격자기반 PQC 부채널 대응

송민호

유튜브: https://youtu.be/lIc9i8HAmSc

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB

부채널 분석

- 암호 알고리즘이 동작 시 발생하는 부가적인 정보를 가지고 비밀 키를 획득하는 공격
- 임베디드 환경에 취약
- 침입 공격, 준침입 공격, 비침입 공격으로 나누어질 수 있음

종류

- 침입 공격
  - 공격자가 물리적인 메모리에 접근하여 실제 비밀키의 정보를 획득하는 것

- 준침입 공격
  - 물리적인 장치에 오류를 주입하여 비밀키를 획득하는 것
  - 대표적으로 DFA(Differential Fault Attack)이 존재

- 비침입 공격
  - 전력, 전자파, 시간 차이 등의 정보를 이용하여 비밀키를 획득하는 것
  - 대표적으로 SPA(Simple Power Analysis), DPA(Differential Power Analysis), TA(Timing Attack)이 존재

Fault Attack

- 임베디드 환경에서 암호 알고리즘 동작 시 특정 지점에 오류 주입
  - ➢ 비밀 값 정보를 분석

- 오류 주입
  - 계산적: 암호 알고리즘 동작 시 특정 위치의 레지스터 내에 워드 값을 랜덤하게 변경(비트, 바이트, 워드 단위 다 가능)
  - 명령어: 오류 주입을 통해 특정 명령어를 건너뛰게 함.

Fault Attack 대응

- Dilithium같은 경우, 서명 생성 시 $z = y + cs$ 식이 필요
  - c = 챌린지 값, s = 비밀, $y(nonce)$ = 결정론적으로 계산된 값
  - c에 오류를 주입하여 값을 그대로 유지
  - ➢ 다른 c가 있는 동일한 메시지의 두 서명의 행렬 계산을 통해 s값 추출 가능

- Double computation
  - 알고리즘을 두 번 실행하여 동일한지 비교를 통해 오류 감지
  - ➢ 실행시간이 두 배로 늘어남.
  - ➢ 동일한 오류 주입 시 오류 감지 실패 가능성 존재

L.G. Bruinderink, P. Pessl, “Differential Fault Attacks on Deterministic Lattice Signatures”, CHES, 2018, 21-43.

Fault Attack 대응

- Verification-after-sign
  - 서명 후 서명을 확인
  - ➢ 두 번 서명 생성하는 것보단 실행시간이 효율적
  - ➢ 유효한 서명을 생성하기 위한 y의 샘플링에 삽입된 결함 감지 불가

- Additional randomness
  - 잡음 y에 솔트를 추가하여 값을 무작위함
  - ➢ 제한된 환경에서 사용 불가
  - ➢ Dilithium의 보안 위반

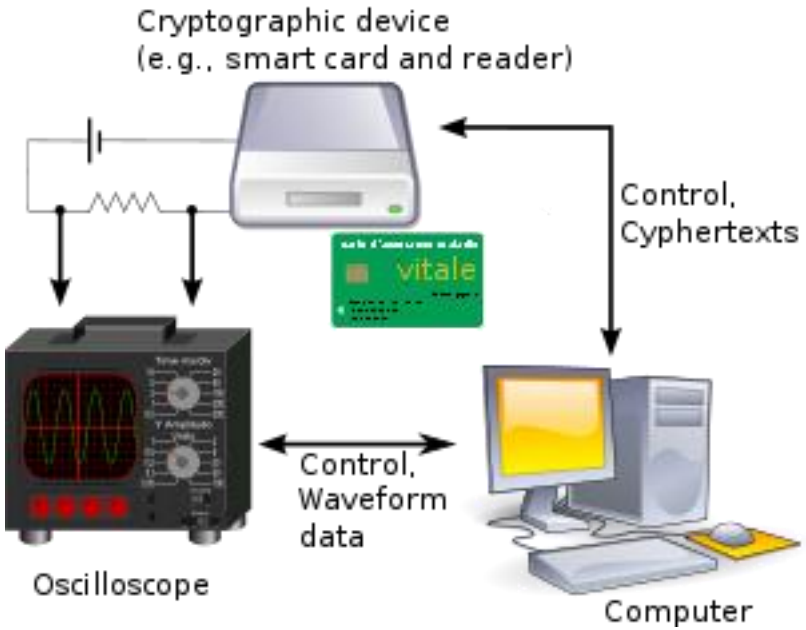
L.G. Bruinderink, P. Pessl, “Differential Fault Attacks on Deterministic Lattice Signatures”, CHES, 2018, 21-43.

Power Analysis

- 암호 알고리즘이 동작 시 발생하는 전력을 통해 비밀 값에 대한 정보를 분석하는 공격 기법

- 분석하는 파형의 수에 따라 분류
  - 단순: 소수의 전력 파형 분석을 통해 비밀 값 분석
  - 차분, 상관: 다수의 전력 파형 분석을 통해 평균의 차나 상관계수 분석을 통해 비밀 값 분석

- 비밀 키에 의존한 연산에 효과적



Power Analysis 대응

- 비밀키의 의존성을 제거하는 구현 필요
  - 비밀 값의 연산 자체를 수정하는 방법
  - 마스킹과 셔플링을 이용해 값을 숨기는 방법

- 입력 배열에 대한 셔플링 연산
  - 입력 배열의 순서를 무작위화함
  - ➢ 수집한 전력분석 파형을 예측하기 어렵게 함

- 피연산자의 값을 무작위하함
  - 컨볼루션의 피연산자의 배열을 초기에 제로가 아닌 무작위화함
  - ➢ 전력소모 패턴 예측하기 어렵게함

M.K. Lee, J. Song, D.H. Choi, D.G. Han. Countermeasures against Power Analysis Attacks for the NTRU Public Key Cryptosystem. IEICE Transactions. (2010)

Timing Attack

- 암호 알고리즘이 동작 시 발생하는 시간 차를 통해 비밀 값에 대한 정보를 분석하는 공격 기법

- 중간 연산 과정에서 발생하는 시간 차이 정보를 이용하여 분석
  - 중간 연산이 비밀 값에 의존하지 않는 Constant-time 구현이 아닐 경우 취약

- 캐시 타이밍 공격
  - 중간 값에 따라 캐시에 저장된 테이블에 접근확인을 통해 비밀키 분석
  - 격자 기반에서 테이블 조회 기반 샘플링인 경우 이 공격에 취약할 수 있음

Timing Attack 대응

- 비밀 값에 의존하지 않는 구현 기법 필요
  - Constant-time 구현
  - 중간 값을 랜덤하게 하여 시간 차 정보를 랜덤하게 하는 마스킹 기법

- 캐시 타이밍 공격 대응
  - 메모리 접근에 대한 셔플링 기법
  - 중간 값을 숨기는 마스킹 기법

Q & A