# Speedup and Password Recovery for Encrypted WinRAR3 without Encrypting Filename on GPUs (2020)

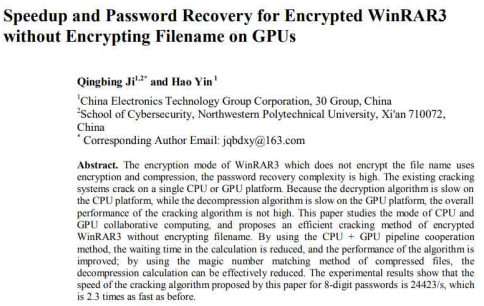
융합보안학과 윤세영

유투브 주소: https://youtu.be/b0IRl3Wr7Is

# 목차

# Speedup and Password Recovery for Encrypted WinRAR3 without Encrypting Filename on GPUs (2020)

- 본 논문에서는 'CPU'와 'GPU'의 장점을 활용하여 압축 파일(WinRAR)의 비밀번호를 복구하는 것을 효과적으로 수행하기 위해 파이프라인 기반의 암호 복구 방법을 제안함.



**Speedup and Password Recovery for Encrypted WinRAR3 without Encrypting Filename on GPUs**

**Qingbing Ji[1,2,*] and Hao Yin[1]**

[1]China Electronics Technology Group Corporation, 30 Group, China
[2]School of Cybersecurity, Northwestern Polytechnical University, Xi'an 710072, China
[*] Corresponding Author Email: jqbdxy@163.com

**Abstract.** The encryption mode of WinRAR3 which does not encrypt the file name uses encryption and compression, the password recovery complexity is high. The existing cracking systems crack on a single CPU or GPU platform. Because the decryption algorithm is slow on the CPU platform, while the decompression algorithm is slow on the GPU platform, the overall performance of the cracking algorithm is not high. This paper studies the mode of CPU and GPU collaborative computing, and proposes an efficient cracking method of encrypted WinRAR3 without encrypting filename. By using the CPU + GPU pipeline cooperation method, the waiting time in the calculation is reduced, and the performance of the algorithm is improved; by using the magic number matching method of compressed files, the decompression calculation can be effectively reduced. The experimental results show that the speed of the cracking algorithm proposed by this paper for 8-digit passwords is 24423/s, which is 2.3 times as fast as before.

# 서론

- 본 논문에서는 압축 파일 중 'WinRAR'을 대상으로 연구를 진행함.

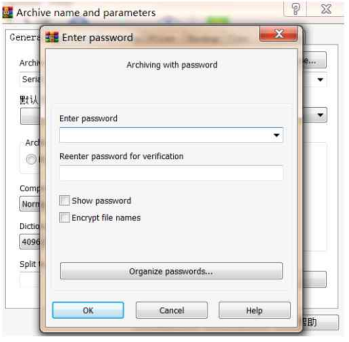
- WinRAR은 AES 암호화 알고리즘 및 SHA-1 해시 함수를 포함한 다양한 암호화 방법을 지원함.
- 암호화된 RAR 파일의 비밀번호를 복구하는 방법으로는 1. 무차별 대입 (brute-force attack) 공격과 2. 사전 파일을 사용한 무차별 대입 (brute-force with a dictionary file) 공격이 있음. (그러나 압축 파일이 점차적으로 증가함에 따라 크래킹 시간은 기하급수적으로 증가함.)

- JTR은 널리 사용되는 암호 복구 소프트웨어로 CPU 플랫폼만 지원함.
- hashcat은 파일 이름을 암호화하지 않은 WinRAR의 암호 복구를 지원하지 않음.

- AES 암호화 알고리즘 및 SHA-1 알고리즘은 GPU에서 초기 계산을 위해 병렬화 됨.
- 이후, CPU가 압축 알고리즘을 이용하여 압축 해제함.

# 본론 - Analysis of Compression Algorithm

- WinRAR의 기본 암호화 모드는 파일 이름을 암호화하지 않는데, 이것은 파일 이름, 크기, 속성, CRC 값 등 인식이 가능한 압축 파일의 영역은 암호화되지 않는다는 것을 의미함.



# 본론 - Analysis of Compression Algorithm

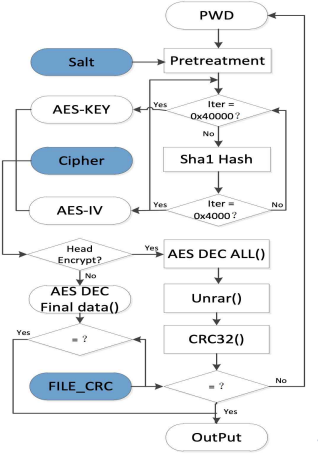
[기본 암호화 모드 진행 방식]

- 원본 파일 압축
- SHA-1 함수를 262144번 실행하여 암호화 키를 생성
- AES-128 알고리즘의 CBC 모드를 통해 압축된 파일을 암호화
- 압축 해제 시에는 위 과정이 반대로 진행되어 먼저 복호화하고, 그 다음에 압축을 해제하며, 압축 해제된 파일의 CRC를 파일 헤더에 저장된 CRC와 비교하여 압축 해제가 올바른지 판단함.

* CRC: Cyclic Redundancy Check, 순환 중복 검사, 전송된 데이터에 오류가 있는지를 확인하기 위한 체크값을 결정하는 방식

# 본론 - Analysis of Compression Algorithm

- The detailed process of WinRAR3 decryption algorithm is as follows:
- Step1. Extract the characteristic value of the file. EErar = {TYPE, SALT, CRC, PSIZE, USIZE, CIPHER}.
- Step2. Get the key and IV of AES. {KEY, IV} = SHA1{0x40000, Unicode (PWD), SALT}.
- Step3. AES decryption. PLAINcp = AES {TYPE, KEY, IV, CIPHER}.
- Step4. Decompression and Verification. CRC = CRC32{UNRAR {PLAINcp}, CRC}.



# 본론 - Optimization of Compression Algorithm

- 압축 알고리즘의 특성 및 압축 파일의 접미사(suffix)를 이용
- 암호화되지 않은 파일 이름 알고리즘의 헤더가 암호화되지 않는 특징을 미리 검증하는 방식
- 1. 압축 검증(GPU) -> 2. 접미사 검증 (CPU) -> 3. CRC 검증 (CPU)

- 1. 압축 검증
- AES 복호화 후
- 압축 모드 및 기타 정보를 포함한 처음 16바이트만을 복호화
- WinRAR 압축은 PPM과 LZ로 나뉘며, PPM 압축은 다음 조건을 모두 충족해야 함.
- 평문의 첫 번째 바이트가 규칙을 따르며, 재설정 비트 값이 1이고 MaxMB의 크기는 128보다 작아야 함.

- a. PLAINcp[0] & 0x80 = 1
- b. PLAINcp[0] & 0x20 = 1
- c. PLAINcp[1] & 0x80 != 1

## 본론 - Optimization of Compression Algorithm

- LZ 압축은 두 가지 조건을 충족해야 함.
- 바이트를 0으로 설정하여 압축 테이블을 사용하지 않았음을 확인
- 동적 허프만 테이블을 충족

- a. PLAINcp[0] & 0x40 = 0
- b. CHECK_HUFFMAN(PLAINcp)

# 본론 - Optimization of Compression Algorithm

- 압축 파일의 접미사를 기반으로, 압축 해제 후 평문의 고정된 시작 필드를 결정
- (일반적인 파일의 접미사는 표 확인)

- 암호화된 데이터를 압축 해제하고 CRC 값을 계산하여 비밀번호가 올바른지 여부를 결정

| the suffix type | the fixed beginning field |
|---|---|
| PDF | 0x25504446 |
| XML | 0x3c3f786d6c |
| DOC/XLS/PPT | 0xd0cf11e0a1b11ae1 |
| ZIP | 0x504b0304 |
| RAR | 0x526172211a0700 |
| JPG | 0xffd8ffe000104a46 |
| GIF | 0x4749463839612602 |
| MP4 | 0x0000002066747970 |

# 본론 - Optimization of Compression Algorithm

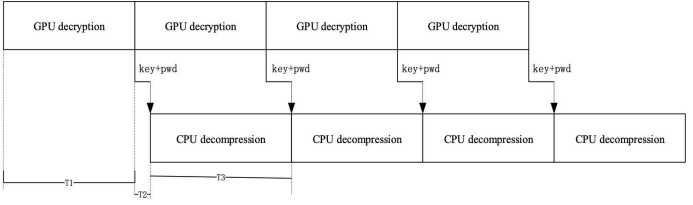
[CPU와 GPU 간의 파이프라인 협력 컴퓨팅 최적화]

- 기존의 크래킹 소프트웨어에서는 WinRAR3 암호를 복구할 때 전체 복호화 알고리즘을 단일 컴퓨팅 플랫폼에서 실행했음.
- 압축 해제와 복호화는 서로 다른 계산 특성을 가지고 있으므로 크래킹의 전반적인 성능에 저하를 불러옴.

- 다음 그림과 같이 GPU는 후보 비밀번호, AES 키 및 벡터를 포함한 계산 결과를 CPU로 출력하여 압축 해제 작업을 수행하고, CPU가 계산을 완료하면 GPU가 다음 라운드의 계산을 수행 함.
- 이때 CPU는 GPU가 데이터를 계산하고 전송할 때 대기 중이고, GPU는 CPU가 데이터를 계산할 때 대기중이므로 효율이 낮음.
- 따라서 CPU와 GPU 간의 파이프라인 모드를 사용하여 계산하고, 효율을 위해 대기 없이 계산 리소스를 교환하는 것을 제안

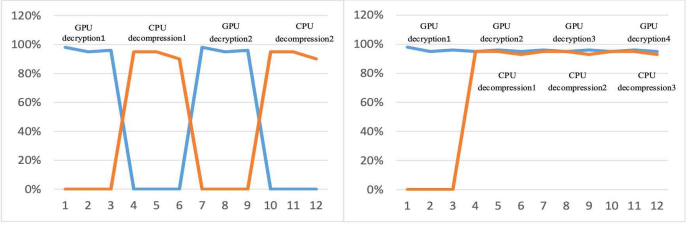
# 본론 - Optimization of Compression Algorithm

- GPU에서 복호화하는 시간을 T1, GPU에서 CPU로 데이터가 전송되는 시간을 T2, CPU에서 압축 해제하는 시간을 T3로 나타냄.
- 최적화 이전에는 T1 + T2 < T3, 최적화 후에는 T3가 크게 줄어들어 T1 + T2가 T3와 거의 동일하게 됨.
- 이렇게 함으로써 CPU가 GPU에서 출력한 비밀번호들의 집합을 복호화하고 검증할 때, GPU는 다음 집합의 계산을 시작하게 될 것임.



# 본론 - Optimization of Compression Algorithm

- GPU 컴퓨팅 공간의 크기를 조정하여 조건을 만족.
- 최적화 전후의 GPU 및 CPU의 자원 활용 비율은 다음 그림과 같음.
- 동일한 시간에, 파이프라인 협력 컴퓨팅 모드의 GPU는 두 개 이상의 데이터 세트를 복호화할 수 있고, CPU는 하나의 데이터 세트를 압축 해제할 수 있음.




a) before optimization

b) after optimization

## 본론 - Experimental results and analysis

- 이 논문에서 진행한 최적화 테스트의 플랫폼 구성 및 실험 결과는 아래의 표와 같음

| | |
|---|---|
| CPU | Xeon(R)E5-2620 |
| GPU | NVIDIA 1080Ti |
| CUDA | 10.2 |
| operating system | Linux CentOS7 |

| size of compressed file | speed before optimization | speed after optimization |
|---|---|---|
| 1K | 10981 | 24423 |
| 10M | 9738 | 22423 |
| 100M | 6235 | 16423 |

# 결론

- 컴퓨터 성능의 빠른 향상으로 GPU 병렬 컴퓨팅을 이용하여 암호를 복구하는 데 집중함
- 복호화와 압축 해제가 혼합되어 있기 때문에 파일 이름을 암호화하지 않는 WinRAR의 암호화 모드는 크래킹이 어려움
- 이 문제를 해결하기 위해 본 논문에서는 이종 다중 코어 아키텍처와 WinRAR3 압축 알고리즘의 일부 특성을 연구하고 CPU+GPU 파이프라인 협력 컴퓨팅 방법을 제안
- 실험 결과로, 본 논문에서 제안한 방식이 크래킹 알고리즘의 성능을 획기적으로 개선할 수 있음을 보여 주었음.


# Q&A