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: jqbdux@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.

1. Introduction

In the era of rapid development of information technology such as Internet technology, big data technology and Internet of things technology, effective compression is necessary to solve the problem of increasing data volume. Compressed file takes up very little space, has the function of encryption and error correction, and saves the transmission bandwidth, which makes it widely used in individuals, institutions and some commercial organizations. The most common compression software is WinRAR, which supports a variety of encryption methods, including AES encryption algorithm and SHA-1 hash function. The mixed way of WinRAR decryption and decompression brings great challenges to the password recovery under the current computer processing capacity.

At present, brute force and brute force with a dictionary file are two ways to recover the password of an encrypted RAR file. However, with the gradual increase of compressed files, the cracking time increases exponentially. Most of the hardware used in password recover is currently a single computing platform on CPU or GPU. Due to the limitations of the single platform, it can’t solve the decryption and decompression calculation well, resulting in the waste of computing resources and the performance of the algorithm is not ideal. For example, JTR is a widely used password recovery software, which only supports CPU platform. Although hashcat is the world's fastest password recovery software on GPU, it does not support password recovery for encrypted WinRAR without encrypting filename. Meanwhile, there is no relevant password recovery algorithm on FPGA platform.

With the continuous improvement of GPU, heterogeneous parallel computing based on CPU + GPU becomes simple. Then, we propose a password recovery method based on pipeline, using the advantages of CPU + GPU heterogeneous platform to complete effectively the password recovery of
compressed files. By studying the encryption characteristics of compressed files and the structural characteristics of CPU and GPU, the password recovery algorithm is divided into two steps by using the advantages of pipeline parallel work. The AES encryption algorithm and SHA-1 algorithm are parallelized by GPU for preliminary calculation, and then the decompression algorithm is put on the CPU for decompression to complete the final verification. The overall architecture makes use of the operating efficiency of each processor, and improves the efficiency of password cracking.

2. Analysis of Compression Algorithm

WinRAR software has two versions, RAR3 and RAR5. The interface of software encryption is shown in Figure 1. The default encryption mode of WinRAR does not encrypt the file name, that is, it encrypts only compressed files, while other perceptible compressed file areas such as file name, size, attributes, CRC values, comments and other blocks will not be encrypted. In this encryption mode, WinRAR first compresses the original file, then generates the encryption key through 262144 times SHA-1 function, and encrypts the compressed file through CBC mode of AES-128 algorithm. When decompressing, the process is reversed, that is, decrypt first, then decompress, and then judge whether the decompression is correct by comparing the CRC of the decompressed file with the CRC saved in the file header.

![Figure 1. The interface of WinRAR3 software encryption](image)

The detailed process of WinRAR3 decryption algorithm is as follows:

Step1. Extract the characteristic value of the file. $E_{E_{RAR}} = \{\text{TYPE, SALT, CRC, PSIZE, USIZE, CIPHER}\}$.

Step2. Get the key and IV of AES. $\{\text{KEY, IV}\} = \text{SHA1}\{0x40000, \text{Unicode (PWD)}, \text{SALT}\}$.

Step3. AES decryption. $\text{PLAINcp} = \text{AES}\{\text{TYPE, KEY, IV, CIPHER}\}$.

Step4. Decompression and Verification. $\text{CRC} = \text{CRC32}\{\text{UNRAR}\{\text{PLAINcp}\}, \text{CRC}\}$.

In the first step, the characteristic values of encrypted files are extracted, including encryption type, random number salt, and final CRC check value. If it is an unencrypted file name, information such as the size of the compressed package, the size before compression, and the compressed ciphertext can also be obtained. In the second step, after combining the user password in Unicode format with 8-byte random number salt, the first 16 bytes of the final result will be generated as the key of AES after a total of 40000 times of SHA-1 hash calculation, while the IV of AES is to save the last byte of SHA-1 calculation result when the number of cycles is a multiple of 0x4000, and generate IV after accumulating 16 calculations. In the third step, the encryption type is used to determine whether the
decrypted content includes the file header, and the key and IV of AES are used to decrypt the data. In the fourth step, the compressed plaintext is decrypted, and the CRC32 value of the whole plaintext is calculated and compared with the extracted CRC value. If it is consistent, the password is correct. The password recovery process is shown in Figure 2.

![Password Recovery Process Diagram](Figure 2)

**Figure 2.** The password recovery process of encrypted WinRAR3 file

3. Optimization of Compression Algorithm

3.1. Optimization of WinRAR3 Compressed Large File Verification

The encryption algorithm of WinRAR3 which does not encrypt the file name uses the conventional method to make all passwords participate in AES decryption and decompression. Then, the CRC verification is carried out. The efficiency is very low. The larger the compressed file is, the slower the decompression speed is, and it is very time-consuming.

The characteristics of the compression algorithm and the suffix of the compressed file are used to verify in advance by the feature that the header of unencrypted file name algorithm isn’t encrypted. We divide the verification into three rounds. The first round of verification is called compression verification, which is executed on the GPU. The second round is called suffix verification, which is carried out on the CPU, and the third round is called CRC verification, which is carried out on the CPU.

In the first round of verification, it needs to be verified after AES decryption, and only needs to decrypt the first 16 bytes, including the attributes of compression mode and other information. WinRAR compression is divided into PPM and LZ. PPM compression needs to meet three conditions: the first byte of plaintext conforms to the rule, the reset bit value is 1 and the size of MaxMB needs to be less than 128. The specific verification process is as follows:

- a. \( \text{PLAINcp}[0] \& 0x80 = 1 \)
- b. \( \text{PLAINcp}[0] \& 0x20 = 1 \)
- c. \( \text{PLAINcp}[1] \& 0x80 \neq 1 \)

As long as the above three conditions are met, PPM compression characteristics can be met, and the next step of verification can be carried out. LZ compression needs to meet two conditions. The
first is to set a byte to zero, which means no compression table is used, and the second is to satisfy the dynamic Huffman table. The specific process of calibration is as follows:
  a. PLAINcp[0] & 0x40 = 0
  b. CHECK_HUFFMAN(PLAINcp)

The next verification can be carried out only if the compressed data meets the compression verification. Otherwise, the password is wrong, and there is no need to carry out the next step of calculation, avoiding the subsequent calculation verification.

The second round of verification is based on the suffix type of the compressed file to determine the fixed beginning field of the plaintext after decompression, for example, the beginning plaintext of the doc file is 0xd0cf11e0a1b11ae1. Only the plaintext that meets the conditions can be checked in the next step. In this way, most of the passwords can be eliminated and the efficiency of the algorithm can be improved while ensuring the correctness of the password. At present, 34 types of fields corresponding to different file suffixes have been summarized. Some common suffix fields are shown in Table 1.

| the suffix type | the fixed beginning field |
|-----------------|---------------------------|
| PDF             | 0x25504446                |
| XML             | 0x3c3f786d6c              |
| DOC/XLS/PPT     | 0xd0cf11e0a1b11ae1        |
| ZIP             | 0x504b0304                |
| RAR             | 0x526172211a0700          |
| JPG             | 0xffd8ffec000104a46      |
| GIF             | 0x4749463839612602        |
| MP4             | 0x0000002066747970        |

In the case of meeting the first two rounds of verification, the main purpose of the final CRC verification is to decompress all encrypted data, calculate and compare the CRC value to determine whether the password is correct.

### 3.2. Optimization of Pipeline Collaborative Computing between CPU and GPU

In the previous cracking software, the whole decryption algorithm was executed on a single computing platform when recovering WinRAR3’ password. Decompression and decryption have different computing characteristics, that is, one is calculation intensive and the other is data intensive, which leads to a big bottleneck in the overall performance of cracking.

According to the different characteristics of the two computing platforms, in order to give full play to the computing advantages of different algorithms on different platforms, we designed the pipeline operation mode of CPU and GPU, that is, put the steps 2 and 3 of Winrar3 decryption algorithm described in Section 2 in GPU, and step 4 in CPU. GPU outputs the calculation results including candidate passwords, AES keys and vectors, to the CPU for decompression operation, and the GPU does the next round of calculation when the CPU completes the calculation. The flow chart is shown in Figure 3.

![Figure 3. The mode of CPU and GPU coordinated computing](image-url)
According to Figure 3, CPU will wait when GPU calculates and transmits data, and GPU will wait when CPU calculates data, resulting in low efficiency. Therefore, we consider to use CPU and GPU pipelining mode to calculate, and exchange area for efficiency to avoid waiting for computing resources.

The time of GPU decryption is recorded as T1, the time of GPU data transmission to CPU is T2, and the time of CPU decompression is T3. Before optimization of Winrar3 compressed large file verification, it is obvious that T1 + T2 < T3. After optimization of Winrar3 compressed large file verification, T3 will be greatly shortened. We adjust the relevant parameters of the first round of verification and the second round of verification to make T1 + T2 approximately equal to T3. In this way, when the CPU decompreses and verifies the output of a set of passwords on the GPU, the GPU begins to decrypt and calculate the next set of passwords, as shown in Figure 4.

We meet this condition by adjusting the size of GPU computing space. The resource utilization ratio of GPU and CPU before and after optimization is shown in Figure 5a) and Fig. 5b). In the same time, the GPU in pipeline collaborative computing mode can decrypt more than two sets of data, and the CPU can decompress one set of data.

![Figure 4. GPU and CPU pipeline collaborative computing](image)

Figure 4. GPU and CPU pipeline collaborative computing

![Figure 5. The resource utilization ratio of GPU and CPU before and after optimization](image)

Figure 5. The resource utilization ratio of GPU and CPU before and after optimization

4. Experimental results and analysis

In this paper, the above optimization is tested. The test platform configuration and experimental results are shown in Table 2 and Table 3.

| Table 2. Experimental environment |
|-------------------------------|
| CPU                           | Xeon(R)E5-2620                  |
| GPU                           | NVIDIA 1080Ti                   |
| CUDA                          | 10.2                            |
| operating system              | Linux CentOS7                   |
Table 3. Speed comparison before and after WinRAR3 optimization

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

From the experimental results, we can see that the performance of this paper is more than twice that before optimization, and the larger the compressed file is, the slower the decompression speed is, which leads to the reduction of cracking speed.

5. Conclusion

With the rapid improvement of computer performance, the use of GPU parallel computing for password recovering has become a research hotspot. Because of the mixture of decryption and decompression, the encryption mode of WinRAR which does not encrypt the file name has become a difficult problem to crack with high performance. To solve this problem, this paper studies the heterogeneous multi-core architecture and some characteristics of WinRAR3 compression algorithm, and proposes a method of CPU+GPU pipeline collaborative computing. The experimental results show that this method can significantly improve the performance of the cracking algorithm. The next step includes the performance optimization of GPU and CPU, and the further optimization of CPU and GPU cooperation.

6. Acknowledgement

This work is partially supported by Sichuan Province Application Fundamental Research Project (No. 2018JY0102)

7. References

[1] Alexander Roshal. WinRAR archiver [DB/OL]. http://www.rarlab.com. 2015.
[2] Radicke S, Hahn J U, Qi W, et al. A Parallel HEVC Intra Prediction Algorithm for Heterogeneous CPU+GPU Platforms[J]. IEEE Transactions on Broadcasting, 2016, 62(1):103-119.
[3] Hu Guang, Ma Jianhua, Huang Benxiong. Password Recovery for RAR Files Using CUDA [C]// Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing. 2009:486-490.
[4] On the security of the WinRAR encryption feature[J]. Gary S.-W. Yeo,Raphael C.-W. Phan. International Journal of Information Security. 2006(2).
[5] Optimization of RAR password brute-force cracking based on OpenCL. ZHOU B, ZHANG Y Q, AN X J,et al. High-Performance Computing China 2014 . 2014.
[6] Enhanced Key Derivation Function of HMAC-SHA-256 Algorithm in LTE Network. CHEN F T, YUAN J L. Fourth International Conference on Multimedia Information NETWORKING and Security. 2012.
[7] An Xiaojing, Jia Haipeng, Zhang Yunquan. Optimized password recovery for encrypted RAR on GPU[C]// Proc of the 17th International Conference on High Performance Computing and Communications, IEEE the 7th International Symposium on Cyberspace Safety and Security, and IEEE the 12th International Conference on Embedded Software and Systems. Piscataway,NJ: IEEE Press,2015: 591-598.
[8] WinRAR temporary folder artefacts[J]. Geoffrey Fellows. Digital Investigation. 2009(1).
[9] NVIDIA company, CUDA C++ Programming Guide, 2020.
[10] https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html.
[11] Zhou Ben, Zhang Yunquan, An Xiaojing, et al. Optimization of RAR password brute-force cracking based on OpenCL[C]// Proc of High-Performance Computing China. Beijing: China Computer Federation,2014: 871-874.