NVIDIA GPU의 텐서 코어를 활용한 암호 구현 예시 테스트 및 분석

융합보안학과 윤세영
유투브 주소: https://youtu.be/2S0Onr5LFFo

목차

동향 파악

TensorCrypto: High Throughput Acceleration of Lattice-based Cryptography Using Tensor Core on GPU

WAI-KONG LEE[1], (Member, IEEE), HWAJEONG SEO[2], (Member, IEEE), ZHENFEI ZHANG [3], and SEONG OUN HWANG.[1], (Senior Member, IEEE)

[1]Department of Computer Engineering, Gachon University, South Korea. (e-mail: waikonglee, sohwang@gachon.ac.kr)
[2]College of IT Engineering at Hansung University, Seoul, South Korea.
[3]Ethereum Foundation.

Corresponding author: Seong Oun Hwang (e-mail: sohwang@gachon.ac.kr).

- 텐서 코어를 기반으로 한 NTRU 구현을 제안
- *NTRU: 격자 기반 암호화를 사용하여 데이터를 암호화하고 해독하는 오픈 소스 공개 키 암호 시스템
- Tensor 코어는 NVIDIA GPU 칩에 새로 도입된 하드웨어
- 장치의 정수 및 부동 소수점 단위보다 행렬 곱셈을 훨씬 빠르게 계산
- 텐서 코어를 사용하여 격자 기반 암호를 가속화할 수 있음 ->
- 격자 기반 암호 시스템에서 시간이 오래 걸리는 다항식 컨볼루션의 속도를 높이기 위해 텐서 코어를 사용
- 텐서 코어를 사용하면 기존 정수 단위로 구현된 버전보다 최소 2배 더 빠름
- NTRU의 다항식 차수는 16의 배수가 아니므로 텐서 코어 기반 다항식 컨볼루션을 사용하려면 수정 필요함 (제로 패딩, sign conversion , type casting)

DPCrypto: Acceleration of Post-Quantum Cryptography Using Dot-Product Instructions on GPUs

Wai-Kong Lee, *Member, IEEE*, Hwajeong Seo, *Member, IEEE*, Seong Oun Hwang, *Senior Member, IEEE*, Ramachandra Achar, *Fellow, IEEE*, Angshuman Karmakar, and Jose Maria Bermudo Mera

- "The dot-product instruction" 을 사용하여 행렬 곱셈 및 다항식 컨볼루션 연산을 가속화할 수 있음을 보여줌
- FrodoKEM (격자기반암호화?) 최적화 구현 제공
- 제안된 FrodoKEM 구현은 V100 GPU의 구현보다 4.37배 더 높은 처리량을 달성함
- 행렬 곱셈과 다항식 컨볼루션 연산은 격자 기반 암호화 방식에서 가장 시간이 많이 걸리는 연산임
- 따라서 제안된 방법이 격자 기반의 다른 KEM 및 서명 방식에 도움이 될 수 있을 것

TensorFHE: Achieving Practical Computation on Encrypted Data Using GPGPU

Shengyu Fan*‡, Zhiwei Wang*†, Weizhi Xu‡, Rui Hou*†, Dan Meng*†, Mingzhe Zhang*
* State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China.
† School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China.
‡ School of Information Science and Engineering, Shandong Normal University, Jinan, China.
damionfan@163.com, {wangzhiwei, hourui, mengdan, zhangmingzhe}@iie.ac.cn, xuweizhi@sdnu.edu.cn

- FHE : 완전 동형 암호화 (Fully Homomorphic Encryption)
- FHE는 신뢰할 수 없는 서버에서 개인 정보를 보호하는 연산을 가능하게 하므로 보안적인 측면에서 좋은 해결책으로 간주됨
- 그러나 성능 오버헤드로 연산 시간이 오래 걸려 광범위한 사용이 불가
- 본 논문에서 제안하는 TensorFHE는 TCU(Tensor Core Unit)를 활용하여 (FHE의 일부인) NTT 연산 능력을 향상시킴
- TensorFHE는 하나의 작업 시간을 단축시키기 보다는 특정 시간에 가능한 한 많은 FHE 작업을 수행하는 데 중점을 뒀음
- 실제 시스템에 해당 FHE 알고리즘을 적용하여 가속화할 수 있음

TESLAC: Accelerating Lattice-Based Cryptography with AI Accelerator

Lipeng Wan[1,2,3], Fangyu Zheng[1,3(✉)], and Jingqiang Lin[4]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
zhengfangyu@iie.ac.cn

- NVIDIA Tensor Core를 기반으로 TESLAC이라는 프로토타입 시스템을 구현
- 텐서코어를 이용한 격자 기반 암호화(Lattice-Based Cryptography) 구현
- 지연 시간이 짧은 공유 메모리를 사용하고, 메모리 액세스를 통합하는 등의 변환 오버헤드를 최소화하기 위한 추가적인 최적화도 함께 했음
- 계산의 정확성과 단순성을 고려하여 Tesla V100의 기술을 사용하여 NIST PQC에서 선택한 LAC를 구현하기로 결정했다고 함
- Tensor Core의 동작 모드에 적응하기 위해 "polynomial multiplication over rings"을 위한 벡터 확장 방법을 제시함

동향 파악

- 텐서코어를 이용한 암호 구현 중에서 격자 기반 암호 구현이 많은 듯
- 텐서코어(AI 가속기) 같은 경우에는 전체 연산 속도를 줄인다기 보다 특정 부분의 연산 (행렬 곱 등)을 줄이는 데 사용되는듯
- (아마도 텐서코어는 특정 연산을 위한 하드웨어기 때문에 그런 것 같고, 연산할 수 있는 단위가 늘어난다면 다른 부분의 가속화도 가능하지 않을까?)

TensorCrypto: High Throughput Acceleration of Lattice-based Cryptography Using Tensor Core on GPU

WAI-KONG LEE[1], (Member, IEEE), HWAJEONG SEO[2], (Member, IEEE), ZHENFEI ZHANG [3], and SEONG OUN HWANG.[1], (Senior Member, IEEE)

[1]Department of Computer Engineering, Gachon University, South Korea. (e-mail: waikonglee, sohwang@gachon.ac.kr)
[2]College of IT Engineering at Hansung University, Seoul, South Korea.
[3]Ethereum Foundation.

Corresponding author: Seong Oun Hwang (e-mail: sohwang@gachon.ac.kr).

환경: RTX3060
WSL(Windows Subsystem for Linux) 사용

코드 실행

직접 코드 돌려보기

NTRU : NTRU-GPU-509

```
sebbang@HJSeo:~/Tensorcrypto/NTRU/NTRU-GPU-509$ ./run_test -m 0
 u16 mode N = 509, blocks: 992 threads: 32
encrypt: gpu u16 mode took 1.140032ms average 2.239749 us
decrypt: gpu u16 mode took 1.929376ms average 3.790523 us
```

```
sebbang@HJSeo:~/Tensorcrypto/NTRU/NTRU-GPU-509$ ./run_test -m 1
 tensor core mode N = 509, blocks: 992 threads: 32
encrypt: gpu tensor core mode took 1.086208ms average 2.134004 us
decrypt: gpu tensor core mode took 1.724832ms average 3.388668 us
```

일반 GPU (-m 0) :
암호화 1.1(ms) / 2.3(us)
복호화 1.9 / 3.7

텐서코어 (-m 1) :
암호화 1.0 / 2.1
복호화 1.7 / 3.3

직접 코드 돌려보기

NTRU : NTRU-GPU-677

```
sebbang@HJSeo:~/Tensorcrypto/NTRU/NTRU-GPU-677$ ./run_test -m 0
 u16 mode N = 677, blocks: 1848 threads: 32
encrypt: gpu u16 mode took 2.118432ms average 3.129146 us
decrypt: gpu u16 mode took 4.103520ms average 6.061329 us
```

```
sebbang@HJSeo:~/Tensorcrypto/NTRU/NTRU-GPU-677$ ./run_test -m 1
 tensor core mode N = 677, blocks: 1848 threads: 32
encrypt: gpu tensor core mode took 1.125824ms average 1.662960 us
decrypt: gpu tensor core mode took 2.440640ms average 3.605081 us
```

일반 GPU (-m 0) :
암호화 2.1(ms) / 3.1(us)
복호화 4.1 / 6.0

텐서코어 (-m 1) :
암호화 1.1 / 1.6
복호화 2.4 / 3.6

직접 코드 돌려보기

TensorFro(Frodo)

```
sebbang@HJSeo:~/Tensorcrypto/TensorFro$ ./TensorFro -m 0

M = 576, PADDING = 16, K= 552, blocks: 70 threads: 576


Running with gpu u16 integer...
gpu u16 took 1.729184 ms average: 3.1326us
gpu u16 result
```

```
sebbang@HJSeo:~/Tensorcrypto/TensorFro$ ./TensorFro -m 1

M = 576, PADDING = 16, K= 552, blocks: 70 threads: 576


Running with gpu u16 integer...
gpu u16 took 0.688448 ms average: 1.2472us
gpu u16 result
```

일반 GPU (-m 0) :
1.7(ms) / 3.1(us)

텐서코어 (-m 1) :
0.6(ms) / 1.2(us)

직접 코드 돌려보기

TensorLAC (LAC)

(코드 실행시 일반과 텐서코어 결과가
한 번에 나오는데 너무 길어서 잘랐음)

```
sebbang@HJSeo:~/Tensorcrypto/TensorLAC$ ./TensorLAC

M = 512, LAC_N= 512, blocks: 64 threads: 512


Running with gpu u32 integer unit...
gpu u32 took 0.504152 ms
gpu u16 result
```

```
Running with wmma...
gpu tensor core took 0.078333ms
gpu tensor core result
batch: 0
```

일반 GPU :
0.5 (ms)

텐서코어 :
0.07(ms)

직접 코드 돌려보기

TensorTRU(NTRU)

```
sebbang@HJSeo: /Tensorcrypto/TensorTRU$ ./TensorTRU

M = 512, PADDING = 3, K= 512, blocks: 64 threads: 51


Running with gpu u16 integer...
gpu u16 took 0.479303 ms average: 0.9361us

Running with wmma...
gpu tensor core took 0.128691ms average: 0.2513us
Speed up: 3.72
```

일반 GPU :
0.47 (ms)

텐서코어 :
0.12 (ms)

현재 진행 중인 부분

- NVIDIA GPU의 텐서 코어를 활용한 암호 구현 예시 테스트 및 분석
- 관련 동향 파악 -> 어떤 논문들이 있는지는 파악했지만 논문 분석을 하기에는 무리가 있었음
- 코드 돌려보고 분석 -> 환경 설정하는 데 시간이 오래 걸려서 아직 코드 분석을 하지는 못함

감사합니다