Trident: A Hybrid Correlation-Collision GPU Cache Timing Attack for AES Key Recovery 논문 리뷰

https://youtu.be/iI4EBuYuhrM

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB

Trident

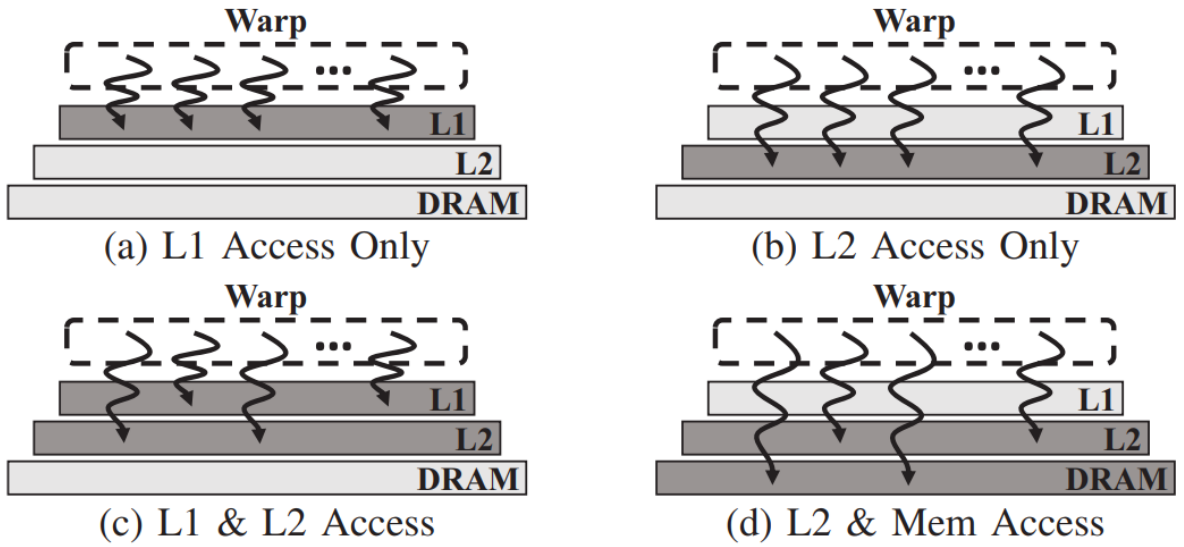
- 이전 연구에서 GPU SIMT 특성과 메모리 통합을 이용하여 AES(Advanced Encryption Standard) 사이드 채널 공격을 수행
- 그러나 최신 GPU에서 실현 가능하지 않음
- 최신 GPU에서 음의 타이밍 상관 관계가 발생할 수 있는 방법을 식별
- GPU에 대한 하이브리드 캐시 충돌 타이밍 공격인 Trident를 제안
  대기 시간 기반 대책인 TridentShield를 제안

AES

- 128비트 버전의 ECB 고려
- 4개의 룩업 테이블(T0-T3 테이블)을 사용하고 마지막 라운드는 단일 룩업 테이블(T4 테이블)을 포함합니다. 마지막 라운드 키와 비트 XOR이 뒤따릅니다. AES의 마지막 라운드는 이 작업에서 고려
- GPU가 수신한 일반 텍스트(즉, 512바이트)는 워프 내의 스레드 간에 분할되고 각 스레드는 병렬로 암호화를 수행
- 마지막 라운드 키 바이트 복구

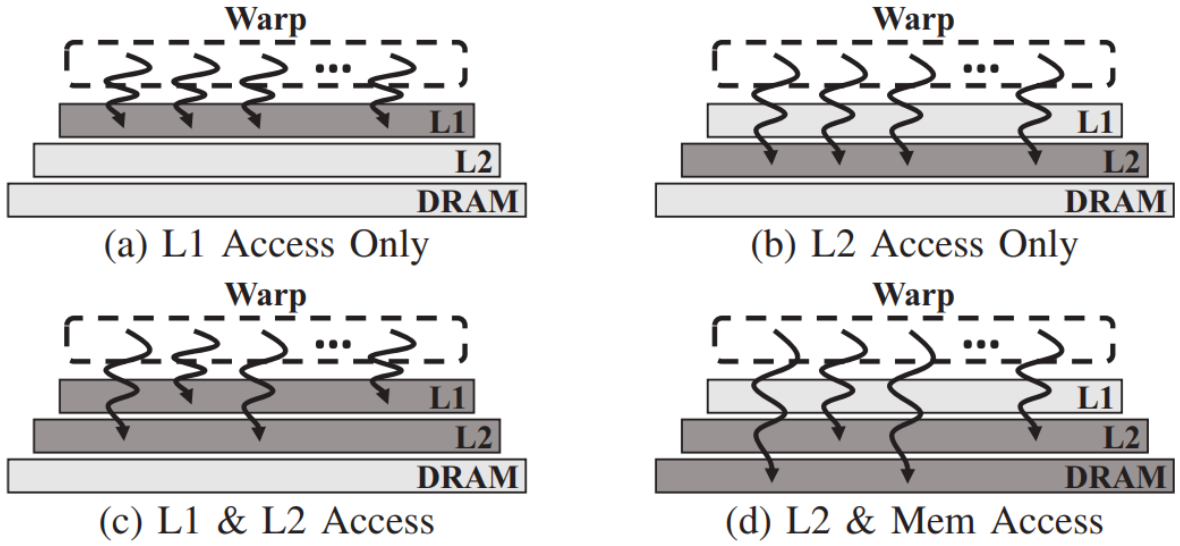
GPU Architecture

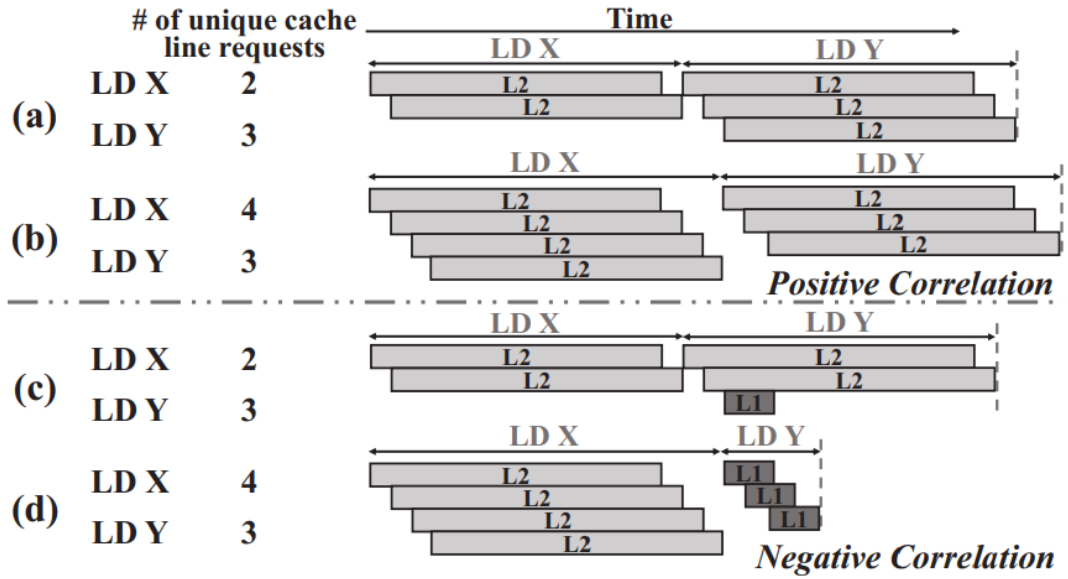
- 메모리 요청 수를 줄이기 위해 최신 GPU는 여러 메모리 요청을 단일 메모리 트랜잭션으로 병합하거나 결합
- 단일 워프의 모든 스레드를 단일 메모리 트랜잭션으로 병합할 수 있지만 최악의 경우 병합을 수행할 수 없으며 결과적으로 32개의 스레드에 대해 32개의 개별 메모리 트랜잭션이 발생
- 병합의 양은 전체 성능 또는 실행 시간에 영향을 미치며 이전 작업은 GPU 부채널 공격에서 이 동작을 악용



SIMT Leakage Side-channel Attack

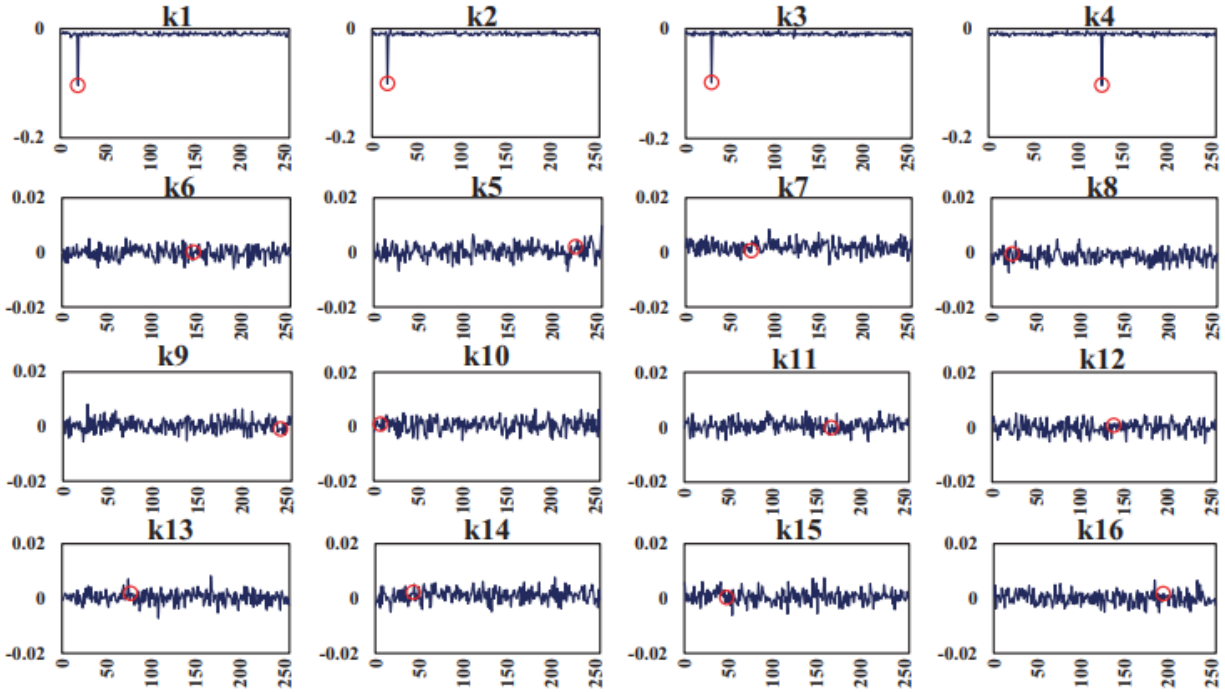
- GPU와 같은 SIMT 아키텍처에서 워프의 읽기 요청은 메모리 대역폭을 최대화하기 위해 결합되거나 병합
- 병합된 요청은 메모리 문제 직렬화 또는 쓰기 되돌림 직렬화에 의해 주기당 하나씩 처리
- 고유한 캐시 라인 요청의 수와 실행 시간은 양의 상관관계가 있으며 이를 SIMT 누출
- 공격자는 키 바이트 추측에 따라 고유한 캐시 라인 2 요청 수를 결정



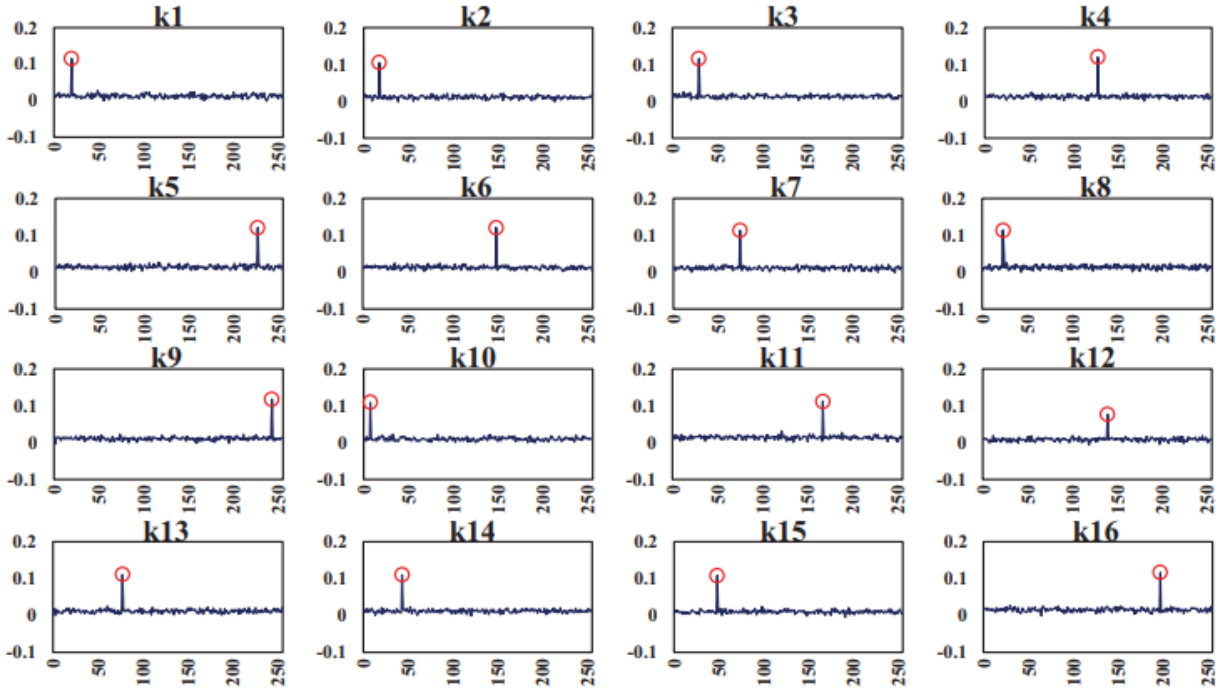


SIMT Leakage Re-visited

- Nvidia Tesla K40 (Kepler),
- GeForce GTX960 (Maxwell), GTX1060 (Pascal) and Tesla V100 (Volta).



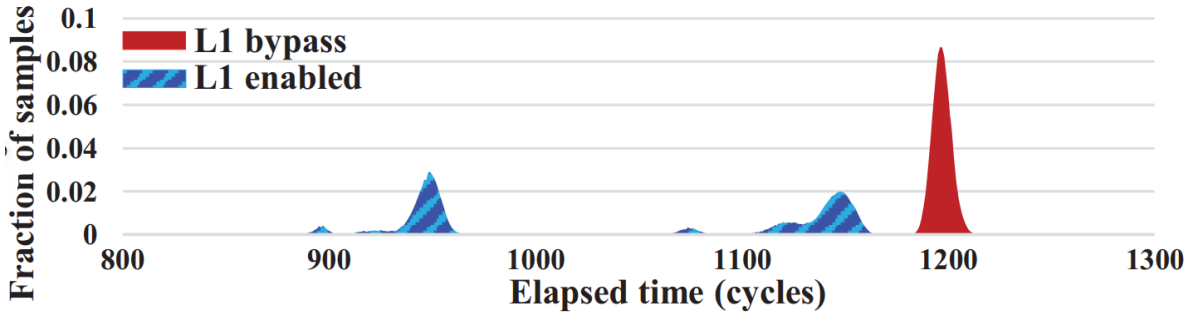
(a) L1 enabled

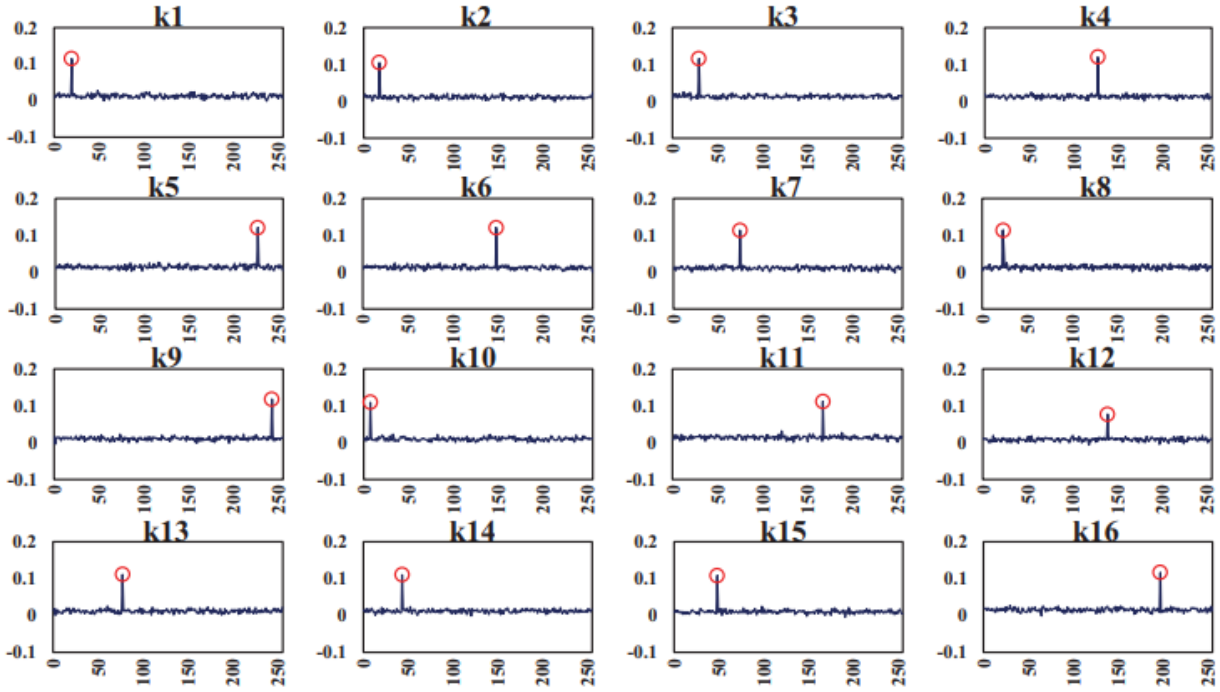


(b) L1 bypass

SIMT Leakage Re-visited

- 최신 GPU에는 몇 MB의 L2 캐시가 있고 AES에 사용되는 T-테이블은 상대적으로 작기 때문에(약 5kB) L2 캐시는 호스트의 데이터 복사본에서 효과적으로 워밍업
- 결과적으로 T-테이블에 대한 초기 메모리 액세스는 주 메모리에 액세스하지 않고 L2 캐시에서 액세스
- L1 캐싱이 활성화되면 SIMT 누출을 기반으로 하는 상관 관계 공격이 더 이상 불가능

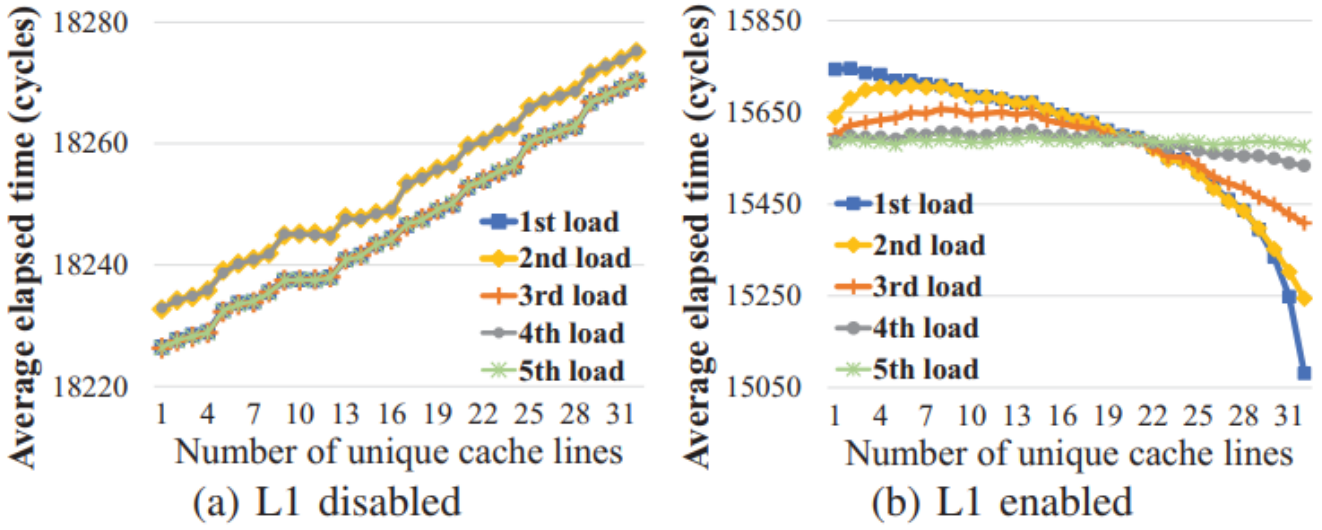


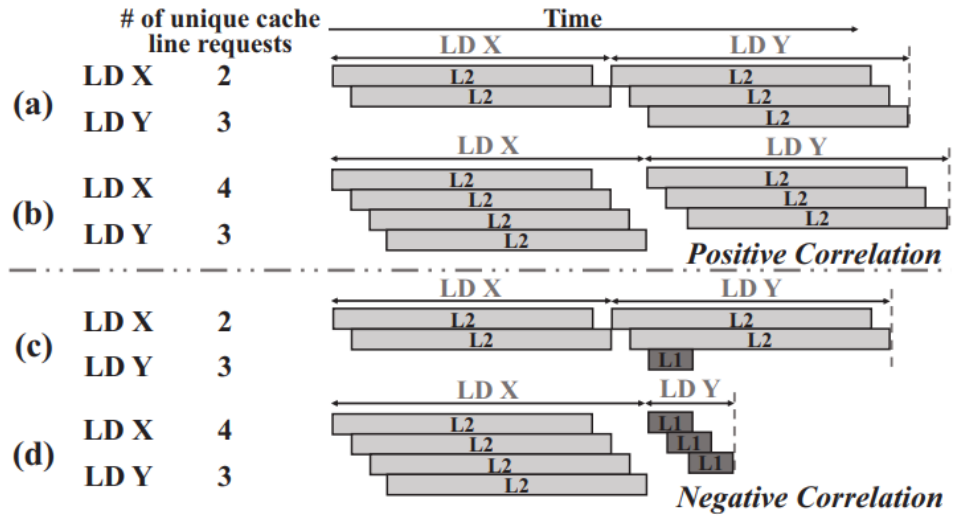


(b) L1 bypass

SIMT Leakage Re-visited

- 16개의 로드가 있는 합성 CUDA 커널을 만들어 AES의 마지막 라운드 동안 메모리 액세스를 평가
- Volta GPU에서 글로벌 로드의 L1 캐싱이 있거나 없는 1000개 샘플의 실행 시간을 평균화





Probabilistic Model

- 캐시 액세스 입도가 음의 상관 관계에 미치는 영향을 분석하는 확률 모델
- 확률 모델의 목표는 전체 T4 테이블이 L1에 얼마나 빨리 로드되는지 분석하는 것
- T4 테이블이 L1에 느린 속도로 로드되는 경우 L1 및 L2 액세스 패턴이 발생하여 음의 상관 관계를 유발

w = # of unique cache lines in the T4-table
m = avg # of unique cache line requests for each load
X_N = # of unique cache lines in L1 after N^{th} load
Y_N = `L1 Access Only` event occurring during N^{th} load

$$P(X_1 = k) = \begin{cases} 1, & \text{if } k = m \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

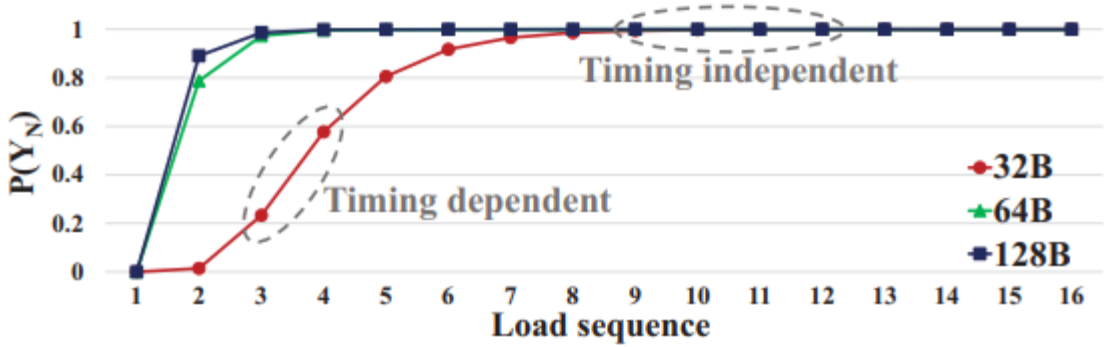


Fig. 7: `L1 Access Only` probability ($P(Y_N)$) for different cache line sizes.

$$P(X_N = k) = \sum_{i=m}^{k} P(\{X_N = k\} \cap \{X_{N-1} = i\}) \quad (2)$$

$$= \sum_{i=m}^{k} P(X_{N-1} = i) \cdot P(X_N = k \mid X_{N-1} = i) \quad (3)$$

$$= \sum_{i=m}^{k} P(X_{N-1} = i) \cdot \frac{\binom{w-i}{k-i} \cdot \binom{i}{m-(k-i)}}{\binom{w}{m}} \quad (4)$$

$$P(Y_N) = \sum_{i=m}^{w} P(\{Y_N\} \cap \{X_{N-1} = i\}) \quad (5)$$

$$= \sum_{i=m}^{w} P(X_{N-1} = i) \cdot P(Y_N \mid X_{N-1} = i) \quad (6)$$

$$= \sum_{i=m}^{w} P(X_{N-1} = i) \cdot P(X_N = i \mid X_{N-1} = i) \quad (7)$$

$$= \sum_{i=m}^{w} P(X_{N-1} = i) \cdot \frac{\binom{i}{m} \cdot \binom{w-i}{0}}{\binom{w}{m}} \quad (8)$$

Limitations of Correlation Timing Attack

- 나머지 키 바이트가 음의 상관 관계를 사용하여 복구할 수 없음
- 첫 번째 - 네 번째 키 바이트에 대한 음의 상관 관계를 통해 이전 키 바이트를 복구할 수 있습니다.
- 이 정보를 사용하여 모든 5번째 후보 키 바이트 값(0 - 255)에 대해 타이밍 정보를 기반으로 5번째 로드 명령에 의해 생성된 요청이 L1 Access Only인지 !(L1 Access Only)인지 구별
- Trident는 모든 타이밍 샘플을 L1 액세스 전용 및 !(L1 액세스 전용)의 두 세트로 분류
- 이 두 세트의 실행 시간 차이가 가장 클 때 키 바이트의 정확한 추측이 결정될 수 있으므로 올바르게 식별 가능

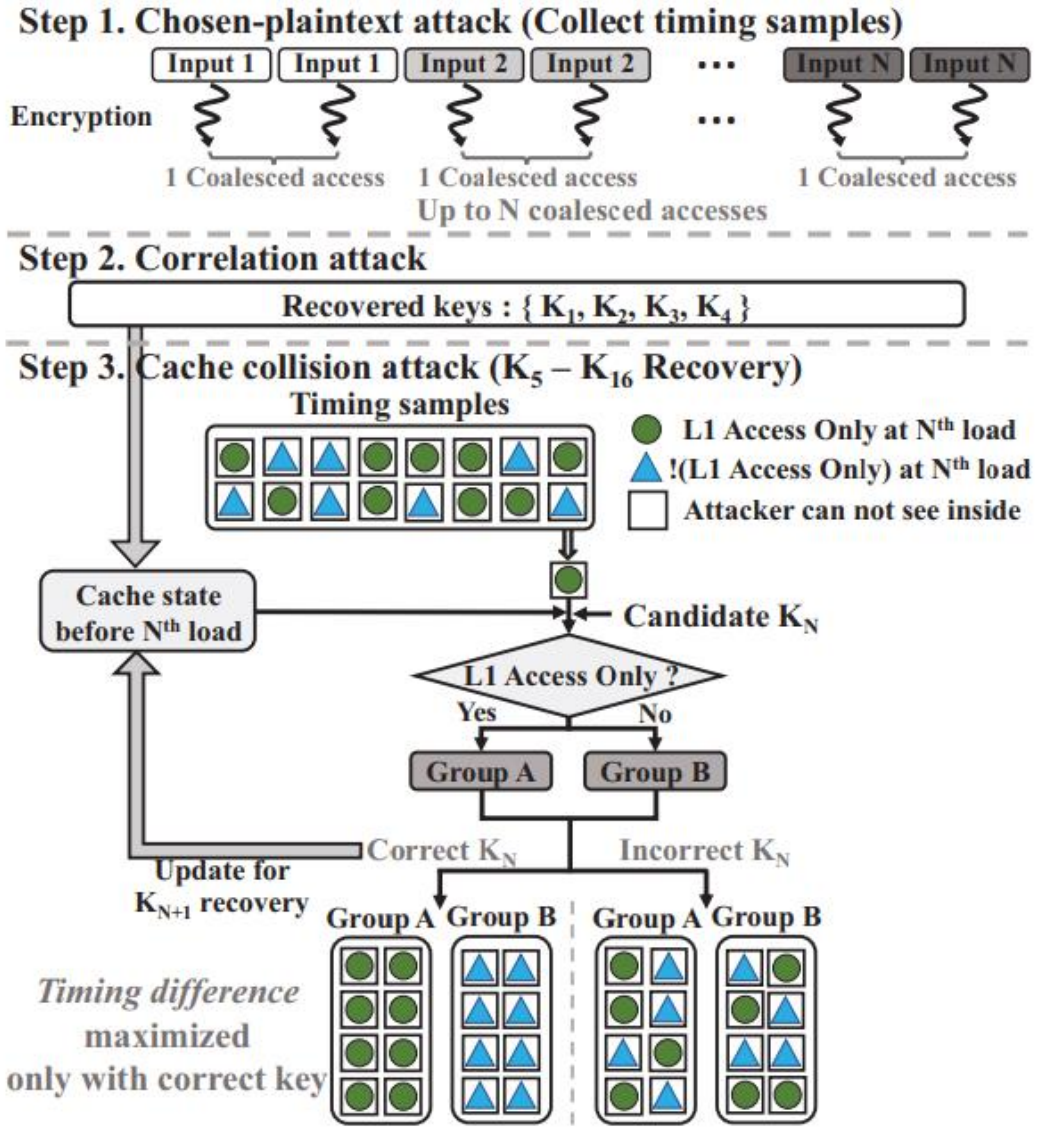


Fig. 8: Trident attack high-level overview diagram.

Chosen Plaintext Attack

- 상당한 양의 타이밍 샘플을 수집
- 동일한 암호문을 생성하고 동일한 T4 테이블 요소에 액세스
- 결과적으로 동일한 일반 텍스트를 사용하는 다른 스레드의 T4 테이블 읽기 요청이 하나의 요청으로 병합
- 4개의 스레드 그룹에 동일한 일반 텍스트가 제공되므로 32-스레드 워프를 8-스레드 워프로 효과적으로 줄이고 대신 최대 8개의 메모리 액세스만 발생
- 32개의 메모리 액세스 중 selected-plaintext 공격을 통해 공격자는 생성되는 고유한 캐시 라인 요청의 수와 그림 11과 같이 L1 캐시가 T4 테이블과 함께 로드되는 속도에 미치는 영향을 제어
- !(L1 액세스 전용) 샘플을 쉽게 생성

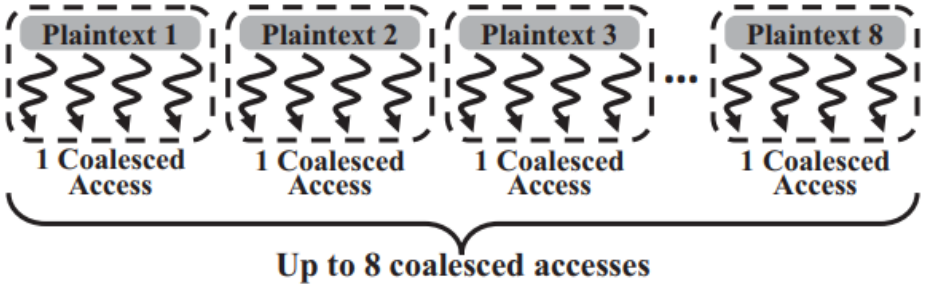


Fig. 10: 8-thread chosen-plaintext attack in *Trident*.

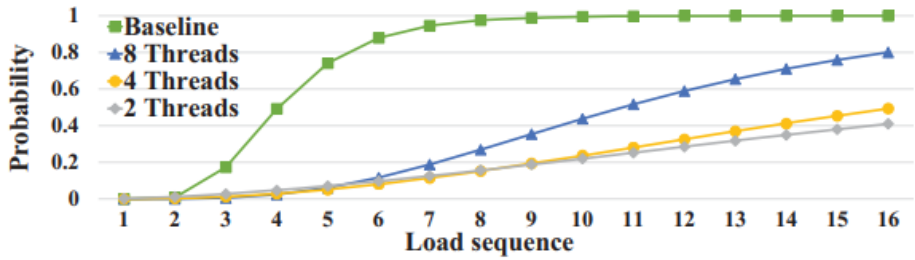


Fig. 11: Probability that `L1 Access Only` occurs for the 2, 4, 8 thread model. Baseline has 32 threads per warp.

Evaluation of Trident Attack

- Trident 공격은 100만 개의 샘플이 필요한 상관 관계 공격에 비해 단 100,000개의 샘플을 사용하여 마지막 라운드 키의 16바이트를 모두 복구
- 4개의 키 바이트는 음의 상관 관계를 사용하여 복구
- k5 ~ k12의 경우 키 바이트가 가장 낮은 값에서 복구되고 나머지 키 바이트의 경우 키 바이트가 가장 높은 값에서 복구

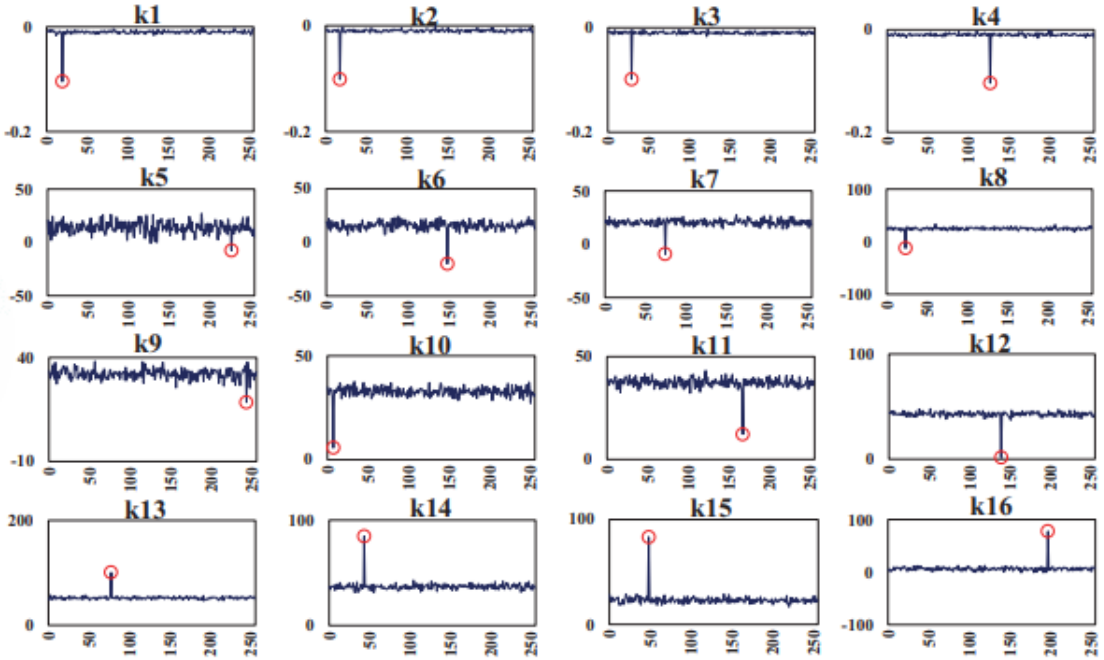


Fig. 12: AES key recovery on Volta GPU with 100,000 samples using 8 thread model under *clean* measurements.

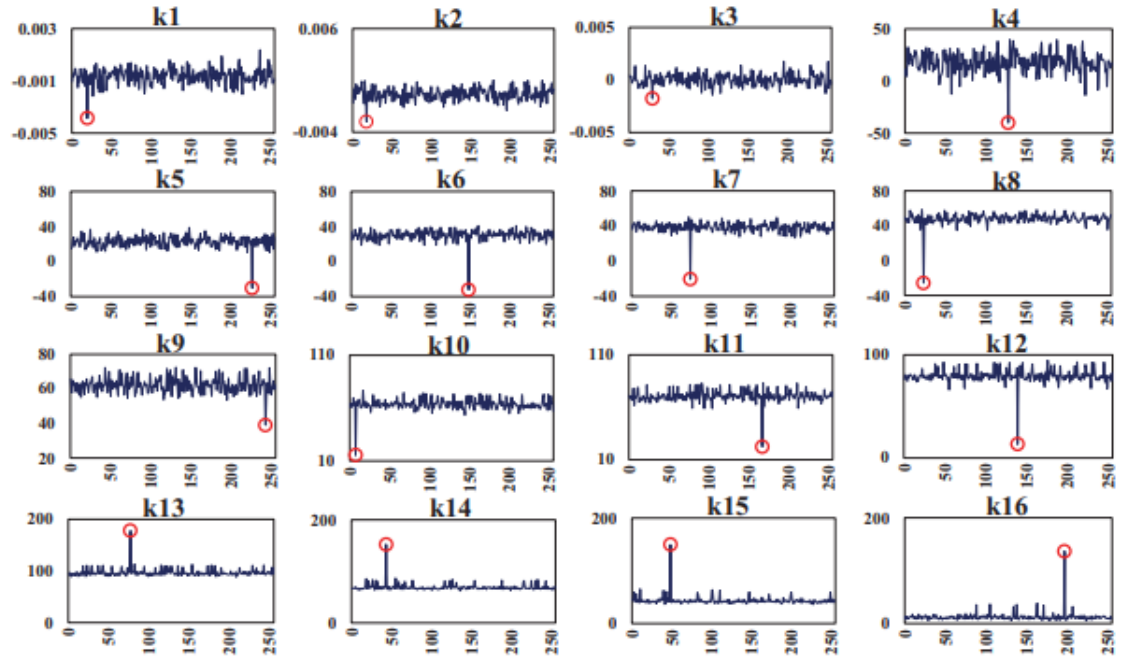


Fig. 13: AES key recovery on Volta GPU with 2 million samples using 8-thread model under *noisy* measurements.

Q & A