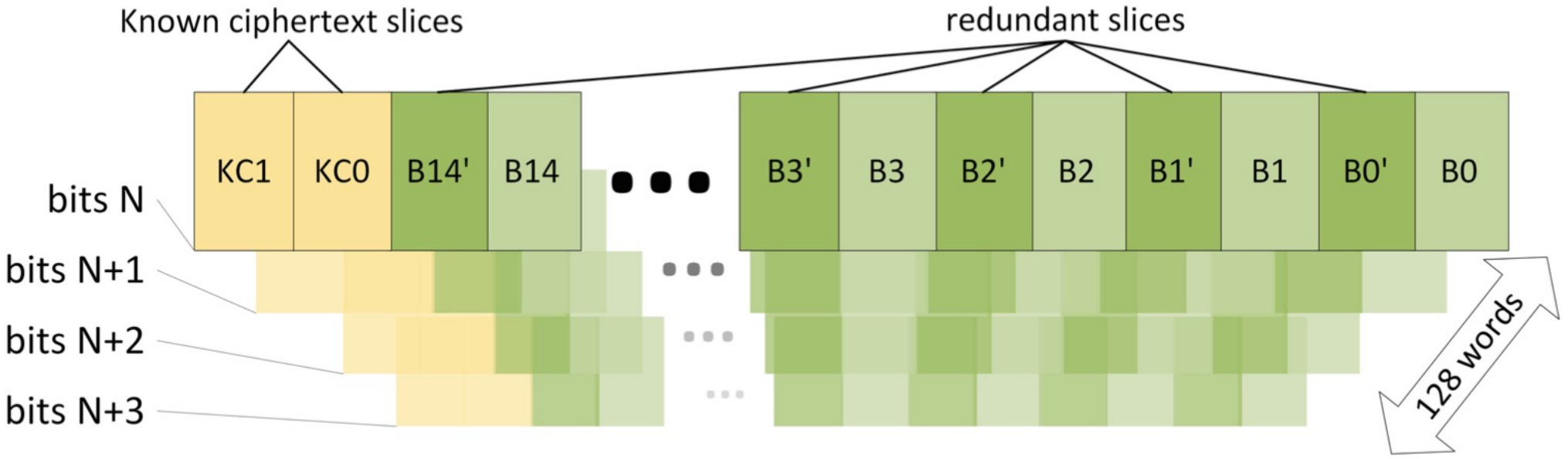
Fault attack resistance

- 오류 주입 공격을 탐지 하는 방법
  - 알려진 암호문을 추가로 넣어서 탐지하는 방법
  - 알려진 암호문은 입력으로 받은 키와는 다른 값으로 암호화를 진행함



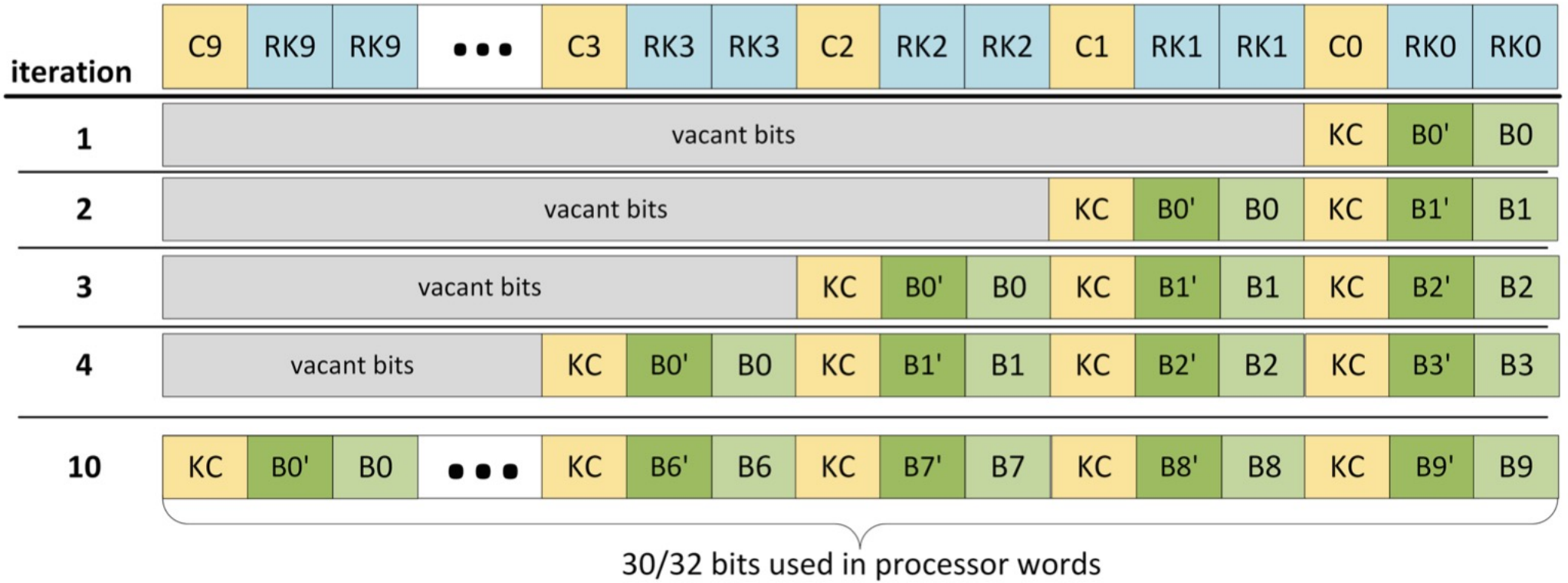
Fault attack resistance

- 오류 주입 공격을 탐지 하는 방법
  - 알려진 암호문을 추가로 넣어서 탐지하는 방법
  - 알려진 암호문은 입력으로 받은 키와는 다른 값으로 암호화를 진행함



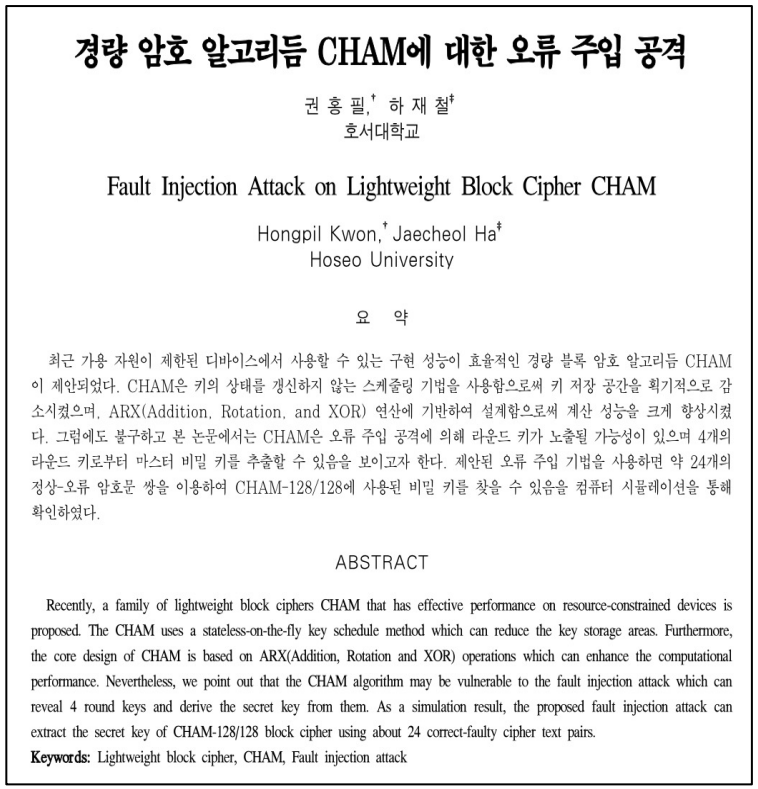
Fault attack resistance

- 오류 주입 공격을 탐지 하는 방법
  - Pipelined Intra-instruction Redundancy



참고자료

- 경량 암호 알고리듬 CHAM에 대한 오류 주입 공격
- Lightweight Fault Attack Resistance in Software Using Intra-instruction Redundancy
- Lightweight Fault Attack Resistance in Software Using Intra-instruction Redundancy, Revisited



경량 암호 알고리듬 CHAM에 대한 오류 주입 공격

권 홍 필,† 하 재 철‡
호서대학교

Fault Injection Attack on Lightweight Block Cipher CHAM

Hongpil Kwon,† Jaecheol Ha‡
Hoseo University

요 약

최근 가용 자원이 제한된 디바이스에서 사용할 수 있는 구현 성능이 효율적인 경량 블록 암호 알고리듬 CHAM이 제안되었다. CHAM은 키의 상태를 갱신하지 않는 스케줄링 기법을 사용함으로써 키 저장 공간을 획기적으로 감소시켰으며, ARX(Addition, Rotation, and XOR) 연산에 기반하여 설계함으로써 계산 성능을 크게 향상시켰다. 그럼에도 불구하고 본 논문에서는 CHAM은 오류 주입 공격에 의해 라운드 키가 노출될 가능성이 있으며 4개의 라운드 키로부터 마스터 비밀 키를 추출할 수 있음을 보이고자 한다. 제안된 오류 주입 기법을 사용하면 약 24개의 정상-오류 암호문 쌍을 이용하여 CHAM-128/128에 사용된 비밀 키를 찾을 수 있음을 컴퓨터 시뮬레이션을 통해 확인하였다.

ABSTRACT

Recently, a family of lightweight block ciphers CHAM that has effective performance on resource-constrained devices is proposed. The CHAM uses a stateless-on-the-fly key schedule method which can reduce the key storage areas. Furthermore, the core design of CHAM is based on ARX(Addition, Rotation and XOR) operations which can enhance the computational performance. Nevertheless, we point out that the CHAM algorithm may be vulnerable to the fault injection attack which can reveal 4 round keys and derive the secret key from them. As a simulation result, the proposed fault injection attack can extract the secret key of CHAM-128/128 block cipher using about 24 correct-faulty cipher text pairs.

Keywords: Lightweight block cipher, CHAM, Fault injection attack

Lightweight Fault Attack Resistance in Software Using Intra-instruction Redundancy

Conor Patrick(✉), Bilgiday Yuce, Nahid Farhady Ghalaty, and Patrick Schaumont

Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, USA
{conorpp,bilgiday,farhady,schaum}@vt.edu

Abstract. Fault attack countermeasures can be implemented by storing or computing sensitive data in redundant form, such that the faulty data can be detected and restored. We present a class of lightweight, portable software countermeasures for block ciphers. Our technique is based on redundant bit-slicing, and it is able to detect faults in the execution of a single instruction. In comparison to earlier techniques, we are able to intercept data faults as well as instruction sequence faults using a uniform technique. Our countermeasure thwarts precise bit-fault injections through pseudo-random shifts in the allocation of data bit-slices. We demonstrate our solution on a full AES design and confirm the claimed security protection through a detailed fault simulation for a 32-bit embedded processor. We also quantify the overhead of the proposed fault countermeasure, and find a minimal increase in footprint (14%), and a moderate performance overhead between 125% to 317%, depending on the desired level of fault-attack resistance.

Keywords: Fault attacks · Fault resistance · Intra-instruction redundancy · Bitslicing · Block ciphers

Lightweight Fault Attack Resistance in Software Using Intra-instruction Redundancy, Revisited

Hwajeong Seo[1], Taehwan Park[2], Janghyun Ji[2], and Howon Kim[2(✉)]

[1] IT Department, Hansung University, 116 Samseong Yoro-16gil, Seongbuk-gu, Seoul 136-792, Republic of Korea
hwajeong84@gmail.com
[2] School of Computer Science and Engineering, Pusan National University, San-30, Jangjeon-Dong, Geumjeong-Gu, Busan 609-735, Republic of Korea
{pth5804,jjh0819,howonkim}@pusan.ac.kr

Abstract. Fast implementations of block cipher is fundamental building block to achieve the high-speed and secure communication between IT platforms. Even though the communication is securely encrypted, the system can be exploited by malicious users if the attackers inject fault signal to the system and extract the user's secret information. For this reason, we need to ensure the high performance encryption together with secure countermeasures against side channel attacks. In this paper, we present a novel countermeasure against fault attack on Single Instruction Multiple Data (SIMD) architecture (e.g., ARM–NEON, INTEL–SSE, INTEL–AVX2). The methods achieved the fault attack resistance with intra-instruction redundancy feature in SIMD instruction set. Finally, we applied the new fault attack countermeasures on the block cipher LEA and achieved the intra-instruction redundancy and high performance over modern ARM-NEON architectures.

Keywords: Fault attack countermeasure
Intra-instruction redundancy · Block cipher · SIMD · LEA
ARM–NEON