2023 암호분석경진대회

5번 문제

목차

0. 정답 관련 파일 목록

KUICS_5번/<파일 번호>: <파일 번호>에서 추출한 파일들과 사용한 스크립트를 저장한 폴더
KUICS_5번/1/1.a.7.7z: 1.png의 Alpha 7채널에서 추출한 파일
KUICS_5번/1/1.r.0.zip: 1.png의 Red 0채널에서 추출한 파일
KUICS_5번/1/blueprint.pdf: 정답 파일
KUICS_5번/1/decrypt.py: enc_blueprint 복호화 python 스크립트
KUICS_5번/1/extracted.zip: partition.vhd의 파일시스템에서 추출한 파일들
KUICS_5번/1/P.txt, P@SSW0rd.png: 1.a.7.7z에서 추출한 파일들
KUICS_5번/1/partition.info.txt: partition.vhd의 파티션 테이블을 분석한 파일
KUICS_5번/1/requirements.txt: decrypt.py를 실행하기 위해 필요한 패키지를 명시한 파일
KUICS_5번/2/2.out.7z: 2.bmp에서 추출한 파일
KUICS_5번/2/enc_3.bmp, enc_4.png, enc_5.png, README_README.txt: 2.out.7z에서 추출한 파일들
KUICS_5번/2/decryptor.py: enc_5.png 복호화 python 스크립트
KUICS_5번/2/5.png: enc_5.png를 복호화한 원본 이미지
KUICS_5번/tools: 분석 과정에서 사용한 tool들을 저장한 폴더

1. 개요

주어진 문제의 목표는 제공된 첨부파일을 분석하여 최종 도면 파일을 획득하는 것이다. 풀이 시 분석 환경은 Ubuntu 22.04 LTS 및 Windows 11을 사용한다.

2. 풀이

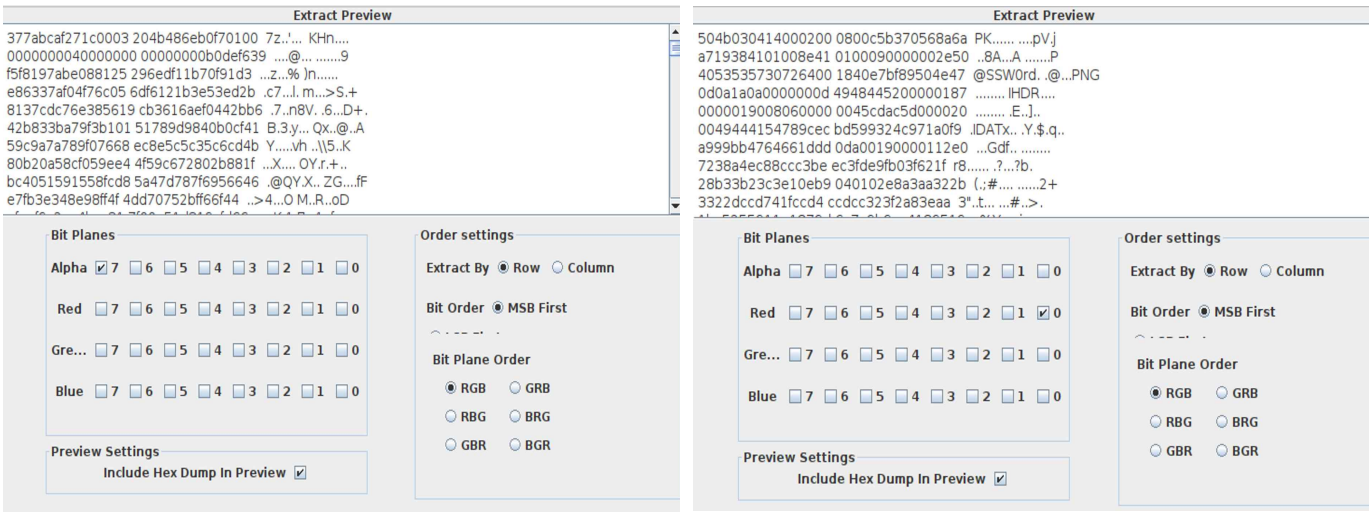
1) 첨부파일 분석

p7zip 유틸리티를 사용하여 주어진 파일을 압축 해제한다. 획득한 파일은 1.png, 2.bmp, 3.bmp, 4.png 이다.
먼저 각 파일들의 file type을 확인하자. file 명령어를 사용한 결과는 다음과 같다.

- 1.png: PNG image data, 1277 x 838, 8-bit/color RGBA, non-interlaced
- 2.bmp: JPEG image data, JFIF standard 1.01, resolution (DPI), density 150x150, segment length 16, baseline, precision 8, 1056x1072, components 3
- 3.bmp: PC bitmap, Windows 3.x format, 2110 x 1192 x 24, image size 7547744, cbSize 7547798, bits offset 54
- 4.png: PNG image data, 2842 x 2423, 8-bit/color RGBA, non-interlaced

2) 1.png 분석

2.bmp, 3.bmp 에 제시된 스테가노그래피 기법과 문제에서 주어진 힌트 중 2번째에서 이미지 색상 값의 특정 비트에 데이터가 숨겨져 있다고 추측할 수 있다. 이미지 스테가노그래피 분석 툴인 stegsolve를 이용하여 숨겨진 데이터를 추출하자.
다음과 같이 Alpha 7채널, Red 0채널에 숨겨진 파일 데이터를 확인할 수 있다.

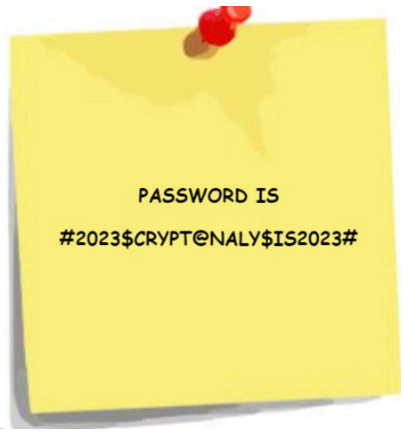


3) 1.a.7, 1.r.0 분석

Alpha 7채널에서 추출한 파일을 1.a.7, Red 0채널에서 추출한 파일을 1.r.0 로 저장한다. file 명령어로 확인한 두 파일은 각각 7z archive, zip archive 이다.

- 1.a.7: 7-zip archive data, version 0.3
- 1.r.0: Zip archive data, at least v2.0 to extract, compression method=deflate

1.r.0 파일을 unzip 유틸리티로 압축 해제하여 .P@SSW0rd 파일을 획득할 수 있다.

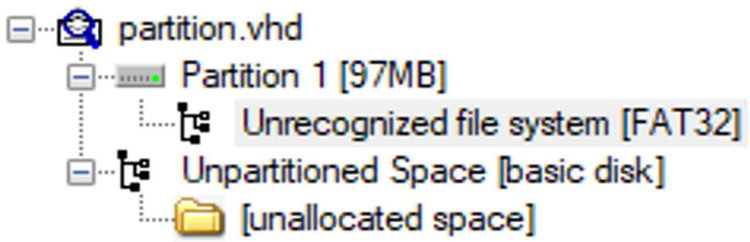


해당 파일에서 획득한 비밀번호로 1.a.7 파일을 압축 해제할 수 있다.
partition.vhd, P.txt 파일을 획득한다.

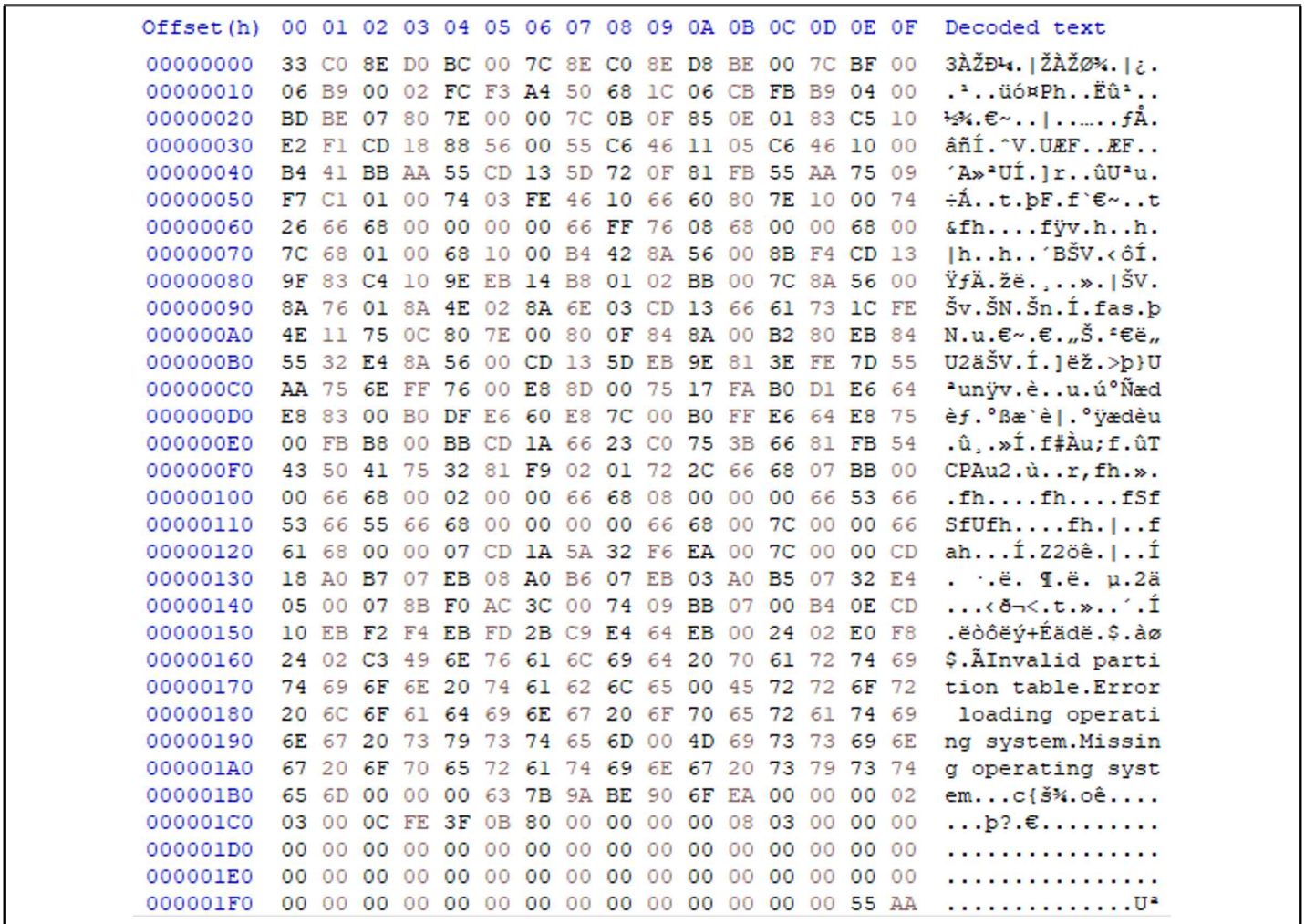
P.txt 파일의 내용은 다음과 같다. 값의 형식으로부터 위도/경도 값과 관련된 것이라 추측할 수 있다.

```
= 37.4747799,126.4097982 || 33.4978454,126.4667004 || 34.4277668, 135.2463426 || 40.6472638, -73.8122274
```

4) partition.vhd 분석



partition.vhd 파일의 경우, 디스크 이미지 분석을 위해 FTK Imager 를 사용한다. 처음 분석 시, 위와 같이 파티션이 제대로 인식되지 않기에 파티션을 복구해야 한다.



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	33	C0	8E	D0	BC	00	7C	8E	C0	8E	D8	BE	00	7C	BF	00	3ÀŽÐ¼.|ŽÀŽØ¾.|¿.
00000010	06	B9	00	02	FC	F3	A4	50	68	1C	06	CB	FB	B9	04	00	.¹..üó¤Ph..Ëû¹..
00000020	BD	BE	07	80	7E	00	00	7C	0B	0F	85	0E	01	83	C5	10	½¾.€~..|.....ƒÅ.
00000030	E2	F1	CD	18	88	56	00	55	C6	46	11	05	C6	46	10	00	âñÍ.ˆV.UÆF..ÆF..
00000040	B4	41	BB	AA	55	CD	13	5D	72	0F	81	FB	55	AA	75	09	´A»ªUÍ.]r..ûUªu.
00000050	F7	C1	01	00	74	03	FE	46	10	66	60	80	7E	10	00	74	÷Á..t.þF.f`€~..t
00000060	26	66	68	00	00	00	00	66	FF	76	08	68	00	00	68	00	&fh....fÿv.h..h.
00000070	7C	68	01	00	68	10	00	B4	42	8A	56	00	8B	F4	CD	13	|h..h..´BŠV.‹ôÍ.
00000080	9F	83	C4	10	9E	EB	14	B8	01	02	BB	00	7C	8A	56	00	ŸƒÄ.žë.¸..».|ŠV.
00000090	8A	76	01	8A	4E	02	8A	6E	03	CD	13	66	61	73	1C	FE	Šv.ŠN.Šn.Í.fas.þ
000000A0	4E	11	75	0C	80	7E	00	80	0F	84	8A	00	B2	80	EB	84	N.u.€~.€.„Š.²€ë„
000000B0	55	32	E4	8A	56	00	CD	13	5D	EB	9E	81	3E	FE	7D	55	U2äŠV.Í.]ëž.>þ}U
000000C0	AA	75	6E	FF	76	00	E8	8D	00	75	17	FA	B0	D1	E6	64	ªunÿv.è..u.ú°Ñæd
000000D0	E8	83	00	B0	DF	E6	60	E8	7C	00	B0	FF	E6	64	E8	75	èƒ.°ßæ`è|.°ÿædèu
000000E0	00	FB	B8	00	BB	CD	1A	66	23	C0	75	3B	66	81	FB	54	.û¸.»Í.f#Àu;f.ûT
000000F0	43	50	41	75	32	81	F9	02	01	72	2C	66	68	07	BB	00	CPAu2.ù..r,fh.».
00000100	00	66	68	00	02	00	00	66	68	08	00	00	00	66	53	66	.fh....fh....fSf
00000110	53	66	55	66	68	00	00	00	00	66	68	00	7C	00	00	66	SfUfh....fh.|..f
00000120	61	68	00	00	07	CD	1A	5A	32	F6	EA	00	7C	00	00	CD	ah...Í.Z2öê.|..Í
00000130	18	A0	B7	07	EB	08	A0	B6	07	EB	03	A0	B5	07	32	E4	. ·.ë. ¶.ë. µ.2ä
00000140	05	00	07	8B	F0	AC	3C	00	74	09	BB	07	00	B4	0E	CD	...‹ð¬<.t.».´.Í
00000150	10	EB	F2	F4	EB	FD	2B	C9	E4	64	EB	00	24	02	E0	F8	.ëòôëý+Éädë.$.àø
00000160	24	02	C3	49	6E	76	61	6C	69	64	20	70	61	72	74	69	$.ÃInvalid parti
00000170	74	69	6F	6E	20	74	61	62	6C	65	00	45	72	72	6F	72	tion table.Error
00000180	20	6C	6F	61	64	69	6E	67	20	6F	70	65	72	61	74	69	loading operati
00000190	6E	67	20	73	79	73	74	65	6D	00	4D	69	73	73	69	6E	ng system.Missin
000001A0	67	20	6F	70	65	72	61	74	69	6E	67	20	73	79	73	74	g operating syst
000001B0	65	6D	00	00	00	63	7B	9A	BE	90	6F	EA	00	00	00	02	em...c{š¾.oê....
000001C0	03	00	0C	FE	3F	0B	80	00	00	00	00	08	03	00	00	00	...þ?.€.........
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA	..............Uª

HxD 로 partition.vhd 를 열었을 때 확인할 수 있는 파티션 테이블을 분석하자.

- section의 마지막 2byte가 55 AA 이므로, MBR 파티션 구조이다.
- 446byte의 boot code 뒤에 있는 partiton entry를 분석하면, 1개의 파티션이 존재하고 다음과 같은 정보를 얻을 수 있다.

- Boot Indicator: Not bootable
- Starting CHS: 0x302
- Partition Type: Windows 95 with 32-bit FAT (using LBA-mode INT 13 extensions)
- Ending CHS: 0xB3FFE
- Starting LBA: 0x80 (0x10000)
- Total Sectors: 0x30800 (101 MB)

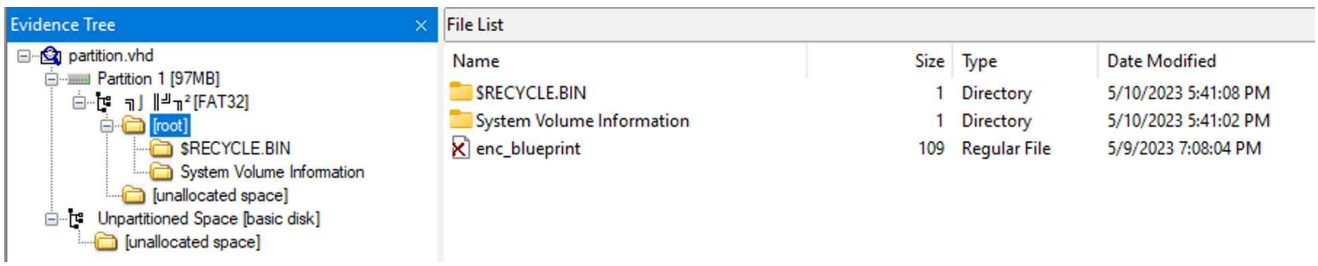
파티션의 시작 주소를 알고 있으므로 0x80=128번 sector로 이동한다.

```
00010000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................    Sector 128
00010010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000100A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000100B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000100C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000100D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000100E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000100F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010180  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000101A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000101B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000101C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000101D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000101E0  45 4E 43 52 59 50 54 45 44 5F 4F 52 5F 44 45 53  ENCRYPTED_OR_DES
000101F0  54 52 4F 59 45 44 5F 50 41 52 54 49 54 49 4F 4E  TROYED PARTITION
```

파티션이 손상되어 있지만, FAT32 파일시스템에 존재하는 backup boot sector를 활용해 원본 sector를 복구할 수 있다. FAT32 시스템에서 boot sector 다음에 나오는 RRaA 문자열을 검색했을 때, 134번 sector에 백업이 존재함을 확인할 수 있다.

```
00010C00  EB 58 90 4D 53 44 4F 53 35 2E 30 00 02 02 2E 1A  ëX.MSDOS5.0.....    Sector 134
00010C10  02 00 00 00 00 F8 00 00 3F 00 FF 00 80 00 00 00  .....ø..?.ÿ.€...
00010C20  00 08 03 00 E9 02 00 00 00 00 00 00 02 00 00 00  ....é...........
00010C30  01 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010C40  80 00 29 F6 97 25 E0 4E 4F 20 4E 41 4D 45 20 20  €.)ö—%àNO NAME
00010C50  20 20 46 41 54 33 32 20 20 20 33 C9 8E D1 BC F4    FAT32   3ÉŽÑ¼ô
00010C60  7B 8E C1 8E D9 BD 00 7C 88 56 40 88 4E 02 8A 56  {ŽÁŽÙ½.|ˆV@ˆN.ŠV
00010C70  40 B4 41 BB AA 55 CD 13 72 10 81 FB 55 AA 75 0A  @´A»ªUÍ.r..ûUªu.
00010C80  F6 C1 01 74 05 FE 46 02 EB 2D 8A 56 40 B4 08 CD  öÁ.t.þF.ë-ŠV@´.Í
00010C90  13 73 05 B9 FF FF 8A F1 66 0F B6 C6 40 66 0F B6  .s.¹ÿÿŠñf.¶Æ@f.¶
00010CA0  D1 80 E2 3F F7 E2 86 CD C0 ED 06 41 66 0F B7 C9  Ñ€â?÷â†ÍÀí.Af.·É
00010CB0  66 F7 E1 66 89 46 F8 83 7E 16 00 75 39 83 7E 2A  f÷áf‰Føƒ~..u9ƒ~*
00010CC0  00 77 33 66 8B 46 1C 66 83 C0 0C BB 00 80 B9 01  .w3f‹F.fƒÀ.».€¹.
00010CD0  00 E8 2C 00 E9 A8 03 A1 F8 7D 80 C4 7C 8B F0 AC  .è,.é¨.¡ø}€Ä|‹ð¬
00010CE0  84 C0 74 17 3C FF 74 09 B4 0E BB 07 00 CD 10 EB  „Àt.<ÿt.´.».Í.ë
00010CF0  EE A1 FA 7D EB E4 A1 7D 80 EB DF 98 CD 16 CD 19  î¡ú}ëä¡}€ëß˜Í.Í.
00010D00  66 60 80 7E 02 00 0F 84 20 00 66 6A 00 66 50 06  f`€~...„ .fj.fP.
00010D10  53 66 68 10 00 01 00 B4 42 8A 56 40 8B F4 CD 13  Sfh....´BŠV@‹ôÍ.
00010D20  66 58 66 58 66 58 66 58 EB 33 66 3B 46 F8 72 03  fXfXfXfXë3f;Før.
00010D30  F9 EB 2A 66 33 D2 66 0F B7 4E 18 66 F7 F1 FE C2  ùë*f3Òf.·N.f÷ñþÂ
00010D40  8A CA 66 8B D0 66 C1 EA 10 F7 76 1A 86 D6 8A 56  ŠÊf‹ÐfÁê.÷v.†ÖŠV
00010D50  40 8A E8 C0 E4 06 0A CC B8 01 02 CD 13 66 61 0F  @ŠèÀä..Ì¸..Í.fa.
00010D60  82 74 FF 81 C3 00 02 66 40 49 75 94 C3 42 4F 4F  ‚tÿ.Ã..f@Iu”ÃBOO
00010D70  54 4D 47 52 20 20 20 20 00 00 00 00 00 00 00 00  TMGR    ........
00010D80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010D90  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010DA0  00 00 00 00 00 00 00 00 00 00 00 00 0D 0A 44 69  ..............Di
00010DB0  73 6B 20 65 72 72 6F 72 FF 0D 0A 50 72 65 73 73  sk errorÿ..Press
00010DC0  20 61 6E 79 20 6B 65 79 20 74 6F 20 72 65 73 74   any key to rest
00010DD0  61 72 74 0D 0A 00 00 00 00 00 00 00 00 00 00 00  art.............
00010DE0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00010DF0  00 00 00 00 00 00 00 00 AC 01 B9 01 00 00 55 AA  ........¬.¹...Uª
00010E00  52 52 61 41 00 00 00 00 00 00 00 00 00 00 00 00  RRaA............    Sector 135
```

해당 sector의 데이터를 128번 sector에 복사하고 FTK Imager을 이용해 디스크 이미지를 분석한다.



삭제된 enc_blueprint 파일을 획득할 수 있다. file 명령어로 확인했을 때 아무 정보도 얻을 수 없으며, 암호화된 파일로 추측된다.

5) 2.bmp 분석

binwalk 유틸리티를 이용해 2.bmp에 숨겨진 데이터를 분석한다.

```
user@cryptocontest:~/prob$ binwalk -eM 2.bmp

Scan Time:     2023-07-21 20:15:24
Target File:   /home/user/prob/2.bmp
MD5 Checksum:  1b4bc74846221aab31d66e197bd4a411
Signatures:    411

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
180550        0x2C146         7-zip archive data, version 0.3
```

파일이 제대로 추출되지는 않았으나, 이미지 파일의 뒤에 7z archive 데이터가 함께 존재함을 확인할 수 있다. python을 이용해 해당 파일을 카빙한다.

```
_=open('2.bmp','rb').read()
open('2.out.7z','wb').write(_[0x2C146:])
```

p7zip 유틸리티로 압축 해제한 결과이다.

```
   Date      Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  ------------------------
2023-03-16 23:17:22 ....A      7547798     16951552  enc_3.bmp
2023-03-16 23:17:23 ....A      8492123               enc_4.png
2023-03-16 23:17:24 ....A      7927887               enc_5.png
2023-03-16 23:42:28 ....A         1225               README_README.txt
------------------- ----- ------------ ------------  ------------------------
2023-03-16 23:42:28           23969033     16951552  4 files
```

6) 5.png 획득

다음과 같은 가정이 성립한다면, AES CTR 모드의 작동 방식에 따라 원본 파일과 암호화된 파일을 XOR 하여 keystream을 획득하고, 이를 이용해 5.png를 복호화 할 수 있다.

- (README_README.txt 파일의 내용을 참고) enc_3.bmp, enc_4.png, enc_5.png 파일들은 3.bmp, 4.png, 5.png 파일들이 랜섬웨어(AES-256bit CTR mode)로 암호화된 파일들이다.
- (파일명을 참고) 3.bmp, 4.png 파일은 문제에서 주어진 원본 파일과 같다.
- 각 파일들은 모두 같은 keystream으로 암호화 되었다.

KUICS_5번/2/decryptor.py를 실행한 결과 가정이 성립함을 확인할 수 있고, 5.png 복호화에 성공하였다.

```
# pip install scrypt
import scrypt
salt = b"contest2023"
data = scrypt.hash(password=P,salt=salt,N=65536,r=8,p=1,buflen=48)
KEY = data[:16]
NONCE = data[16:32]
AD = data[32:]
```

문제 조건에서 “총 **4가지** 데이터가 은닉 또는 삭제” 되었다고 하였다. 현재까지 획득한 파일들을 종합해 보면 다음과 같이 4가지임을 확인할 수 있다.

- 1.png → .P@SSW0rd, P.txt, partiton.vhd(=enc_blueprint)
- 2.bmp → 5.png

7) enc_blueprint 복호화

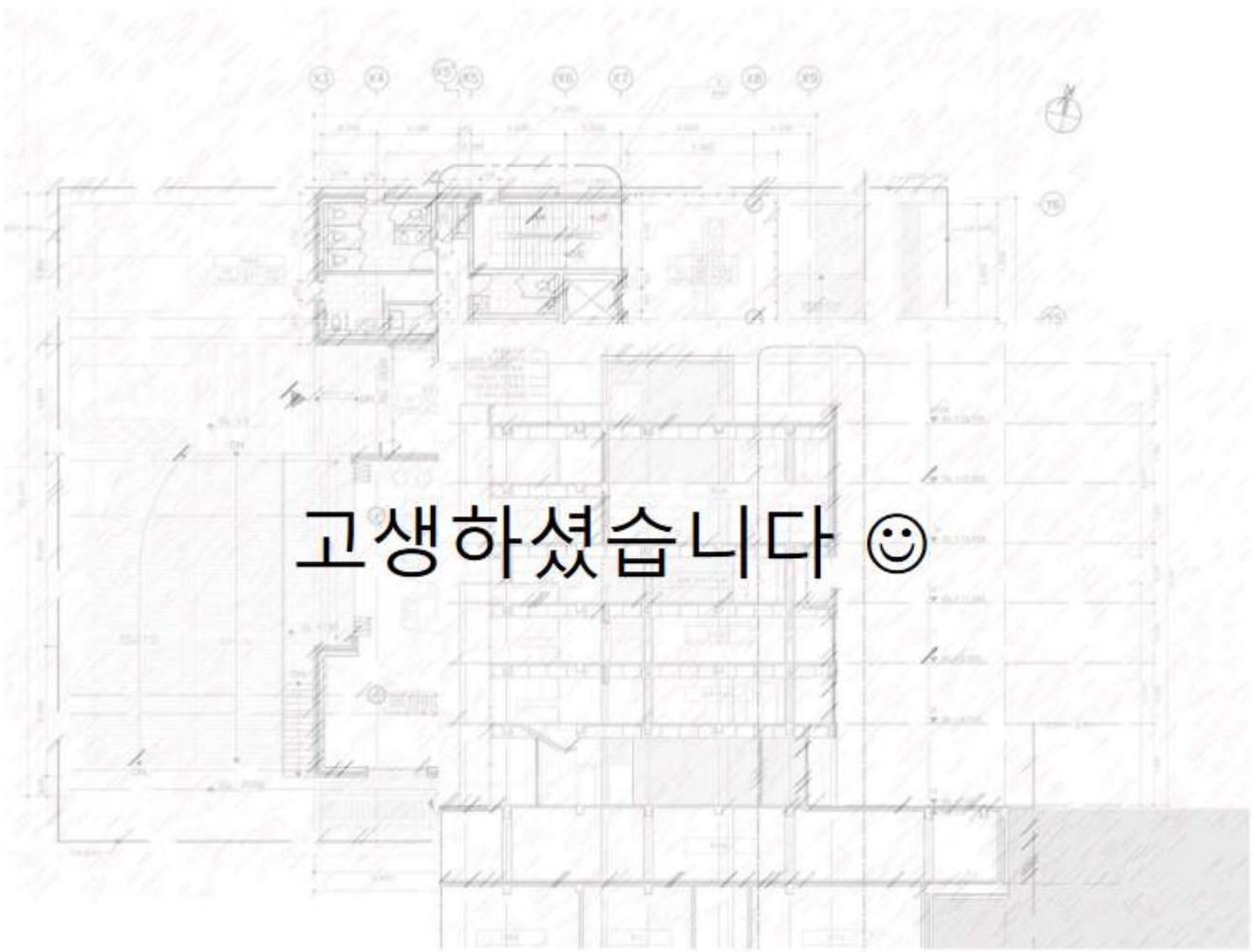
5.png에서 주어진 코드와 1.png의 내용을 참고하면, enc_blueprint 파일은 Ascon 알고리즘으로 암호화 되었음을 추측할 수 있다. 또한, 5.png 코드 중 password=P 라는 부분을 보면 password는 P.txt 파일의 내용임을 추측할 수 있다. 추측들을 바탕으로 복호화 코드(decrypt.py)를 작성하였다.

password의 경우, P.txt 의 각 좌표들이 다음과 같이 공항 어딘가의 문자열을 나타내고 있다.
사진은 Google Earth에서 캡처하였다.

IATA 공항 코드 등을 시도해 본 후, 4개의 단어를 단순히 concat 한 것(**INCHEONJEJUKANSAI31L-13R**)이 정답임을 확인했다.

3. 정답

KUICS_5번/1/decrypt.py를 실행하여 유출된 도면을 획득하였다.



4. 참고 자료

http://forensic-proof.com/archives/372
https://c0msherl0ck.github.io/file%20system/post-VBR/