

# Known Plaintext Attack

A Known-Plaintext Attack is a type of encryption breaking technique when it comes to .zip archives. This can be useful for an attacker who is trying to gain access to files on machine or from a forensics investigator POV. The attack can only be used in a specific scenario, but it is quite good at what it does.

## Resources needed:

- Windows machine (host or VM)
- Kali linux

## Software Downloaded/Installed:

- 7zip
- cmake
- pkcrack

## PART I: Gather the Necessary Files

On the target machine we need several files and then put them in an archive with password protection to simulate an encrypted archive which we will crack on our attack machine.

We will need 3+ images and a text file.

- 1) Gather 3 images
  - a. Either use your browser and download 3 random images from the internet or it could be images on your file system. It does not matter.
- 2) Creating the text file
  - a. The first text file we will create will be called "**Known.txt**" In the file, we will add the following text:

"In order for this attack to work we must know (have access to) one of the files in the archive. This is the key to the known plaintext attack. The program "pkcrack" is a command line tool that's relatively simple and straight forward but can be finicky. Follow all the directions I have written, and this should go smoothly. Also, I'm purposely making this text drag on because the more bytes this file is, the faster it will work. If the file is not a certain amount of bytes it will not work at all.

Now here is the excerpt from the song "I'm Blue" by Eiffel 65

'Yo listen up here's a story  
About a little guy that lives in a blue world  
And all day and all night and everything he sees is just blue  
Like him inside and outside  
Blue his house with a blue little window  
And a blue Corvette

And everything is blue for him  
And himself and everybody around  
'Cause he ain't got nobody to listen  
I'm Blue da ba dee da ba daa  
Da ba dee da ba daa, da ba dee da ba daa  
daa, da ba dee da ba daa  
Da ba dee da ba daa, da ba dee da ba daa  
daa, da ba dee da ba daa  
I'm Blue da ba dee da ba daa  
Da ba dee da ba daa, da ba dee da ba daa  
daa, da ba dee da ba daa

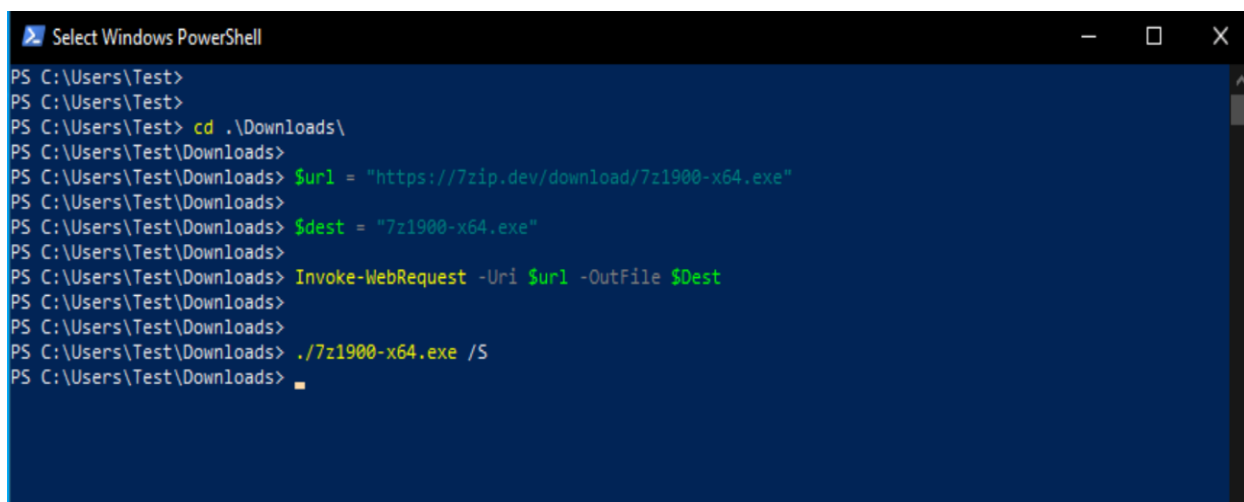
Da ba dee da ba daa, da ba dee da ba daa, da ba dee da ba daa  
I have a blue house with a blue window  
Blue is the color of all that I wear  
Blue are the streets and all the trees are too  
I have a girlfriend and she is so blue'

I will explain more in the actual written guide.  
FILLER FILLER FILLER FILLER"

## PART II: Installing 7zip via PowerShell

We will need a compression tool for this lab. I've tested other compression tools and the only one that seemed to work for me was 7zip so I will be demonstrating that in this guide. If you already have 7zip installed on your Windows machine then go ahead and skip to PART III.

- 1) Open PowerShell and enter the following commands
  - a. `cd .\Downloads\`
  - b. `$url = "https://7zip.dev/download/7z1900-x64.exe"`
  - c. `$dest = "7z1900-x64.exe"`
  - d. `Invoke-WebRequest -Uri $url -OutFile $Dest`
    - i. This may take up to 30 seconds to download from the website
  - e. `./7z1900-x64.exe /S`
    - i. This will require you to allow the installation of 7zip

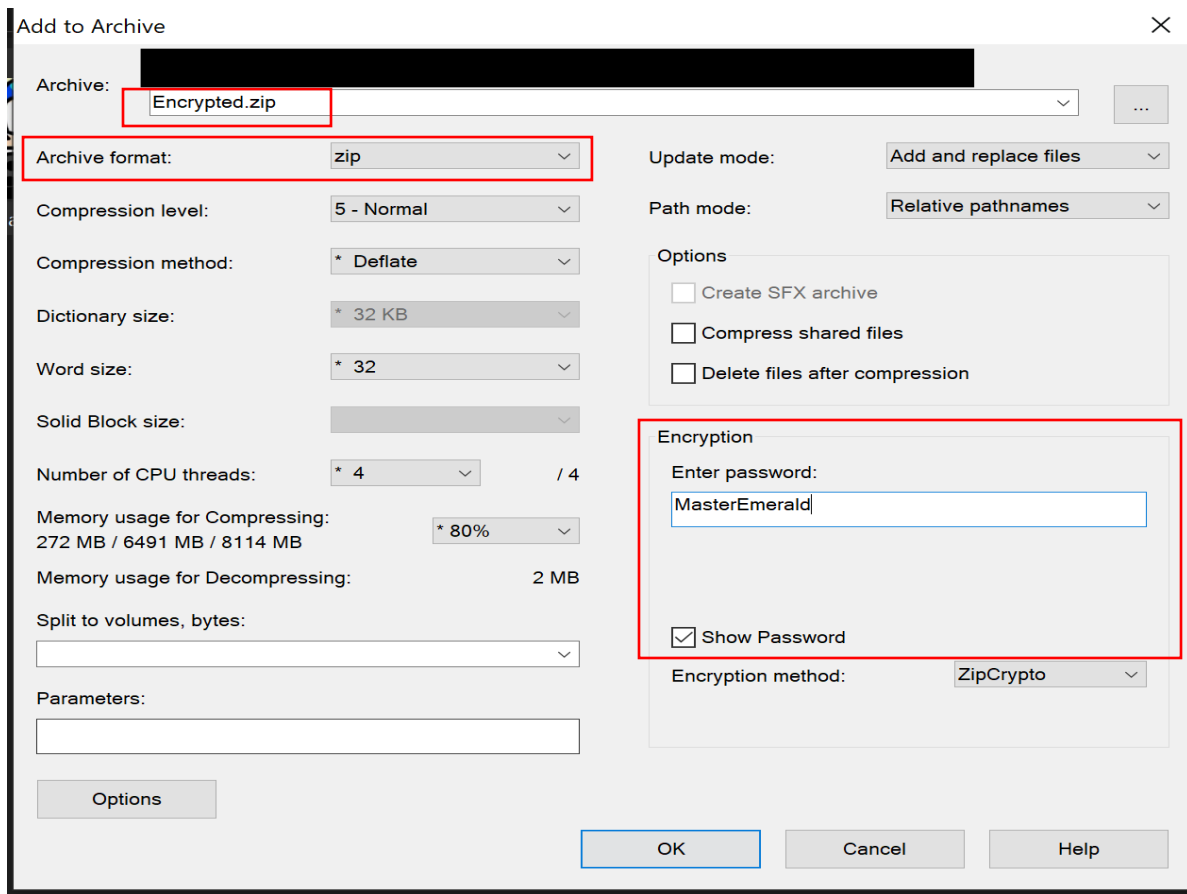


```
Select Windows PowerShell
PS C:\Users\Test>
PS C:\Users\Test>
PS C:\Users\Test> cd .\Downloads\
PS C:\Users\Test\Downloads>
PS C:\Users\Test\Downloads> $url = "https://7zip.dev/download/7z1900-x64.exe"
PS C:\Users\Test\Downloads> $dest = "7z1900-x64.exe"
PS C:\Users\Test\Downloads> Invoke-WebRequest -Uri $url -OutFile $Dest
PS C:\Users\Test\Downloads>
PS C:\Users\Test\Downloads> ./7z1900-x64.exe /S
PS C:\Users\Test\Downloads>
```

## PART III: Archiving Files and Adding Password Protection

Okay great, we got that out of the way. Now we can encrypt these files with password protection.

- 1) Creating an encrypted archive
  - a. To do so you must select all the files (3 images and 1 text file) and right click.
    - i. Choose `7zip > Add to archive....`
  - b. Title the archive "`Encrypted.zip`"
  - c. Archive format: zip
    - i. This is critical
  - d. For encryption, enable password protection. I will use the password of "`MasterEmerald`".

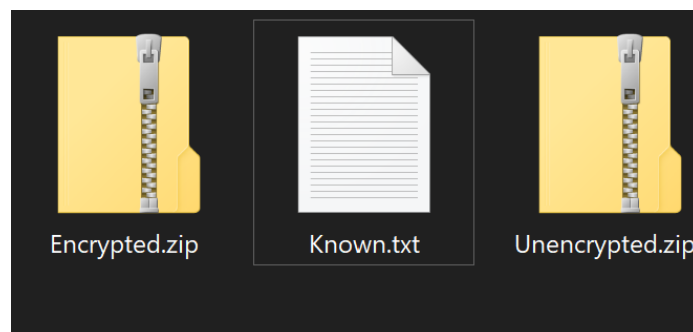


Now try opening the Encrypted.zip using 7zip (this may have to be done by right clicking). Inside the archive we can see file names but if we try viewing them, it asks for a password.

## 2) Creating an unencrypted archive

- a. Select the file "Known.txt" and create an archive
  - i. Follow the same steps as above except **DO NOT** add a password

At this point we should have 2 archives (**Encrypted.zip** & **Unencrypted.zip**) and the Known.txt file.

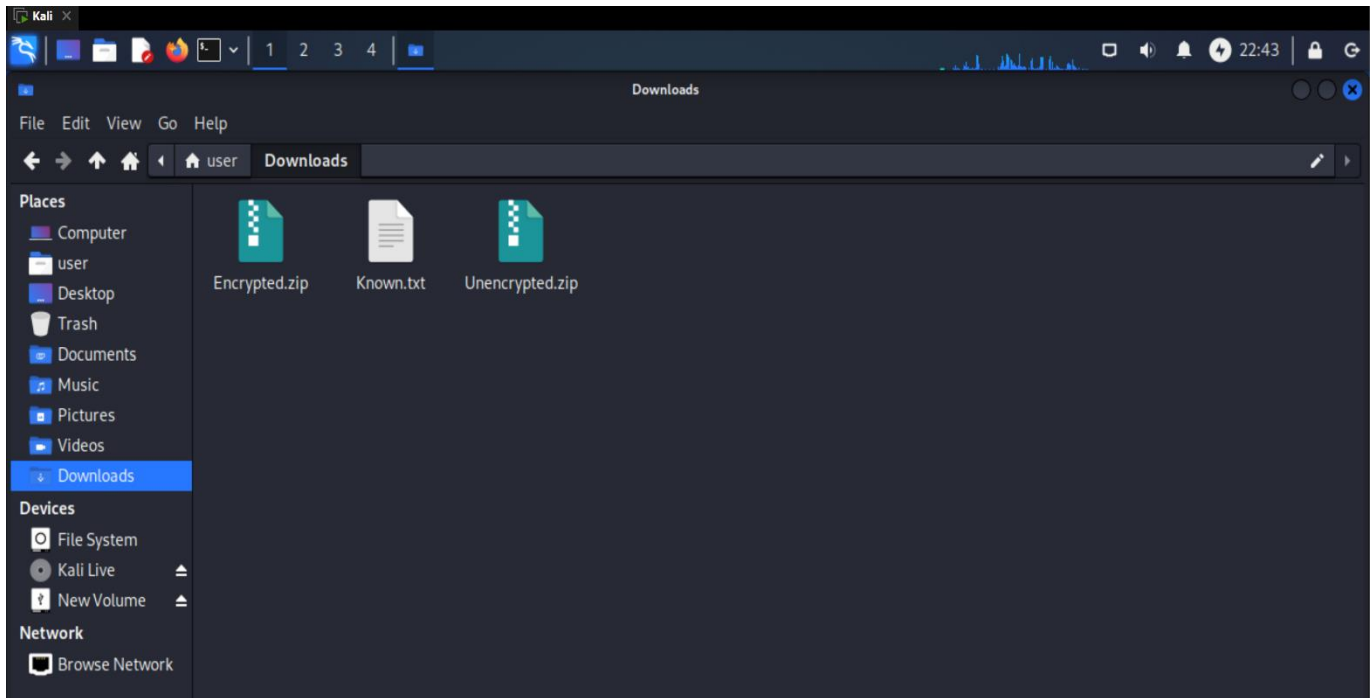


## PART IV: Exfiltrating the files to your attack machine

I'm going to use a USB device to exfiltrate the archives and file to my Kali machine but, it can be any method you would like (email, ftp, cloud storage, USB, etc..)

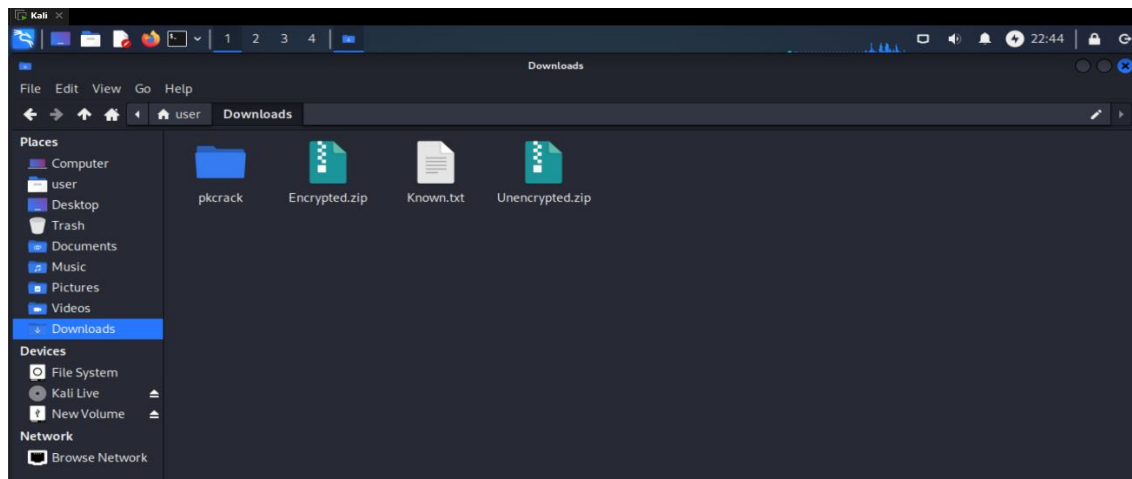
To easily follow along, transfer these files to the "Downloads" directory on your **kali machine**.

- Transfer **Encrypted.zip**, **Unencrypted.zip**, and **Known.txt**



## PART V: Downloading/Installing CMAKE and PKCRACK

- 1) Next on the Kali VM, run the following commands to download and install cmake
  - a. `sudo apt show cmake`
  - b. `sudo apt install cmake g++ make`
- 2) Then download and install pkcrack
  - a. `cd Downloads`
  - b. `git clone https://github.com/keyunluo/pkcrack`
  - c. `mkdir pkcrack/build`
  - d. `cd pkcrack/build`
  - e. `cmake ..`
  - f. `make`



## PART VI: Understanding PKCRACK

First, knowing how PKCRACK works is important. PKCRACK is a Known-Plaintext Attack (KPA) tool. Pkcrack is only able to crack an encrypted archive if we know one of the files that is inside the archive. If we have one of the files then we must create another archive that is unencrypted with the file that we know. In parts 1-3 we set up the encrypted archive and the unencrypted archive and transferred that to our Kali machine including the Known.txt file.

In order to condense this information down, PKCRACK relies on 3 factors:

- 1) the encrypted archive (Encrypted.zip)
- 2) the known-plaintext file (Known.txt)
- 3) an unencrypted archive (Unencrypted.zip)

## PART VII: USING PKCRACK

We should be all set to run pkcrack at this point. The 2 archives (Encrypted.zip & Unencrypted.zip), the known-plaintext file (Known.txt), and pkcrack should all be in the Downloads directory.

The command context for pkcrack (not the actual command we run):

```
sudo pkcrack -C encrypted-ZIP -c ciphertextname -P plaintext-ZIP -p plaintextname -d decrypted-ZIP -a
```

### 1) Running pkcrack

- Open a command prompt and navigate to your Downloads directory if you are not there already

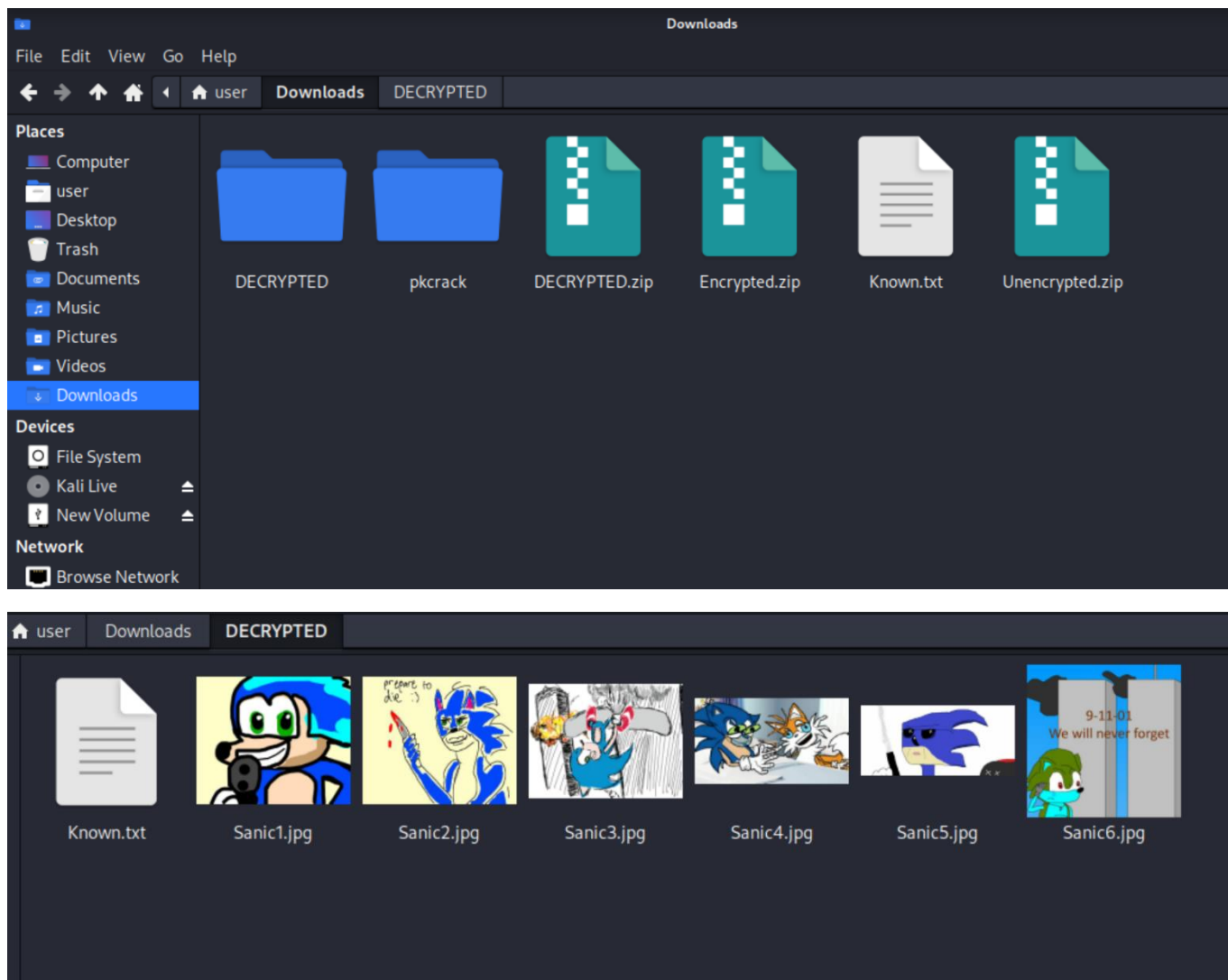
```
sudo pkcrack/bin/pkcrack -C Encrypted.zip -c Known.txt -P Unencrypted.zip -p Known.txt -d DECRYPTED.zip -a
```

- -d: used to create the archive where the contents from Encrypted.zip will go to
- -a: absolutely no idea what it does but it's required

```
user@kali: ~/Downloads
File Actions Edit View Help
$ cd Downloads

(user@kali)-[~/Downloads]
$ sudo pkcrack/bin/pkcrack -C Encrypted.zip -c Known.txt -P Unencrypted.zip -p Known
n.txt -d DECRYPTED.zip -a
[sudo] password for user:
Files read. Starting stage 1 on Thu Jan 19 10:42:09 2023
Generating 1st generation of possible key2_673 values ... done.
Found 4194304 possible key2-values.
Now we're trying to reduce these...
Done. Left with 13198 possible Values. bestOffset is 24.
Stage 1 completed. Starting stage 2 on Thu Jan 19 10:42:29 2023
Searching ... 28.5%28.2%3-Searching ... 28.2%28.1%g ... 27.9%g ... 27.1%
Ta-daaaaa! key0=59d35d67, key1=5c647d1f, key2=d3330c6b
Probabilistic test succeeded for 654 bytes.
Ta-daaaaa! key0=59d35d67, key1=5c647d1f, key2=d3330c6b
Probabilistic test succeeded for 654 bytes.
Ta-daaaaa! key0=59d35d67, key1=5c647d1f, key2=d3330c6b
Probabilistic test succeeded for 654 bytes.
Ta-daaaaa! key0=59d35d67, key1=5c647d1f, key2=d3330c6b
Probabilistic test succeeded for 654 bytes.
Ta-daaaaa! key0=59d35d67, key1=5c647d1f, key2=d3330c6b
Probabilistic test succeeded for 654 bytes.
Ta-daaaaa! key0=59d35d67, key1=5c647d1f, key2=d3330c6b
Probabilistic test succeeded for 654 bytes.
Stage 2 completed. Starting zipdecrypt on Thu Jan 19 10:52:49 2023
Decrypting Known.txt (5bf16fc0a84bfb4974249229) ... OK!
Decrypting Sanic1.jpg (a1eb225594a2364609411da1) ... OK!
Decrypting Sanic2.jpg (83900e0ce2f0939bd1148420) ... OK!
Decrypting Sanic3.jpg (677c1159b7dcde1f82c9d942) ... OK!
Decrypting Sanic4.jpg (3e548b33c6151213353af6d5) ... OK!
Decrypting Sanic5.jpg (4cd9c81be99d0186006f1720) ... OK!
Decrypting Sanic6.jpg (12bfd667b17d09db2084140e) ... OK!
Finished on Thu Jan 19 10:52:49 2023

(user@kali)-[~/Downloads]
$
```



Hooray! We did it Sonic!

### \*\*\* Additional Information:

Using pkcrack can be quite frustrating as it has not been updated in a few years and there is very little online resources regarding it. However, it does work if you use it right.

- The tool cannot crack .7z archives
- The official README.md file states that the less bytes the known-plain text file is, the longer it will take. I have found that it takes a long time if the file size is not at least 1KiB (1,000 bytes).
- ALSO, the known-plain text file does not have to be a text (.txt) file. It can be other file formats like JPEG (.jpg), PNG, etc..
  - This can be confusing because we refer to the file that we know as the “known-plain text file) but it can be other file types

Resources:

<https://github.com/keyunluo/pkcrack>

<https://www.youtube.com/watch?v=ZvYGd6qUjFU>