

BÁO CÁO BÀI TẬP

Môn học: **Pháp chứng kỹ thuật số**

Tên chủ đề: **Bài tập Queen of Stegano**

GVHD: Nghi Hoàng Khoa

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT334.O21.ANTN

STT	Họ và tên	MSSV	Email
1	Hà Thị Thu Hiền	21522056	21522056@gm.uit.edu.vn
2	Phạm Ngọc Thơ	21522641	21522641@gm.uit.edu.vn
3	Nguyễn Ngọc Nhung	21521248	21521248@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	Câu 1	100%

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành



BÁO CÁO CHI TIẾT

1. Câu 1: Queen of Stegano

Descriptions:

- My picture is corrupted
- My zip file lost password
- So sad :{(

Format flag: W1{...}

 CrackMe	6/6/2024 12:15 AM	Compressed (zipp...	2,691 KB
 Chall	6/6/2024 12:15 AM	PNG File	2,692 KB

- Ta có tài nguyên là 1 file zip và 1 ảnh dạng png, nhưng để chắc chắn ta thử xem file ảnh có thật sự được coi là png hay không.

```
(hahien@hahien)-[~]  
$ file Chall.png  
Chall.png: data  
  
(hahien@hahien)-[~]  
$
```

- Điều này có nghĩa là lệnh file không nhận dạng được định dạng cụ thể của tệp Chall.png, và chỉ trả về "data". Thông thường, một tệp PNG sẽ được nhận dạng là image/png, nhưng trong trường hợp này, có thể tệp Chall.png không thực sự là một tệp hình ảnh hợp lệ hoặc bị hỏng.
- Bây giờ, ta sẽ kiểm tra trực tiếp nội dung nhị phân của tệp, bằng cách thực hiện lệnh **xxd Chall.png | head -n 20** dùng để hiển thị 20 dòng đầu tiên của tệp Chall.png dưới dạng hexdump.

```

(hahien@hahien)-[~]
$ xxd Chall.png | head -n 20
00000000: 0000 0af0 0000 0834 0802 0000 00ea 0e34 .....4.....4
00000010: 8400 0100 0049 4441 5478 9cec fd5d 97e4 .....IDATx... ]..
00000020: aaae 2e08 3f8f 7064 adb5 778f d1ff ffcf ....?.pd..w....
00000030: 75df bfe3 9c3e 7b7d cdaa cc30 d27b 21c0 u....>{ }...0.{!.
00000040: 181b 871d 1991 1f55 d6ac e1e9 2430 0810 .....U....$0..
00000050: 4248 42f0 fff9 7fff 1f1a 4123 6920 0918 BHB.....A#i..
00000060: 4100 0680 fe3f 2361 0680 3453 cf9f 7ea7 A....?#a..4S..~.
00000070: 1940 1024 cd00 1848 7f38 58c4 1a88 c86a .@.$...H.8X....j
00000080: 7a01 92e5 0940 5557 d301 ddf8 7c4f ba18 z....@UW....|O..
00000090: 0048 95ee 29a9 f4f5 62d6 f137 0034 4df5 .H.. )...b..7.4M.
000000a0: 143c 0580 9901 80dd 68f5 ddd0 6baf cc33 .<.....h...k..3
000000b0: 946c 099f dde5 3c0a 9f26 43c9 d6cb dfc3 .l....<..8C....
000000c0: b3f4 7f53 4eec e4ef c1d1 7e38 8a67 38d8 ...SN.....~8.g8.
000000d0: 9f9f 3e2e 0df4 f0e9 e67f 727b d999 9047 ..>.....r{...G
000000e0: f1ec e223 6135 bdf0 9fbd e51c c4a7 9998 ...#a5.....
000000f0: d3e7 7667 7bdb 723a e9d2 49bf 59fe 4e7e ..vg{.r:..I.Y.N~
00000100: 52d2 7bf3 bda9 77d4 631c 9208 abe5 3408 R.{...w.c.....4.
00000110: 4ced e27a 7a2f 3f70 acff 2df7 6853 4eaf L..zz/?p..-hSN.
00000120: bdec 7473 b7df e2b8 fe41 0f68 0014 2cf5 ..ts.....A.h...
00000130: 5ab5 1e19 a7de d654 ef31 bea7 b831 7f77 Z.....T.1...1.w

```

- Thông thường, signature của tệp PNG chuẩn bao gồm 8 byte đầu tiên:

89 50 4E 47 0D 0A 1A 0A

- Nhưng theo kết quả trên thì chưa khớp với định dạng chuẩn của PNG.

PNG là một định dạng tệp hình ảnh có cấu trúc cụ thể. Mỗi tệp PNG bao gồm các chunk (khối dữ liệu) bắt buộc, như IHDR (Image Header), IDAT (Image Data), và IEND (Image End). Dưới đây là cấu trúc cơ bản:

- Signature: 8 byte
- IHDR chunk: Mô tả thông tin cơ bản về hình ảnh (chiều rộng, chiều cao, độ sâu màu, v.v.)
- IDAT chunk: Chứa dữ liệu hình ảnh nén
- IEND chunk: Đánh dấu kết thúc tệp PNG

Trong đầu ra từ xxd, chunk IDAT bắt đầu từ byte thứ 16:

00000010: 8400 0100 0049 4441 5478 9cec fd5d 97e4IDATx...]..

- Thiếu chunk IHDR và IEND.
- Bây giờ ta sẽ thêm signature và chunk cần thiết, ta thêm signature của file png là 89 50 4E 47 0D 0A 1A 0A, thấy rằng nó cũng thiếu một chunk quan trọng là IHDR, ta thêm 00 00 00 0D 49 48 44 52, và phần ở byte 0 đến chunk IDAT sẽ thuộc chunk IHDR. Và cuối cùng là thêm IEND là 49 45 4E 44 AE 42 60 82.
- Đầu tiên, tạo các file chứa signature, chunk IHDR, và chunk IEND.
 - Tạo file chứa signature và IHDR.

```
(hahien@hahien)-[~]
$ echo -n -e '\x89\x50\x4E\x47\x0D\x0A\x1A\x0A\x00\x00\x00\x0D\x49\x48\x44\x52\x00\x00\x0A\xF0\x00\x00\x08\x34\x08\x02\x00\x00\x00\xEA\x0E\x34' > png_header.bin
```

- Tạo file chứa IEND.

```
(hahien@hahien)-[~]
$ echo -n -e '\x49\x45\x4E\x44\xAE\x42\x60\x82' > png_iend.bin
```

- Tạo file tạm chứa phần IDAT hiện tại.

```
(hahien@hahien)-[~]
$ dd if=Chall.png of=png_idat.bin bs=1 skip=16

2756443+0 records in
2756443+0 records out
2756443 bytes (2.8 MB, 2.6 MiB) copied, 7.7423 s, 356 kB/s
```

- Kết hợp các file thành một tệp PNG hợp lệ.

```
(hahien@hahien)-[~]
$ cat png_header.bin png_idat.bin png_iend.bin > Fixed_Chall.png
```

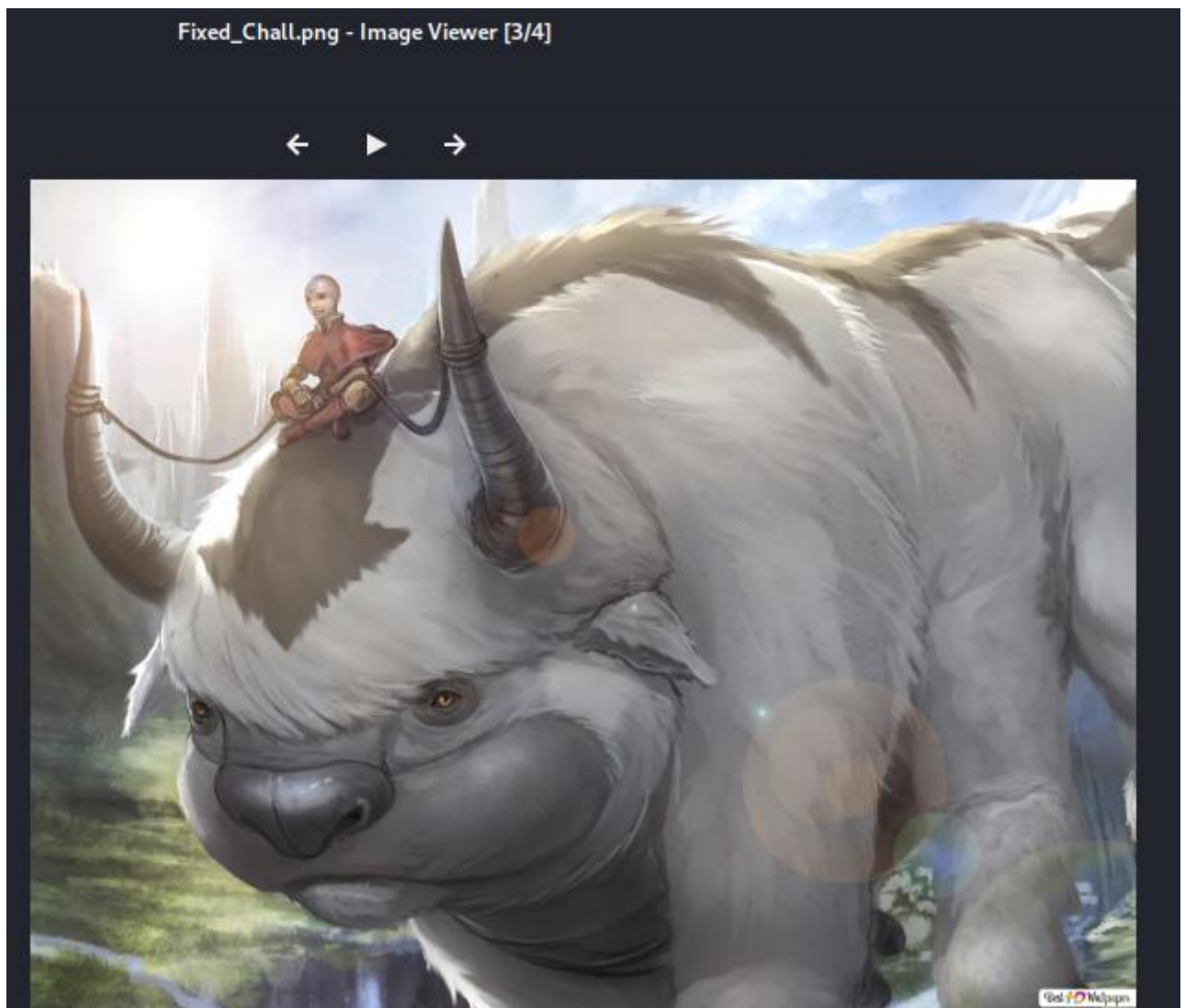
- Thử lại với xxd xem kết quả.

```
(hahien@hahien)-[~]
$ xxd Fixed_Chall.png | head -n 20
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452 .PNG.....IHDR
00000010: 0000 0af0 0000 0834 0802 0000 00ea 0e34 .....4.....4
00000020: 8400 0100 0049 4441 5478 9cec fd5d 97e4 .....IDATx ... ] ..
00000030: aae 2e08 3f8f 7064 adb5 778f d1ff ffcf ....?.pd..w.....
00000040: 75df bfe3 9c3e 7b7d cdaa cc30 d27b 21c0 u....>{ } ... 0.{!.
00000050: 181b 871d 1991 1f55 d6ac e1e9 2430 0810 .....U....$0 ..
00000060: 4248 42f0 fff9 7fff 1f1a 4123 6920 0918 BHB.....A#i ..
00000070: 4100 0680 fe3f 2361 0680 3453 cf9f 7ea7 A....?#a..4S..~.
00000080: 1940 1024 cd00 1848 7f38 58c4 1a88 c86a .@.$ ...H.8X....j
00000090: 7a01 92e5 0940 5557 d301 ddf8 7c4f ba18 z....@UW....|O..
000000a0: 0048 95ee 29a9 f4f5 62d6 f137 0034 4df5 .H.. ) ...b..7.4M.
000000b0: 143c 0580 9901 80dd 68f5 ddd0 6baf cc33 .<.....h ...k..3
000000c0: 946c 099f dde5 3c0a 9f26 43c9 d6cb dfc3 .l....<..8C.....
000000d0: b3f4 7f53 4eec e4ef c1d1 7e38 8a67 38d8 ...SN.....~8.g8.
000000e0: 9f9f 3e2e 0df4 f0e9 e67f 727b d999 9047 ..>.....r{ ...G
000000f0: f1ec e223 6135 bdf0 9fbd e51c c4a7 9998 ...#a5.....
00000100: d3e7 7667 7bdb 723a e9d2 49bf 59fe 4e7e ..vg{.r:..I.Y.N~
00000110: 52d2 7bf3 bda9 77d4 631c 9208 abe5 3408 R.{ ...w.c.....4.
00000120: 4ced e27a 7a2f 3f70 acff 2df7 6853 4eaf L..zz/?p..-.hSN.
00000130: bdec 7473 b7df e2b8 fe41 0f68 0014 2cf5 ..ts.....A.h.. ,.
```

```
(hahien@hahien)-[~]  
$ xxd Fixed_Chall.png | tail -n 2  
002a0f70: 8590 b7ce 17d9 2800 0000 0049 454e 44ae .....(....IEND.  
002a0f80: 4260 82                                     B`.
```

- Kiểm tra lại với pngcheck và không có lỗi.

```
(hahien@hahien)-[~]  
$ pngcheck Fixed_Chall.png  
OK: Fixed_Chall.png (2800x2100, 24-bit RGB, non-interlaced, 84.4%).  
(hahien@hahien)-[~]
```



- Sau một lúc thực hiện kiểm tra với các tool steganography và không có kết quả, và trong bức ảnh là con appa, chúng ta thử với tool appa, decode và có được phần đầu của flag: **W1-Y0u-4r3**.

```
(hahien@hahien)-[~]  
$ git clone https://github.com/csisl/appa.git  
Cloning into 'appa' ...  
remote: Enumerating objects: 68, done.  
remote: Counting objects: 100% (6/6), done.  
remote: Compressing objects: 100% (6/6), done.  
remote: Total 68 (delta 1), reused 1 (delta 0), pack-reused 62  
Receiving objects: 100% (68/68), 619.15 KiB | 1.11 MiB/s, done.  
Resolving deltas: 100% (29/29), done.
```



```

(hahien@hahien)-[~/appa]
$ ls
appa.py      encoding.jpg      flying_appa.png
avatar_appa.png  flying_appa_new.png  README.md

(hahien@hahien)-[~/appa]
$ chmod 777 *

(hahien@hahien)-[~/appa]
$ python3 appa.py -d ~/Fixed_Chall.png
⇒ Decoding image: True

⇒ Pixel data
First 3 pixels/possible text: [(222, 223, 222), (223, 222, 223), (223, 22
3, 222)]
Possible text found at pixels[(223, 223, 222)]
Possible text found at pixels[(224, 225, 224)]
Possible text found at pixels[(224, 225, 224)]
Possible text found at pixels[(224, 225, 224)]
Possible text found at pixels[(224, 224, 224)]
Possible text found at pixels[(224, 223, 224)]
Possible text found at pixels[(224, 223, 224)]
Possible text found at pixels[(224, 224, 224)]
Possible text found at pixels[(225, 224, 224)]
.
Possible text found at pixels[(224, 225, 224)]
Possible text found at pixels[(224, 225, 224)]
Possible text found at pixels[(224, 224, 224)]
Possible text found at pixels[(224, 223, 224)]
Possible text found at pixels[(224, 223, 224)]
Possible text found at pixels[(224, 224, 224)]
Possible text found at pixels[(225, 224, 224)]

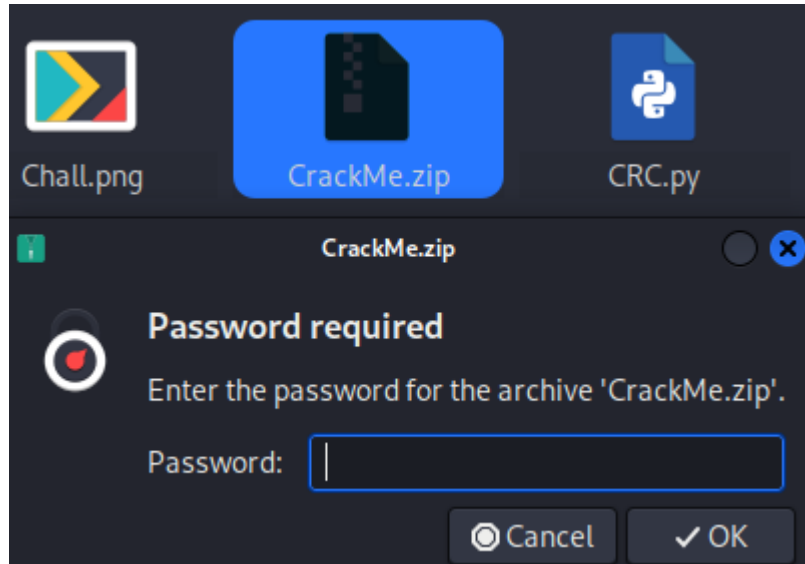
⇒ Translating each pixel for possible text
[(222, 223, 222), (223, 222, 223), (223, 223, 222), (224, 224, 223), (223
, 224, 224), (224, 225, 224), (224, 224, 225), (224, 225, 225), (224, 225, 224),
(224, 225, 224), (225, 225, 224), (224, 225, 224), (224, 224, 225), (225, 224, 22
4), (224, 224, 224), (224, 225, 225), (223, 224, 223), (224, 223, 224), (224, 224
, 223), (224, 223, 223), (224, 223, 224), (224, 224, 223), (223, 224, 223), (224
, 224, 224), (224, 225, 225), (225, 224, 224), (225, 224, 224), (224, 224, 225), (
225, 224, 224), (225, 225, 225)]
01010111 = W
00110001 = 1
00101101 = -
01011001 = Y
00110000 = 0
01110101 = u
00101101 = -
00110100 = 4
01110010 = r
00110011 = 3

W1-Y0u-4r3

(hahien@hahien)-[~/appa]

```

- Tiến đến, xem thử file zip.



- Vì không giải nén được, nên em sử dụng lệnh **unzip -lv** để hiển thị thông tin chi tiết về các tệp trong một tệp nén (tệp zip) mà không cần giải nén chúng.

```
(root@hahien)-[/home/hahien]
# unzip -lv CrackMe.zip
Archive: CrackMe.zip
Length  Method  Size  Cmpr   Date    Time    CRC-32  Name
-----  -
19      Stored  19    0%    2022-05-08 13:37  a5f3d07b  CrackMe/flag.txt
2756483 Defl:N  2754894 0%    2022-05-08 13:25  ab6a1c78  Chall.png
-----
2756502      2754913 0%
2 files
```

- Ở đây, ta thấy trong đây có file Chall.png và flag.txt, rất có thể đây là file Chall.png trước khi bị chỉnh sửa, từ đây có thể thực hiện crack file zip nhờ vào kỹ thuật plaintext-known.
- Kỹ thuật "plaintext-known" (hoặc "known-plaintext attack") là một phương pháp tấn công mật mã học trong đó kẻ tấn công có quyền truy cập vào cả văn bản gốc (plaintext) và văn bản mã hóa (ciphertext) của một hoặc nhiều thông điệp. Bằng cách sử dụng thông tin này, kẻ tấn công có thể cố gắng khám phá khóa mã hóa hoặc thu thập thêm thông tin để giải mã các thông điệp khác mã hóa bằng cùng một khóa.
- Đầu tiên, tạo 1 tệp zip mới chứa Fixed_chall.png như sau:


```
(root@hahien)-[/home/hahien]
# zip Fixed_chall Fixed_Chall.png
adding: Fixed_Chall.png (deflated 0%)

(root@hahien)-[/home/hahien]
#
```

- **Tiếp theo, kiểm tra giá trị CRC của các tệp:**

- Kiểm tra thông tin chi tiết của tệp Chall.png trong tệp zip ban đầu (CrackMe.zip) và so sánh giá trị CRC với tệp Fixed_chall.png trong tệp zip mới (Fixed_chall.zip).
- Giá trị CRC là một giá trị kiểm tra (checksum) được sử dụng để xác minh tính toàn vẹn của dữ liệu. Nó đảm bảo rằng nội dung của hai tệp là giống nhau.

```
(root@hahien)-[/home/hahien]
# zipdetails Fixed_chall.zip

000000 LOCAL HEADER #1      04034B50
000004 Extract Zip Spec     14 '2.0'
000005 Extract OS           00 'MS-DOS'
000006 General Purpose Flag 0000
[Bits 1-2]
000008 Compression Method   0008 'Deflated'
00000A Last Mod Time         58C630E4 'Thu Jun 6 02:07:08 2024'
00000E CRC                   AB6A1C78
000012 Compressed Length     002A094E
000016 Uncompressed Length   002A0F83
00001A Filename Length       000F
00001C Extra Length          001C
00001E Filename              'Fixed_Chall.png'
00002D Extra ID #0001        5455 'UT: Extended Timestamp'
00002F Length                 0009
000031 Flags                  '03 mod access'
000032 Mod Time               66618A4C 'Thu Jun 6 06:07:08 2024'
000036 Access Time            66618A4F 'Thu Jun 6 06:07:11 2024'
00003A Extra ID #0002        7875 'ux: Unix Extra Type 3'
00003C Length                 000B
00003E Version                01
00003F UID Size               04
000040 UID                    000003E8
000044 GID Size               04
000045 GID                    000003E8
000049 PAYLOAD

2A09A1 Compression Method    0008 'Deflated'
2A09A3 Last Mod Time         58C630E4 'Thu Jun 6 02:07:08 2024'
2A09A7 CRC                   AB6A1C78
2A09AB Compressed Length     002A094E
2A09AF Uncompressed Length   002A0F83
2A09B3 Filename Length       000F
2A09B5 Extra Length          0018
2A09B7 Comment Length        0000
2A09B9 Disk Start            0000
2A09BB Int File Attributes    0000
[Bit 0]
2A09BD Ext File Attributes    81A40000
2A09C1 Local Header Offset    00000000
2A09C5 Filename              'Fixed_Chall.png'
2A09D4 Extra ID #0001        5455 'UT: Extended Timestamp'

(root@hahien)-[/home/hahien]
# zipdetails CrackMe.zip

000000 LOCAL HEADER #1      04034B50
000004 Extract Zip Spec     0A '1.0'
000005 Extract OS           00 'MS-DOS'
000006 General Purpose Flag 0009
[Bit 0]
[Bit 3]
000008 Compression Method   0000 'Stored'
00000A Last Mod Time         54A86CAE 'Sun May 8 09:37:28 2022'
00000E CRC                   ASF3D07B
000012 Compressed Length     0000001F
000016 Uncompressed Length   00000013
00001A Filename Length       0010
00001C Extra Length          001C
00001E Filename              'CrackMe/flag.txt'
00002E Extra ID #0001        5455 'UT: Extended Timestamp'
000030 Length                 0009
000032 Flags                  '03 mod access'
000033 Mod Time               6277FFD7 'Sun May 8 13:37:27 2022'
000037 Access Time            6278015E 'Sun May 8 13:43:58 2022'
00003B Extra ID #0002        7875 'ux: Unix Extra Type 3'
00003D Length                 000B
00003F Version                01
000040 UID Size               04
000041 UID                    000003E8
000045 GID Size               04
000046 GID                    000003E8

000081 Compression Method    0008 'Deflated'
000083 Last Mod Time         54A86B2F 'Sun May 8 09:25:30 2022'
000087 CRC                   AB6A1C78
00008B Compressed Length     002A095A
00008F Uncompressed Length   002A0F83
000093 Filename Length       0009
000095 Extra Length          001C
000097 Filename              'Chall.png'
0000A0 Extra ID #0001        5455 'UT: Extended Timestamp'
0000A2 Length                 0009
0000A4 Flags                  '03 mod access'
0000A5 Mod Time               6277FD0A 'Sun May 8 13:25:30 2022'
0000A9 Access Time            62780120 'Sun May 8 13:42:56 2022'
0000AD Extra ID #0002        7875 'ux: Unix Extra Type 3'
0000AF Length                 000B
```

- Trong hình ảnh, chúng ta có thể thấy rằng giá trị CRC của cả hai tệp là AB6A1C78, điều này xác nhận rằng nội dung của Fixed_chall.png và Chall.png là giống nhau.
- **Tiếp theo, sử dụng pkcrack:**
- Công cụ pkcrack có thể được sử dụng để thực hiện tấn công plaintext-known như sau:
 - Sử dụng tệp zip chứa plaintext đã biết (Fixed_chall.zip với Fixed_Chall.png).
 - Sử dụng tệp zip được mã hóa (CrackMe.zip với Chall.png và flag.txt).
 - Chạy pkcrack để tìm ra khóa mã hóa được sử dụng trong CrackMe.zip.

```
(root@hahien)-[/home/hahien]
# git clone https://github.com/keyunluo/pkcrack.git

Cloning into 'pkcrack' ...
remote: Enumerating objects: 83, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 83 (delta 0), reused 2 (delta 0), pack-reused 76
Receiving objects: 100% (83/83), 147.92 KiB | 1003.00 KiB/s, done.
Resolving deltas: 100% (15/15), done.
```

```
(root@hahien)-[/home/hahien/pkcrack]
# mkdir build

(root@hahien)-[/home/hahien/pkcrack]
# cd build

(root@hahien)-[/home/hahien/pkcrack/build]
# cmake ..
CMake Deprecation Warning at CMakeLists.txt:1 (cmake_minimum_required):
  Compatibility with CMake < 3.5 will be removed from a future version of
  CMake.

  Update the VERSION argument <min> value or use a ...<max> suffix to tell
  CMake that the project does not need compatibility with older versions.

-- The C compiler identification is GNU 13.2.0
-- The CXX compiler identification is GNU 13.2.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done (1.1s)
-- Generating done (0.0s)
```

```
(root@hahien)-[/home/hahien/pkcrack/build]
# make
[ 4%] Building C object common/CMakeFiles/common.dir/crc.c.o
[ 9%] Building C object common/CMakeFiles/common.dir/debug.c.o
[ 13%] Building C object common/CMakeFiles/common.dir/exfunc.c.o
[ 18%] Building C object common/CMakeFiles/common.dir/keystuff.c.o
[ 22%] Building C object common/CMakeFiles/common.dir/mktmpdbl.c.o
[ 27%] Building C object common/CMakeFiles/common.dir/readhead.c.o
[ 31%] Building C object common/CMakeFiles/common.dir/stage1.c.o
[ 36%] Building C object common/CMakeFiles/common.dir/stage2.c.o
[ 40%] Building C object common/CMakeFiles/common.dir/stage3.c.o
[ 45%] Building C object common/CMakeFiles/common.dir/writehead.c.o
[ 50%] Building C object common/CMakeFiles/common.dir/zipdecrypt.c.o
[ 54%] Linking C static library ../lib/libcommon.a
[ 54%] Built target common
[ 59%] Building C object pkcrack/CMakeFiles/pkcrack.dir/main.c.o
[ 63%] Linking C executable /home/hahien/pkcrack/bin/pkcrack
[ 63%] Built target pkcrack
[ 68%] Building C object findkey/CMakeFiles/findkey.dir/findkey.c.o
[ 72%] Linking C executable /home/hahien/pkcrack/bin/findkey
[ 72%] Built target findkey
[ 77%] Building C object zipdecrypt/CMakeFiles/zipdecrypt.dir/zdmain.c.o
[ 81%] Linking C executable /home/hahien/pkcrack/bin/zipdecrypt
[ 81%] Built target zipdecrypt
[ 86%] Building C object extract/CMakeFiles/extract.dir/extract.c.o
[ 90%] Linking C executable /home/hahien/pkcrack/bin/extract
[ 90%] Built target extract
[ 95%] Building C object makekey/CMakeFiles/makekey.dir/makekey.c.o
[100%] Linking C executable /home/hahien/pkcrack/bin/makekey
```

```
(root@hahien)-[/home/hahien/pkcrack]
# ./bin/pkcrack -C /home/hahien/CrackMe.zip -c Chall.png -P /home/hahien/Fixed_
chall.zip -p Fixed_Chall.png -d /home/hahien/Decrypted.zip
```

```
Files read. Starting stage 1 on Thu Jun  6 09:58:40 2024
Generating 1st generation of possible key2_2754905 values ... done.
Found 4194304 possible key2-values.
Now we're trying to reduce these ...
Lowest number: 988 values at offset 2735799
Lowest number: 972 values at offset 2735794
Lowest number: 943 values at offset 2735790
Lowest number: 936 values at offset 2735335
Lowest number: 929 values at offset 2735294
Lowest number: 912 values at offset 2735289
Lowest number: 886 values at offset 2735288
Lowest number: 814 values at offset 2734609
Lowest number: 738 values at offset 2734607
Lowest number: 716 values at offset 2734603
Lowest number: 698 values at offset 2734601
Lowest number: 651 values at offset 2734600
Lowest number: 608 values at offset 2734598
```

```
Lowest number: 265 values at offset 2715591
Lowest number: 245 values at offset 2715548
Lowest number: 232 values at offset 2677033
Lowest number: 220 values at offset 2677032
Lowest number: 208 values at offset 2677029
Lowest number: 198 values at offset 2677027
Lowest number: 192 values at offset 2677013
Lowest number: 167 values at offset 2676901
Lowest number: 148 values at offset 2676899
Lowest number: 131 values at offset 2676853
Lowest number: 120 values at offset 2676851
Lowest number: 113 values at offset 2676849
Lowest number: 112 values at offset 2676837
Lowest number: 99 values at offset 2676824
Done. Left with 99 possible Values. bestOffset is 2676824.
Stage 1 completed. Starting stage 2 on Thu Jun  6 10:00:17 2024
Ta-daaaaa! key0=3fed6504, key1=c84f9c32, key2=1ef9a15e
Probabilistic test succeeded for 78086 bytes.
Ta-daaaaa! key0=3fed6504, key1=c84f9c32, key2=1ef9a15e
Probabilistic test succeeded for 78086 bytes.
Ta-daaaaa! key0=3fed6504, key1=c84f9c32, key2=1ef9a15e
Probabilistic test succeeded for 78086 bytes.
Ta-daaaaa! key0=3fed6504, key1=c84f9c32, key2=1ef9a15e
Probabilistic test succeeded for 78086 bytes.
Stage 2 completed. Starting zipdecrypt on Thu Jun  6 10:00:25 2024
Decrypting CrackMe/flag.txt (587d4c939d09af5a1309ae6c) ... OK!
Decrypting Chall.png (ea740ba832f58e3f5aa72f6b) ... OK!
Finished on Thu Jun  6 10:00:25 2024
```

```
(root@hahien)-[/home/hahien/pkcrack]
#
```



```
(root@hahien)-[/home/hahien]
# ls
 BaoMatWeb      Decrypted.zip    Fixed_chall.zip  pkcrack          Public
 Chall.png      Desktop          gatsby-serif-theme plain_chall.zip  stego.py
 CrackMe.zip    Documents        MADOC            png_header.bin   Stegsolve.
 CRC.py         Downloads        Music            png_idat.bin     Templates
 decode.py      Fixed_Chall.png  Pictures          png_iend.bin     Videos

(root@hahien)-[/home/hahien]
# unzip Decrypted.zip
Archive:  Decrypted.zip
 extracting: CrackMe/flag.txt
 replace Chall.png? [y]es, [n]o, [A]ll, [N]one, [r]ename: r
 new name: CHALL.png
 inflating: CHALL.png

(root@hahien)-[/home/hahien]
#
```

```
Open  ▼  +  flag.txt [Read-Only]
~ /CrackMe
1 -K1ng-0f-F0r3ns1cs
```

- Vậy nửa flag còn lại là: **-K1ng-0f-F0r3ns1cs**
- Kết luận, ta có **Flag: W1{Y0u-4r3-K1ng-0f-F0r3ns1cs}**

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File **.DOCX và .PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng **Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.**
- Đặt tên theo định dạng: [Mã lớp]-ExeX_GroupY. (trong đó X là Thứ tự Bài tập, Y là mã số thứ tự nhóm trong danh sách mà GV phụ trách công bố).
Ví dụ: [NT101.K11.ANTT]-Exe01_Group03.
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- **Không đặt tên đúng định dạng – yêu cầu, sẽ KHÔNG chấm điểm bài nộp.**
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá:

- Hoàn thành tốt yêu cầu được giao.
- Có nội dung mở rộng, ứng dụng.

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT