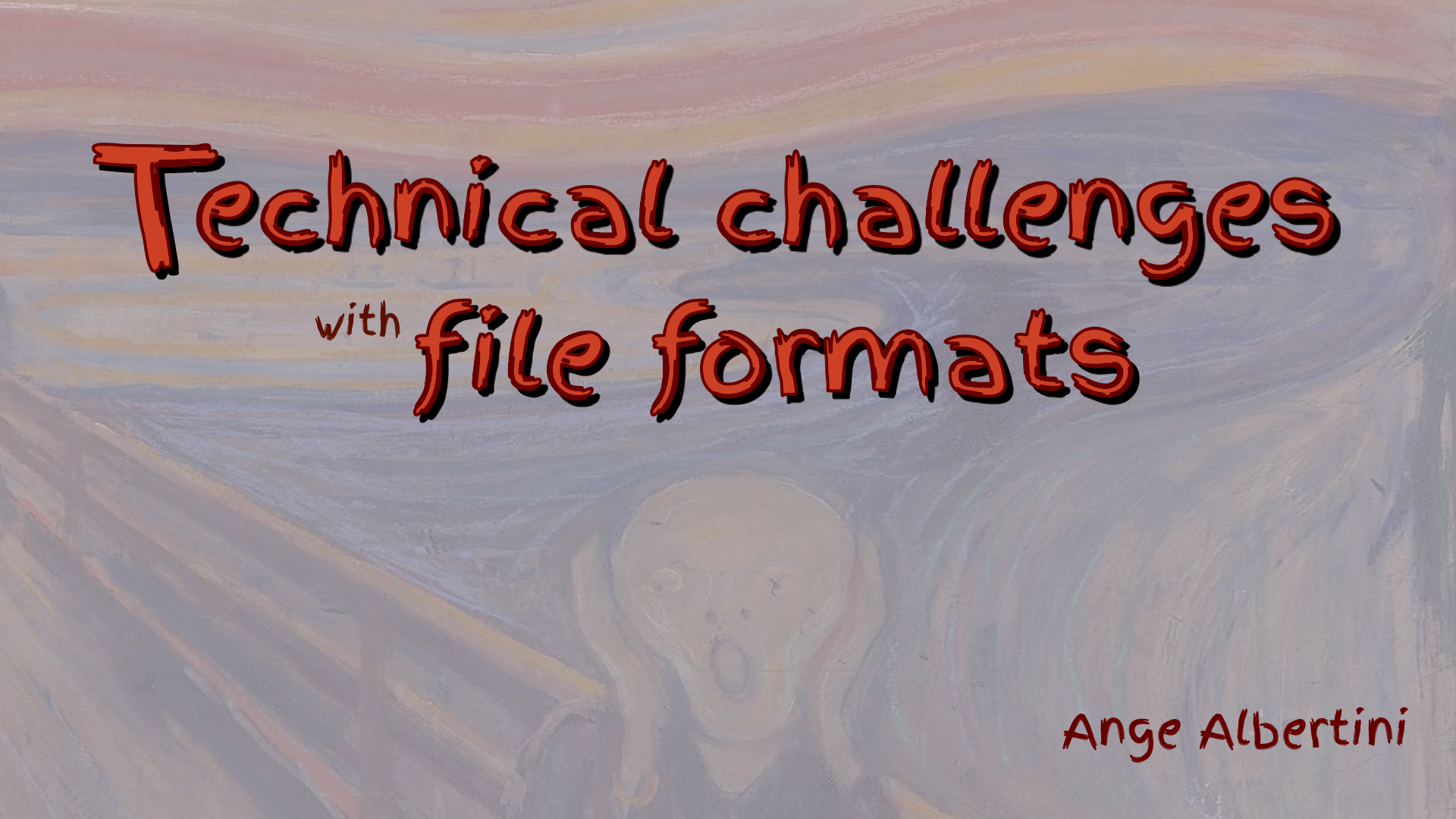

Technical challenges
with file formats
Ange Albertini

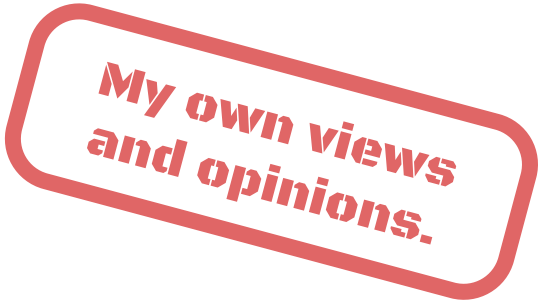
Technical challenges with file formats

MP3, Mitra, MD5

Ange Albertini
Google

15 Dec. 2022

About the author



Reverse engineering since the 80s.

Arcade games preservation at CPS2Shock.

File craft - Corkami, PoC or GTFO.

Malware analyst and infosec engineer
at Symantec, Avira, Google.

*https://github.com/angea/pocorgtfo/blob/master/README.md

Let's start "easy"...

What's an MP3 file ?

A bit of file format archeology ;)

1994: L3ENC

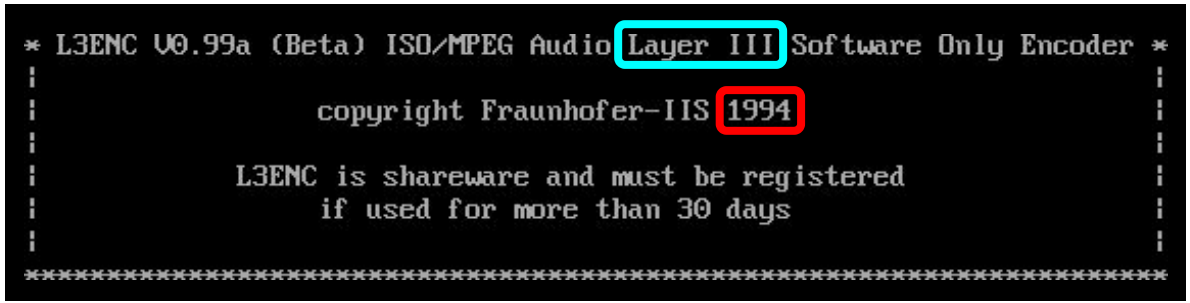
Let's call these "original MP3" files "L3" for now.

The first "Mp3" encoder. File extension: `.L3`

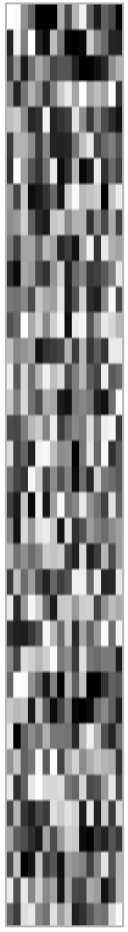
Files are raw dumps of "Layer 3" bistreams.

No header, just a sequence of frames.

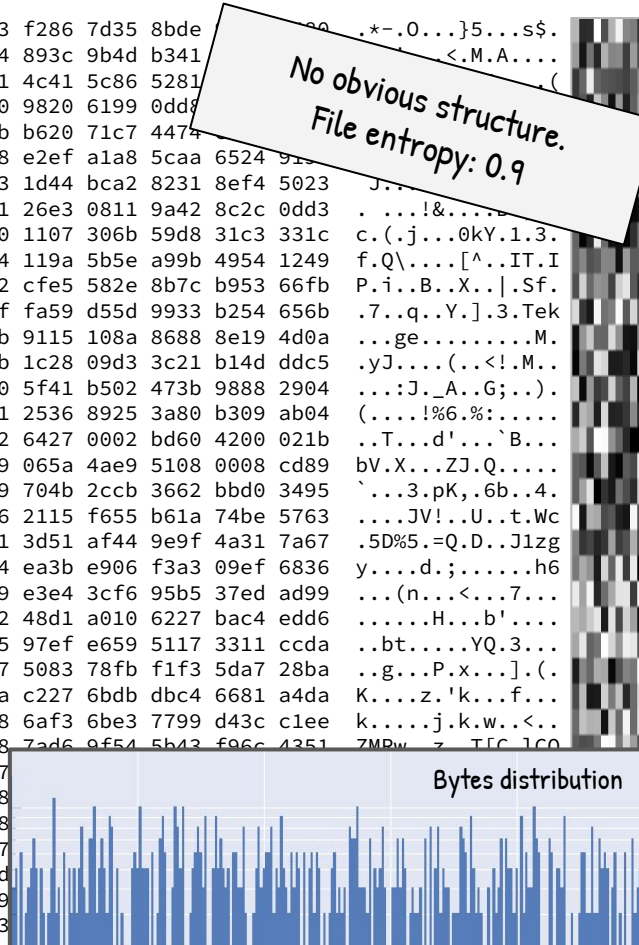
A pure data format: no file structure, no metadata.



```
0000: fffb 9064 0000 02ad 49b4 f061 1b60 4801  ...d....I..a.`H.
0010: 22f0 7fff 240a bd77 0400 08cd 896a 2422  "...$..w.....j$"
0020: 4093 1970 a888 4555 5420 0000 1228 8eb2  @..p..EUT ...(..
0030: 8a22 74b2 8931 80c8 e9a4 ffa6 bda4 fd34  ."t..1.........4
0040: bfcc 8d25 32f3 2332 3fe2 8e38 874b 2b11  ...%2.#2?..8.K+.
0050: 3cb2 b28e 3887 2323 2774 bfe6 4469 7e53  <...8.##'t..Di~S
0060: 32fc 9e55 3272 eb0c 30b4 071f e930 6b41  2..U2r..0....0kA
0070: 9744 c60d 1a34 6813 ccf9 b9c2 cd01 c5ec  .D...4h.........
0080: 8c99 cf62 2719 c6d0 88b1 b432 8949 a940  ...b'......2.I.@
0090: 7941 a4a6 67b9 a324 3d11 a8f2 34e9 afa2  yA..g..$=...4...
00a0: 2c09 a99c 4831 94ce 1f69 8338 4059 81e6  ,...H1...i.8@Y..
00b0: 7771 884c c6a9 a0e3 1c51 cf2d 1e89 f94a  wq.L.....Q.-...J
00c0: 5167 ffa9 d188 cff6 0eea f653 9ce8 d6b7  Qg.........S....
00d0: bf87 94a4 1333 3b43 b33d b636 5c33 ebeb  .....3;C.=.6\3..
00e0: ee5b d66f b6f1 bfdb 4a7f b65c 43ee 5969  .[.o....J..\C.Yi
00f0: c67e d972 4dac a51c 117c 8676 2111 6ffc  .~.rM....|.v!.o.
0100: 4b47 7cb6 ca83 eb3c 6788 7bdb 6fee f63b  KG|....<g.{.o..;
0110: d93f 6521 f1f8 c08b 9301 77e5 66f8 2f59  .?e!......w.f./Y
0120: 6f44 35f6 ca41 8b4f 820c 6c26 cf79 6c5a  oD5..A.O..l&.ylZ
0130: 411c c409 ec16 9ff9 7187 e992 7041 e064  A.......q...pA.d
0140: 8613 d894 e2ec c513 db31 529c 7f6d 8c9f  .........1R..m..
0150: b872 906c 86ca c4cc 4221 da1d 3db6 4d9d  .r.l....B!..=.M.
0160: 3bbf 6c9d 4d21 663d 53ef ef67 ea25 27e7  ;.l.M!f=S..g.%'.
0170: e729 98f7 d169 1cc7 cb9c cdd1 4490 bdab  .)...i......D...
0180: 331f 1d35 57f8 bb6b c82a a661 99fc 6ff3  3..5W..k.*.a..o.
0190: 8944 dbcf b7ce fb78 8e3c de76 6b7a 7a1e  .D.....x.<.vkzz.
01a0: 65ff fb92 6417 8002 b461 c200 0133 6256  e...d....a...3bV
01b0: c8b8 c018 c35c 0b41 7f08 000c 6d89 5f1f  .....\.A....m._.
01c0: e300 618d 71b2 7776 722a b7ef b04e 2370  ..a.q.wvr*...N#p
01d0: 93c4 6276 f75c fd42 add1 8cad c4b5 af36  ..bv.\.B.......6
01e0: e62e 62e6 fed7 d6dc 7867 d649 5399 e64b  ..b.....xg.IS..K
01f0: 48b3 341e 4c3a deca e2d0 20e7 21ca 3cdb  H.4.L:.... .!.<.
0200: a65c 4c2f 18a4 4d9b d0ca 0c04 3821 5538  .\L/..M.....8!U8
0210: 49c1 0015 c23b 560f 9b8e 8441 85e4 53b4  I....;V....A..S.
0220: eb50 e987 d089 a115 a648 42a9 9b11 44b7  .P.......HB...D.
0230: 653c dde8 8771 e19c c720 3058 7972 d9fa  e<...q... 0Xyr..
```



```
0240: b62a 2dff 4fb3 f286 7d35 8bde  .*-.O...}5...s$.
0250: 9a15 7427 b084 893c 9b4d b341  ..t'...<.M.A....
0260: 3866 ec63 5651 4c41 5c86 5281  8f.cVQLA\.R.(...
0270: 3958 24c5 8a10 9820 6199 0dd8  9X$..... a.....
0280: cf1c cc90 5d6b b620 71c7 4474  ....]k. q.Dt...
0290: ce76 93ea 4268 e2ef a1a8 5caa 6524 91..  .v..Bh....\.e$..
02a0: 204a fb0a e253 1d44 bca2 8231 8ef4 5023  J...S.D...1..P#
02b0: 0a20 18c2 e021 26e3 0811 9a42 8c2c 0dd3  . ...!&....B.,..
02c0: 6304 28ea 6a10 1107 306b 59d8 31c3 331c  c.(.j...0kY.1.3.
02d0: 6686 515c 8584 119a 5b5e a99b 4954 1249  f.Q\....[^..IT.I
02e0: 50d1 6993 8d42 cfe5 582e 8b7c b953 66fb  P.i..B..X..|.Sf.
02f0: da37 17f6 717f fa59 d55d 9933 b254 656b  .7..q..Y.].3.Tek
0300: 1194 c867 65db 9115 108a 8688 8e19 4d0a  ...ge.........M.
0310: 0479 4ae4 d51b 1c28 09d3 3c21 b14d ddc5  .yJ....(..<!.M..
0320: c20a 983a 4ad0 5f41 b502 473b 9888 2904  ...:J._A..G;..).
0330: 28b8 a61f d821 2536 8925 3a80 b309 ab04  (....!%6.%:.....
0340: accc 54ff fb92 6427 0002 bd60 4200 021b  ..T...d'...`B...
0350: 6256 0758 c019 065a 4ae9 5108 0008 cd89  bV.X...ZJ.Q.....
0360: 60a2 e300 3319 704b 2ccb 3662 bbd0 3495  `...3.pK,.6b..4.
0370: f917 b0a8 4a56 2115 f655 b61a 74be 5763  ....JV!..U..t.Wc
0380: 8235 4425 35a1 3d51 af44 9e9f 4a31 7a67  .5D%5.=Q.D..J1zg
0390: 79ab 921a d564 ea3b e906 f3a3 09ef 6836  y....d.;......h6
03a0: eba2 f328 6ea9 e3e4 3cf6 95b5 37ed ad99  ...(n...<...7...
03b0: b9ab cae5 1be2 48d1 a010 6227 bac4 edd6  ......H...b'....
03c0: a6e6 6274 fdd5 97ef e659 5117 3311 ccda  ..bt.....YQ.3...
03d0: 1c87 67ac a707 5083 78fb f1f3 5da7 28ba  ..g...P.x...](.
03e0: 4bba 171b d07a c227 6bdb dbc4 6681 a4da  K....z.'k...f...
03f0: 6bc1 b1b3 12c8 6af3 6be3 7799 d43c c1ee  k.....j.k.w..<..
0400: 5a4d 5277 fcf8 7ad6 9f54 5b43 f96c 4351  ZMRw..z..T[C.lCQ
0410: 6d84 9b09 b697 ...  m.....
0420: 82c4 ec80 2aa8 ...  ....*.
0430: 27a5 5063 9128 ...  '.Pc.(
0440: 9a6f 6e96 1547 ...  .on..G
0450: 4b45 9342 b21d ...  KE.B..
0460: 43df cacf d5d9 ...  C.....
0470: 0611 7efe aa93 ...  ..~...
```



No obvious structure.
File entropy: 0.9

50ms of **silence**, encoded by the original L3enc

```
0000: fffb 9064 0000 02ad 49b4 f061 1b60 4801  ...d....I..a.`H.
0010: 22f0 7fff 240a bd77 0400 08cd 896a 2422  "...$..w.....j$"
0020: 4093 1970 a888 4555 5420 0000 1228 8eb2  @..p..EUT ...(..
0030: 8a22 74b2 8931 80c8 e9a4 ffa6 bda4 fd34  ."t..1.........4
0040: bfcc 8d25 32f3 2332 3fe2 8e38 874b 2b11  ...%2.#2?..8.K+.
0050: 3cb2 b28e 3887 2323 2774 bfe6 4469 7e53  <...8.##'t..Di~S
0060: 32fc 9e55 3272 eb0c 30b4 071f e930 6b41  2..U2r..0....0kA
0070: 9744 c60d 1a34 6813 ccf9 b9c2 cd01 c5ec  .D...4h.........
0080: 8c99 cf62 2719 c6d0 88b1 b432 8949 a940  ...b'......2.I.@
0090: 7941 a4a6 67b9 a324 3d11 a8f2 34e9 afa2  yA..g..$=...4...
00a0: 2c09 a99c 4831 94ce 1f69 8338 4059 81e6  ,...H1...i.8@Y..
00b0: 7771 884c c6a9 a0e3 1c51 cf2d 1e89 f94a  wq.L.....Q.-...J
00c0: 5167 ffa9 d188 cff6 0eea f653 9ce8 d6b7  Qg.........S....
00d0: bf87 94a4 1333 3b43 b33d b636 5c33 ebeb  .....3;C.=.6\3..
00e0: ee5b d66f b6f1 bfdb 4a7f b65c 43ee 5969  .[.o....J..\C.Yi
00f0: c67e d972 4dac a51c 117c 8676 2111 6ffc  .~.rM....|.v!.o.
0100: 4b47 7cb6 ca83 eb3c 6788 7bdb 6fee f63b  KG|....<g.{.o..;
0110: d93f 6521 f1f8 c08b 9301 77e5 66f8 2f59  .?e!......w.f./Y
0120: 6f44 35f6 ca41 8b4f 820c 6c26 cf79 6c5a  oD5..A.O..l&.ylZ
0130: 411c c409 ec16 9ff9 7187 e992 7041 e064  A.......q...pA.d
0140: 8613 d894 e2ec c513 db31 529c 7f6d 8c9f  .........1R..m..
0150: b872 906c 86ca c4cc 4221 da1d 3db6 4d9d  .r.l....B!..=.M.
0160: 3bbf 6c9d 4d21 663d 53ef ef67 ea25 27e7  ;.l.M!f=S..g.%'.
0170: e729 98f7 d169 1cc7 cb9c cdd1 4490 bdab  .)...i......D...
0180: 331f 1d35 57f8 bb6b c82a a661 99fc 6ff3  3..5W..k.*.a..o.
0190: 8944 dbcf b7ce fb78 8e3c de76 6b7a 7a1e  .D.....x.<.vkzz.
01a0: 65ff fb92 6417 8002 b461 c200 0133 6256  e...d....a...3bV
01b0: c8b8 c018 c35c 0b41 7f08 000c 6d89 5f1f  .....\.A....m._.
01c0: e300 618d 71b2 7776 722a b7ef b04e 2370  ..a.q.wvr*...N#p
01d0: 93c4 6276 f75c fd42 add1 8cad c4b5 af36  ..bv.\.B.......6
01e0: e62e 62e6 fed7 d6dc 7867 d649 5399 e64b  ..b.....xg.IS..K
01f0: 48b3 341e 4c3a deca e2d0 20e7 21ca 3cdb  H.4.L:.... .!.<.
0200: a65c 4c2f 18a4 4d9b d0ca 0c04 3821 5538  .\L/..M.....8!U8
0210: 49c1 0015 c23b 560f 9b8e 8441 85e4 53b4  I....;V....A..S.
0220: eb50 e987 d089 a115 a648 42a9 9b11 44b7  .P.......HB...D.
0230: 653c dde8 8771 e19c c720 3058 7972 d9fa  e<...q... 0Xyr..
0240: b62a 2dff 4fb3 f286 7d35 8bde 9273 2480  .*-.O...}5...s$.
0250: 9a15 7427 b084 893c 9b4d b341 cbd5 aba0  ..t'...<.M.A....
0260: 3866 ec63 5651 4c41 5c86 5281 0eaa 8a28  8f.cVQLA\.R....(
0270: 3958 24c5 8a10 9820 6199 0dd8 30f7 1f40  9X$.... a...0..@
0280: cf1c cc90 5d6b b620 71c7 4474 ca90 0c27  ....]k. q.Dt...'
0290: ce76 93ea 4268 e2ef a1a8 5caa 6524 919b  .v..Bh....\.e$..
02a0: 204a fb0a e253 1d44 bca2 8231 8ef4 5023   J...S.D...1..P#
02b0: 0a20 18c2 e021 26e3 0811 9a42 8c2c 0dd3  . ...!&....B.,..
02c0: 6304 28ea 6a10 1107 306b 59d8 31c3 331c  c.(.j...0kY.1.3.
02d0: 6686 515c 8584 119a 5b5e a99b 4954 1249  f.Q\....[^..IT.I
02e0: 50d1 6993 8d42 cfe5 582e 8b7c b953 66fb  P.i..B..X..|.Sf.
02f0: da37 17f6 717f fa59 d55d 9933 b254 656b  .7..q..Y.].3.Tek
0300: 1194 c867 65db 9115 108a 8688 8e19 4d0a  ...ge.........M.
0310: 0479 4ae4 d51b 1c28 09d3 3c21 b14d ddc5  .yJ....(..<!.M..
0320: c20a 983a 4ad0 5f41 b502 473b 9888 2904  ...:J._A..G;..).
0330: 28b8 a61f d821 2536 8925 3a80 b309 ab04  (....!%6.%:.....
0340: accc 54ff fb92 6427 0002 bd60 4200 021b  ..T...d'...`B...
0350: 6256 0758 c019 065a 4ae9 5108 0008 cd89  bV.X...ZJ.Q.....
0360: 60a2 e300 3319 704b 2ccb 3662 bbd0 3495  `...3.pK,.6b..4.
0370: f917 b0a8 4a56 2115 f655 b61a 74be 5763  ....JV!..U..t.Wc
0380: 8235 4425 35a1 3d51 af44 9e9f 4a31 7a67  .5D%5.=Q.D..J1zg
0390: 79ab 921a d564 ea3b e906 f3a3 09ef 6836  y....d.;......h6
03a0: eba2 f328 6ea9 e3e4 3cf6 95b5 37ed ad99  ...(n...<...7...
03b0: b9ab cae5 1be2 48d1 a010 6227 bac4 edd6  ......H...b'....
03c0: a6e6 6274 fdd5 97ef e659 5117 3311 ccda  ..bt.....YQ.3...
03d0: 1c87 67ac a707 5083 78fb f1f3 5da7 28ba  ..g...P.x...].(.
03e0: 4bba 171b d07a c227 6bdb dbc4 6681 a4da  K....z.'k...f...
03f0: 6bc1 b1b3 12c8 6af3 6be3 7799 d43c c1ee  k.....j.k.w..<..
0400: 5a4d 5277 fcf8 7ad6 9f54 5b43 f96c 4351  ZMRw..z..T[C.lCQ
0410: 6d84 9b09 b697 00b4 10c5 cf5b 5852 80c6  m..........[XR..
0420: 82c4 ec80 2aa8 c96e cedb 262a 4de8 1f97  ....*..n..&*M...
0430: 27a5 5063 9128 093d b75a 248f a9e6 790f  '.Pc.(.=.Z$...y.
0440: 9a6f 6e96 1547 466d 0fd5 51ac 9152 93b3  .on..GFm..Q..R..
0450: 4b45 9342 b21d b25e e133 3ea6 d96a c2fd  KE.B...^.3>..j..
0460: 43df cacf d5d9 58d8 cf3b 9e33 a4c1 e186  C.....X..;.3....
0470: 0611 7efe aa93                           ..~...
```

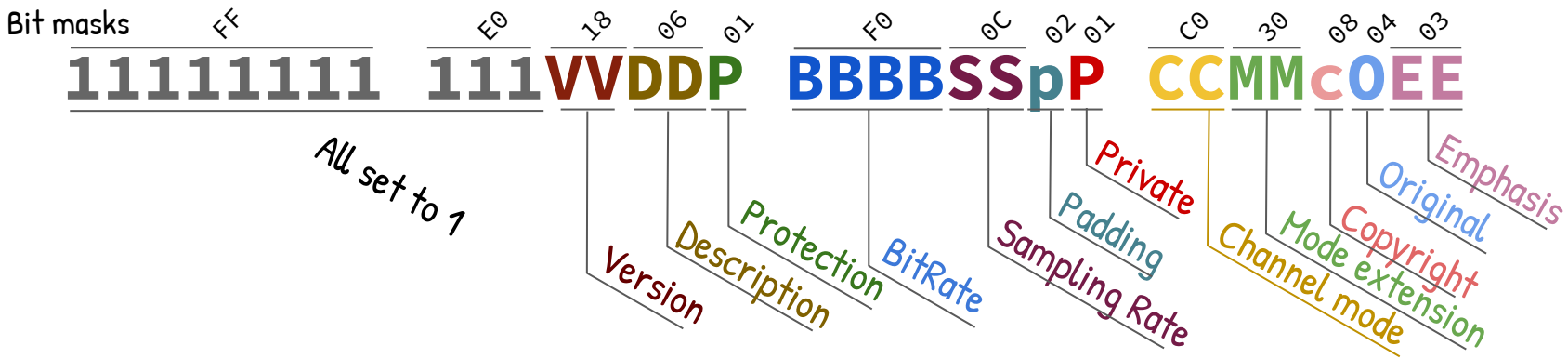
File contents: 3 Layer3 frames

Layer3 frames

Each frame has a 4 bytes header
that starts with FF E? or FF F?

Examples of frame headers:

```
FF FB 90 64
FF FB A0 44
FF FB E0 40
FF FB 50 00
FF .. .. ..
```



How do you even identify such a file?

By heuristic

No clear identification:

Locate FF. Read parameters from the 3rd byte.
Compute frame length. Check for FF at the next offset.
Repeat enough times.

$$Length_{Frame} = 144 * \frac{BitRate}{SampleRate} + Padding$$

```
11111111 111..... BBBBSSp. ........


Bitrate  kbps
 0000     free
 0001     32
 0010     40
 0011     48
 0100     56
 0101     64
 0110     80
 0111     96
 1000     112
 1001     128
 1010     160
 1011     192
 1100     224
 1101     256
 1110     320
 1111     bad


Sampling   Hz
 00         44100
 01         48000
 10         32000
 11         res.

From 104 (0x68) to 14400 (0x3840)
```

FFMPEG

```
$ ffprobe -show-frames 50ms.mp3
[...]
[mp3 @ 0000000002653600] Format mp3 detected only with low score of 1, misdetection possible!
[mp3 @ 0000000002653600] Estimating duration from bitrate, this may be inaccurate
Input #0, mp3, from '50ms':
  Duration: 00:00:00.07, start: 0.000000, bitrate: 128 kb/s
    Stream #0:0: Audio: mp3, 44100 Hz, stereo, s16p, 128 kb/s
[FRAME]
```

MPG123

```
1068            if(!hd)
1069            {
1070                    if(NOQUIET) warning("Cannot read next header, a one-frame stream? Duh...");
1071                    return PARSE_END;
1072            }
```

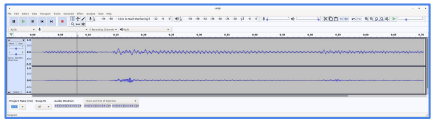
Too few L3 frames can't be reliably identified!

Expected behavior

I should see a muscular girl in the JPEG file

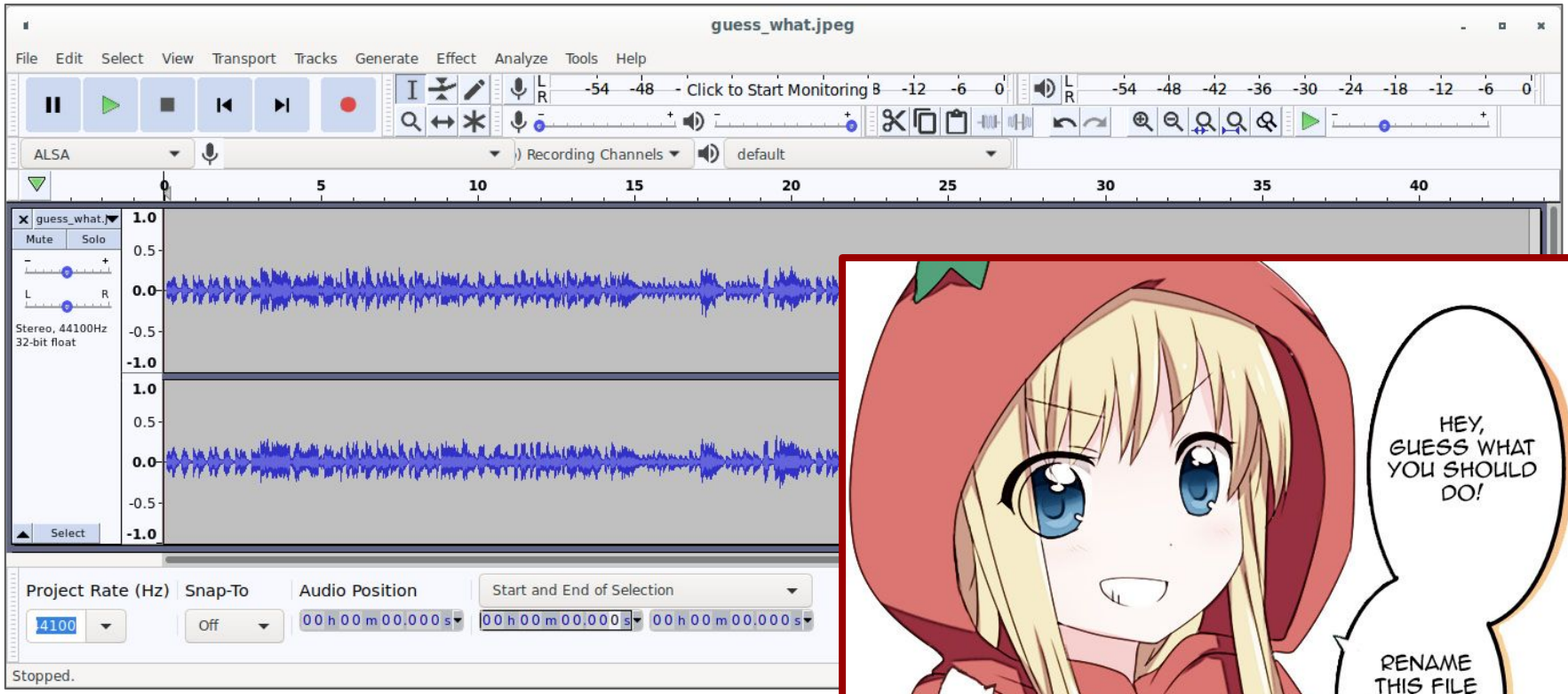
Actual behavior

I hear industrial music instead



IDENTIFY FFS... YOU MEAN LIKE JPG ? -> FUNNY ACCIDENTS...

https://github.com/mpv-player/mpv/issues/3973



guess_what.jpeg

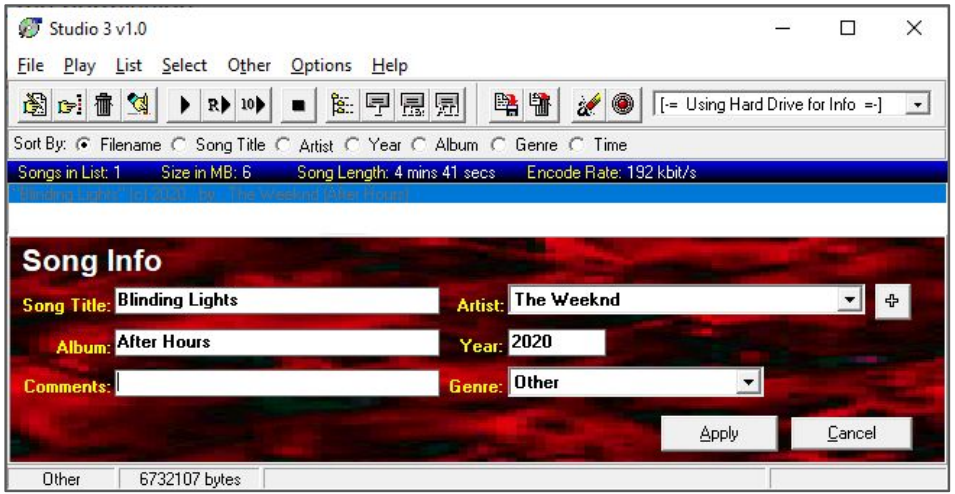
HEURISTIC PARSING
-> SKIPPED JPG HEADER
-> JPG/MP3 POLYGLOT

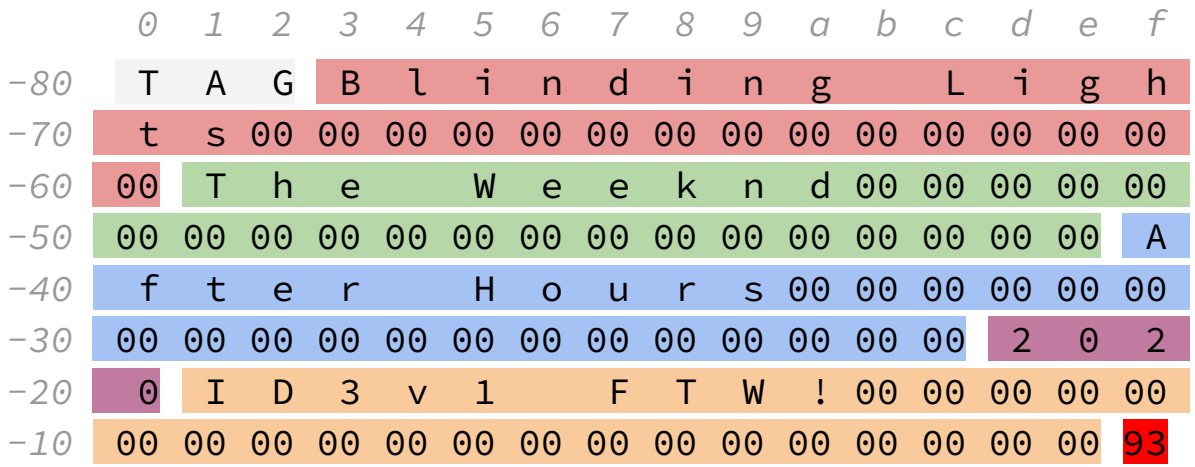
...OR ABUSES!

1996: ID3v1 HACK

L3 = pure data format -> no metadata

-> "Id3v1" footer hack





	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
-80	T	A	G	B	l	i	n	d	i	n	g		L	i	g	h
-70	t	s	00	00	00	00	00	00	00	00	00	00	00	00	00	00
-60	00	T	h	e		W	e	e	k	n	d	00	00	00	00	00
-50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	A
-40	f	t	e	r		H	o	u	r	s	00	00	00	00	00	00
-30	00	00	00	00	00	00	00	00	00	00	00	00	00	2	0	2
-20	0	I	D	3	v	1		F	T	W	!	00	00	00	00	00
-10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	93

Field	Size
TAG	3
Song	30
Artist	30
Album	30
Year	4
Comment	30
Genre	1

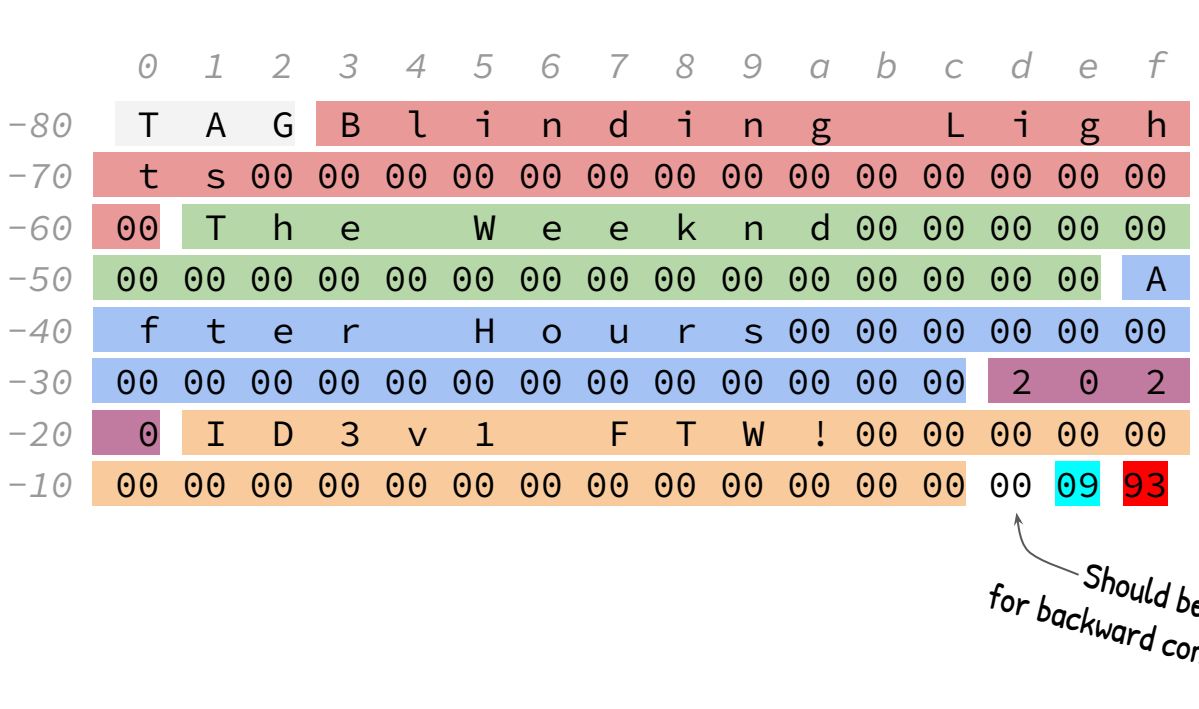
Total: 128

1996: ID3v1

A 128 byte long **footer** (de-facto "standard")
with **hardcoded** lengths!

-> not extendable

Cf ID3 tag version 1

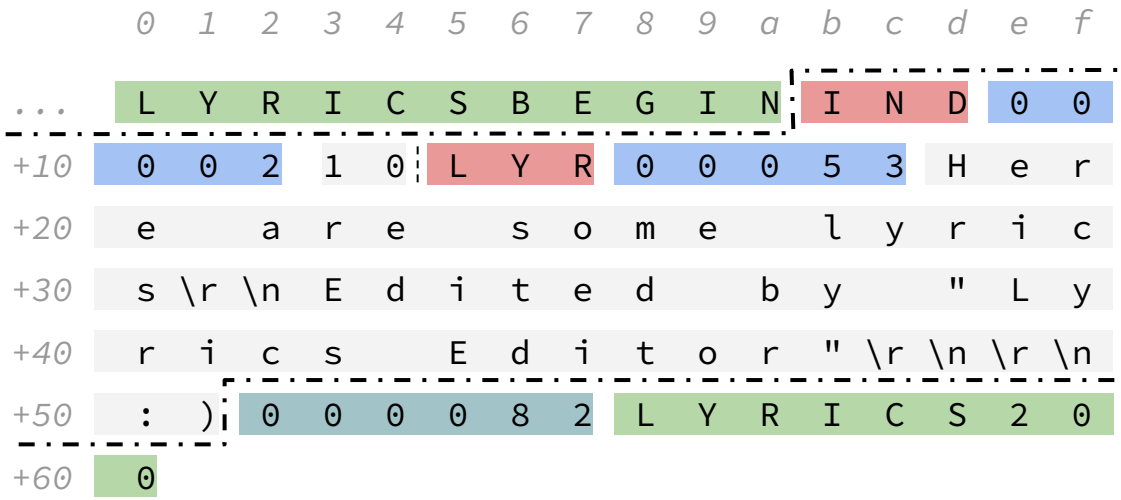


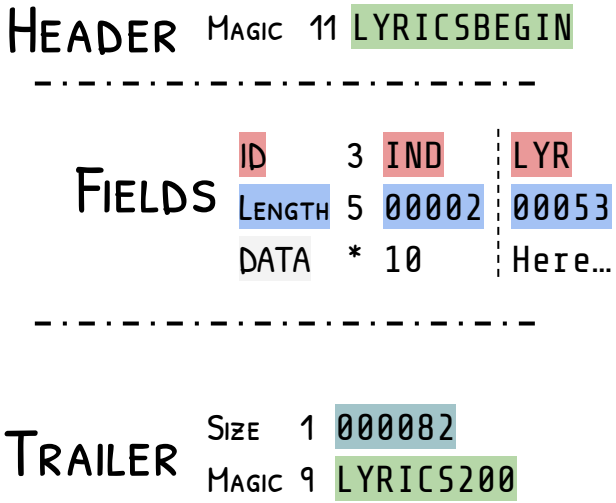
Field	Size
TAG	3
Song	30
Artist	30
Album	30
Year	4
Comment	28
TrackNb	1
Genre	1

Total: 128

1997: ID3v1.1 – a hack of a hack…

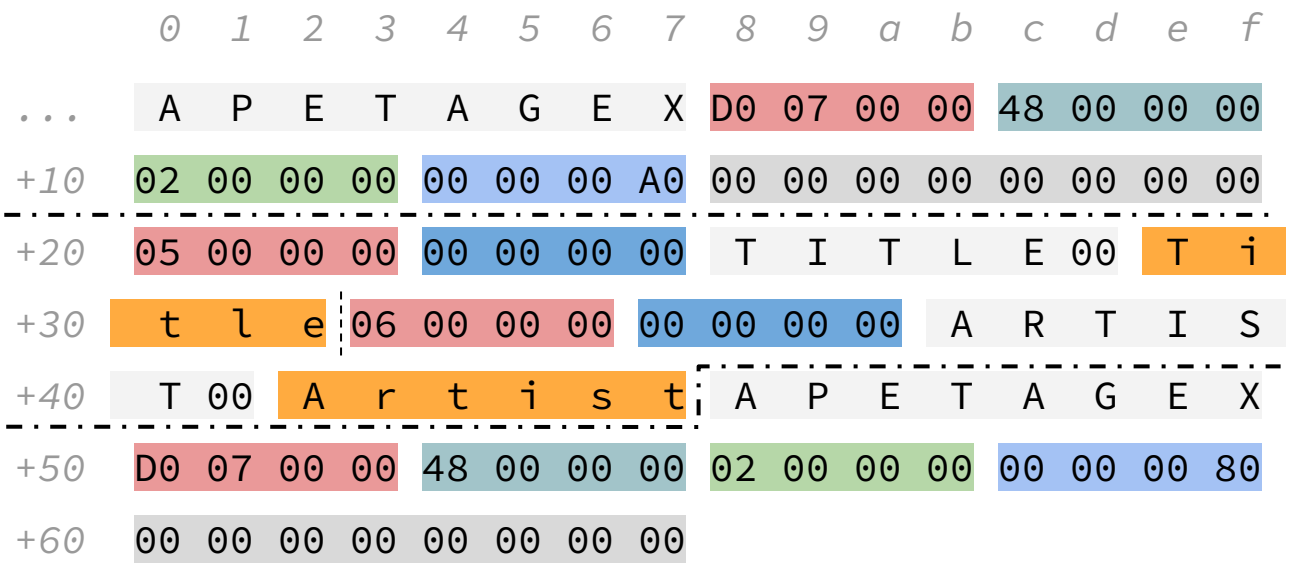
Shortened comment -> 1 byte for TrackNb





Extra footers were then defined
(to be appended before the ID3v1.x)

Ex: the Lyrics3 footer…



Header

Preamble	8	APE TAG EX
Version	4	2.0 (0x7d0)
Size	4	0x48
Count	4	02
Flags	4	bHdr/isHeader
Reserved	8	0

Item x2

Size	4	05	06
Flags	4	00	00
Key	0	TITLE\0	ARTIST\0
Value	Size	Title	Artist

Size

Footer

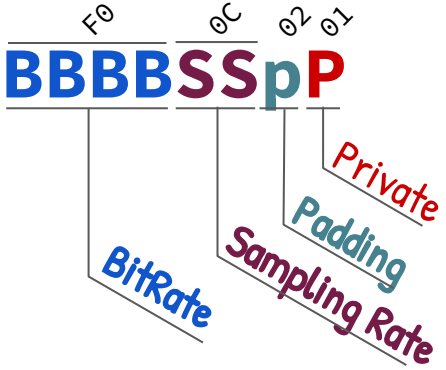
Preamble	8	APE TAG EX
Version	4	2.0 (0x7d0)
Size	4	0x48
Count	4	02
Flags	4	bHdr/isFooter
Reserved	8	0

Like the header but with different flags

...the APEv2 footer...

...and so on...

Constant / Variable Bitrate?



Original MP3 = CBR
-> Silences are encoded like the rest...

Each frame has its own bitrate -> VBR possible
but duration and seeking are broken.

-> Store frame index as the "Xing" structure
in the **first frame**!

```
00   FF E3 48 C4 00 00 00 00 00 00 00 00 00 X  i  n
10    g  00 00 00 0F 00 00 00 80 00 00 25 20 00 04 06
     [...]
```

LAME also does that in further frames.

Since Layer3 didn't allow metadata...

A BIG MESS OF MANY STRUCTURES!

No official header...

In-frame: Xing, Lame

Footers: APEv2, Lyrics3, TAG+, ID3v1

MP3 (pure data)

```
0000: ff f3 90 64 00 11 c9 f7 49 2f c6 c8 00 11 f2 2a
0010: 9c cb 87 50 00 14 08 03 01 00 80 a0 40 20 1c 00
0020: 00 00 0d e0 3d 89 e6 f1 e0 16 1d 89 20 e6 7d c3
      [...]
```

Same Layer3 bitstreams used in 3 different containers...

RIFF/WAV with MP3-data

```
0000:  R  I  F  F fa 3d 05 00  W  A  V  E  f  m  t 20
0010: 1e 00 00 00 55 00 02 00 22 56 00 00 10 27 00 00
0020: 01 00 00 00 0c 00 01 00 02 00 00 00 01 00 01 00
0030: 71 05  f  a  c  t 04 00 00 00 f8 2f 14 00  d  a
0040:  t  a d6 3d 05 00 ff f3 90 64 00 11 c9 f7 49 2f
0050: c6 c8 00 11 f2 2a 9c cb 87 50 00 14 08 03 01 00
0060: 80 a0 40 20 1c 00 00 00 0d e0 3d 89 e6 f1 e0 16
      [...]
```

0055: WAVE_FORMAT_MPEGLAYER3

RIFF/MP3

```
0000:  R  I  F  F 72 3f 05 00  R  M  P  3  d  a  t  a
0010: d6 3d 05 00 ff f3 90 64 00 11 c9 f7 49 2f c6 c8
0020: 00 11 f2 2a 9c cb 87 50 00 14 08 03 01 00 80 a0
0030: 40 20 1c 00 00 00 0d e0 3d 89 e6 f1 e0 16 1d 89
     [...]
```

RIFF is a container format used in WAV, AVI, ASF, WebP...

Raw Layer3 streams could also be eventually in 'proper' formats. However...

1998: ID3v2. "At last" a proper **header** ?

A proper "header" with <u>variable</u> lengths!

```
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00  I  D  3 03 00 00 00 00 00 2F T  P  E  1 00 00
10 00 07 00 00 00 A  r  t  i  s  t  T  A  L  B 00
20 00 00 06 00 00 00 A  l  b  u  m  T  Y  E  R 00
30 00 00 05 00 00 00 Y  e  a  r  T  I  T  2 00 00
40 00 05 00 00 00 N  a  m  e
```

Header

Magic	3	ID3
Version	2	3.0
Flags	1	00
Size	4	2F

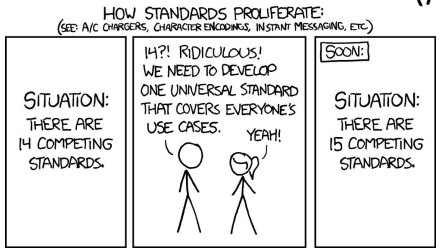
Frames

ID	4	TPE1
Size	4	7
Flags	2	00 00
Data	*	00 ARTIST

Lead **Per**former (text frame)

Encoding

...but why not using RIFF(MP3)? 🤷



ID3v2 ID3 tag version 2.3.0

This file is artificially small:
a single frame (0.025s) with a single metadata entry.

And yet…one header, 2 footers.

The same metadata present in 3 structures.

None of these structures is aware of the rest:
just concatenated structures knowing their own size.

Format redundancy dilemma: which one is right?

```
000  .I .D .3 03 00 00 00 00 00 10 .T .I .T .2 00 00
010  00 06 00 00 00 .S .o .n .g 00 FF FB 90 64 00 00
020  02 AD 49 B4 F0 61 1B 60 48 01 22 F0 7F FF 24 0A
030  BD 77 04 00 08 CD 89 6A 24 22 40 93 19 70 A8 88
040  45 55 54 20 00 00 00 12 28 8E B2 8A 22 74 B2 89 31
050  80 C8 E9 A4 FF A6 BD A4 FD 34 BF CC 8D 25 32 F3
060  23 32 3F E2 8E 38 87 4B 2B 11 3C B2 B2 8E 38 87
070  23 23 27 74 BF E6 44 69 7E 53 32 FC 9E 55 32 72
080  EB 0C 30 B4 07 1F E9 30 6B 41 97 44 C6 0D 1A 34
090  68 13 CC F9 B9 C2 CD 01 C5 EC 8C 99 CF 62 27 19
0A0  C6 D0 88 B1 B4 32 89 49 A9 40 79 41 A4 A6 67 B9
0B0  A3 24 3D 11 A8 F2 34 E9 AF A2 2C 09 A9 9C 48 31
0C0  94 CE 1F 69 83 38 40 59 81 E6 77 71 88 4C C6 A9
0D0  A0 E3 1C 51 CF 2D 1E 89 F9 4A 51 67 FF A9 D1 88
0E0  CF F6 0E EA F6 53 9C E8 D6 B7 BF 87 94 A4 13 33
0F0  3B 43 B3 3D B6 36 5C 33 EB EB EE 5B D6 6F B6 F1
100  BF DB 4A 7F B6 5C 43 EE 59 69 C6 7E D9 72 4D AC
110  A5 1C 11 7C 86 76 21 11 6F FC 4B 47 7C B6 CA 83
120  EB 3C 67 88 7B DB 6F EE F6 3B D9 3F 65 21 F1 F8
130  C0 8B 93 01 77 E5 66 F8 2F 59 6F 44 35 F6 CA 41
140  8B 4F 82 0C 6C 26 CF 79 6C 5A 41 1C C4 09 EC 16
150  9F F9 71 87 E9 92 70 41 E0 64 86 13 D8 94 E2 EC
160  C5 13 DB 31 52 9C 7F 6D 8C 9F B8 72 90 6C 86 CA
170  C4 CC 42 21 DA 1D 3D B6 4D 9D 3B BF 6C 9D 4D 21
180  66 3D 53 EF EF 67 EA 25 27 E7 E7 29 98 F7 D1 69
190  1C C7 CB 9C CD D1 44 90 BD AB 33 1F 1D 35 57 F8
1A0  BB 6B C8 2A A6 61 99 FC 6F F3 89 44 DB CF B7 CE
1B0  FB 78 8E 3C DE 76 6B 7A 7A 1E 65 .A .P .E .T .A
1C0  .G .E .X D0 07 00 00 32 00 00 00 01 00 00 00 00
1D0  00 00 A0 00 00 00 00 00 00 00 00 04 00 00 00 00
1E0  00 00 00 .T .I .T .L .E 00 .S .o .n .g .A .P .E
1F0  .T .A .G .E .X D0 07 00 00 32 00 00 00 01 00 00
200  00 00 00 00 80 00 00 00 00 00 00 00 .T .A .G
210  .S .o .n .g 00 00 00 00 00 00 00 00 00 00 00 00
220  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
230  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
250  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
260  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
270  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
280  00 00 00 00 00 00 00 00 00 00 00 00 FF
```

ID3v2.3

Layer 3 frames

(MP3 data)

APEv2

ID3v1

L3 format "mistakes"

No easy identifications

-> still giving some headaches today

No metadata: not forward-thinking

-> now a stack of hacks and "standards"
with redundant information.

This is not unique to MP3

Many formats pose similar "challenges":

PDF, DOC, NSF, PSD...

Formats also evolve over time:

- "Deprecated" floppy-oriented features:
  Ex: multi-volume archives, incremental PDFs...
- New web-oriented features:
  Ex: linearized PDFs, fast start MP4s,
  top-down ZIP parsing...

More format identification challenges?

Older magics signatures:

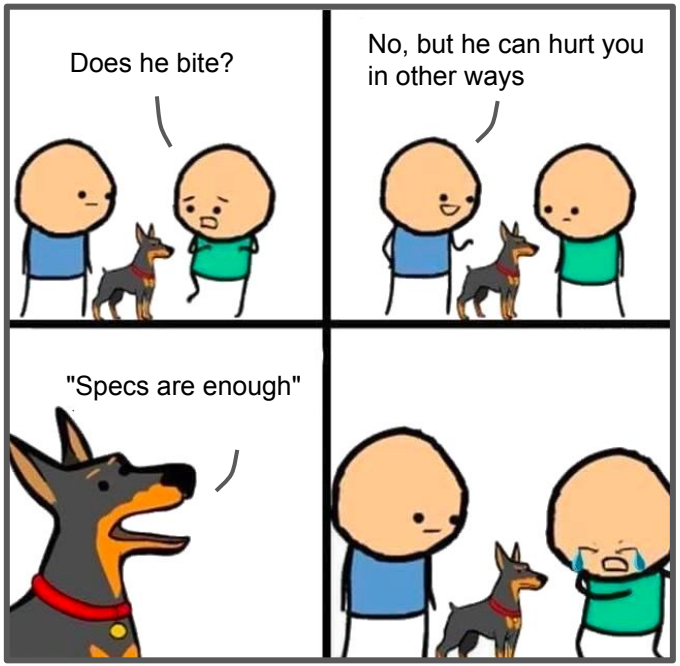
Rar 1.3-1.4: RE~^

Truncated but supported signatures:

%PDF-\0

Too many to mentions!

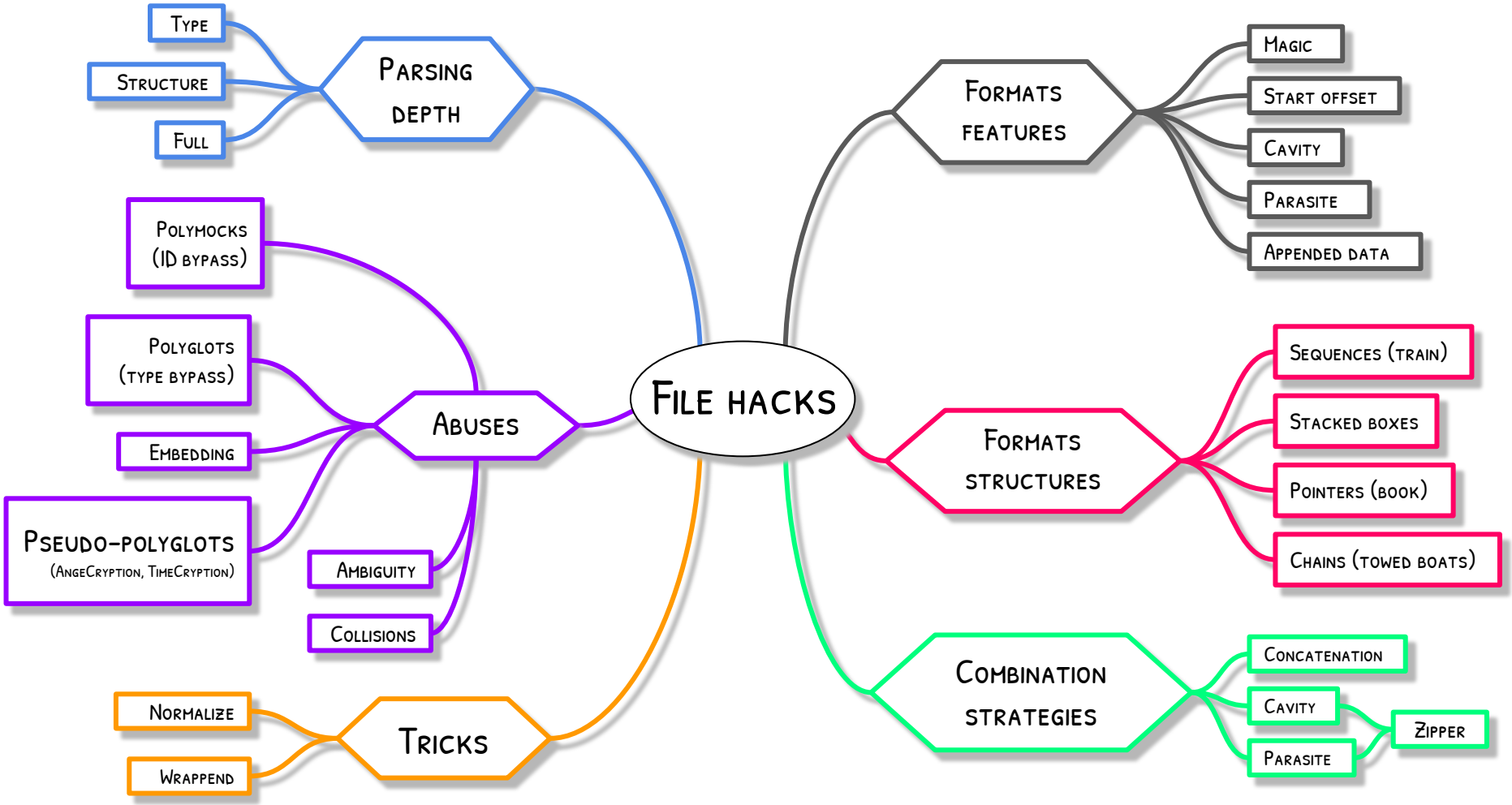
Let's stop the pain here...

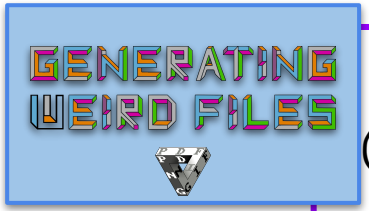


Some more recent news...

File format hacks

A bit of file format messology ;)


File hacks
Parsing depth
Type
Structure
Full
Abuses
Polymocks (ID bypass)
Polyglots (type bypass)
Embedding
Pseudo-polyglots
(AngeCryption, TimeCryption)
Ambiguity
Collisions
Tricks
Normalize
Wrappend
Formats features
Magic
Start offset
Cavity
Parasite
Appended data
Formats structures
Sequences (train)
Stacked boxes
Pointers (book)
Chains (towed boats)
Combination strategies
Concatenation
Cavity
Parasite
Zipper



Polymocks
(ID bypass)

Polyglots
ype bypass)

Embedding

Abuses

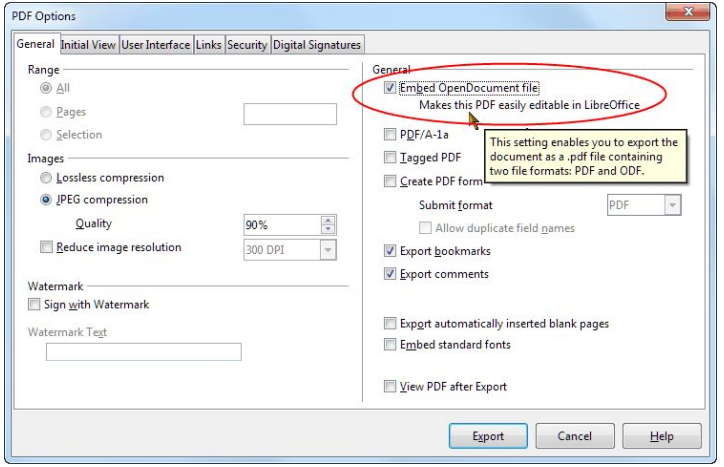
Near polyglots
(AngeCryption, TimeCryption)

Ambiguity

Collisions

My talks on the topics

Polyglots in the wild

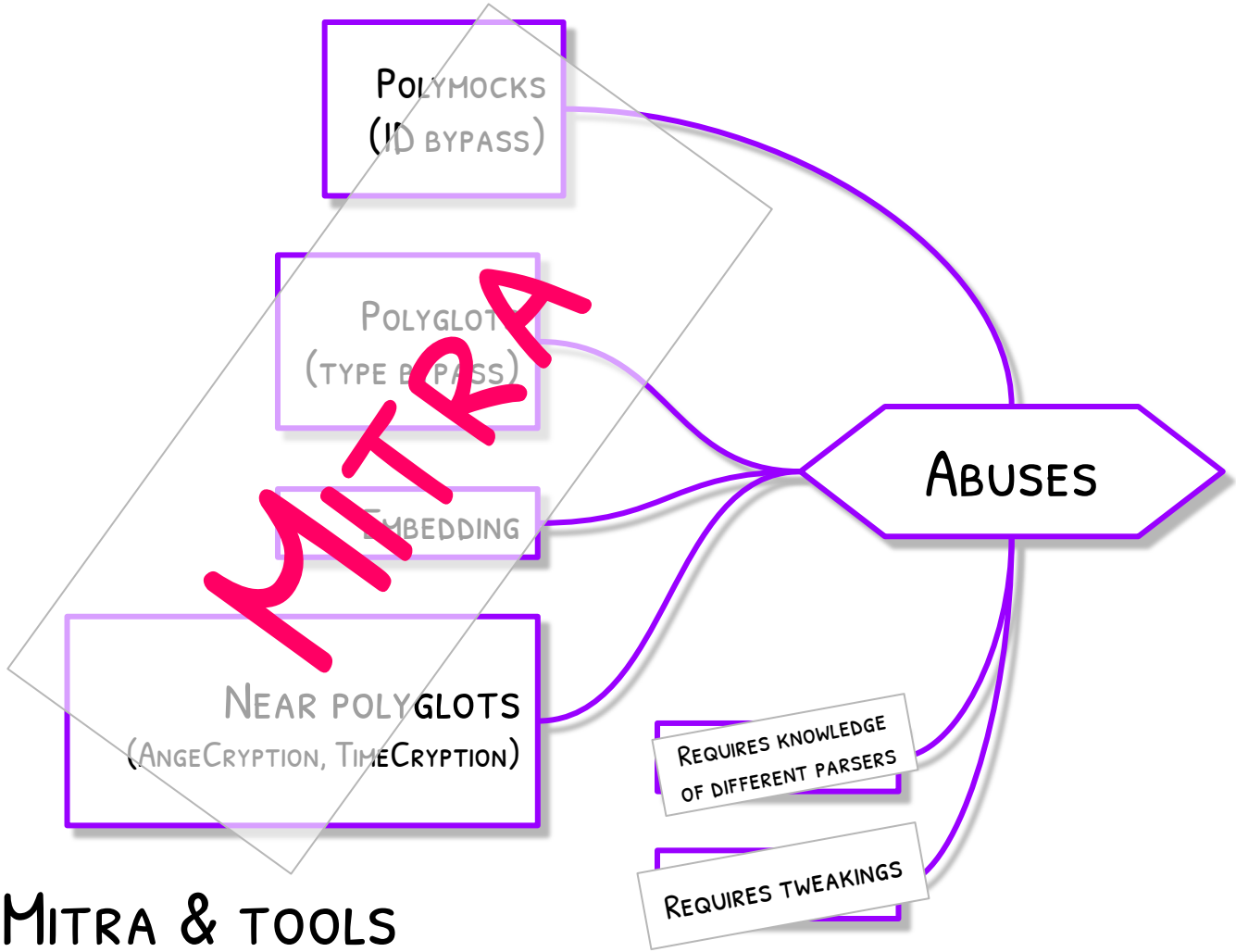


Clean:

- hybrid ISOs : Iso + MBR
- self-extracting archives (executable+archive)
- hybrid PDFs: PDFs with embedded OpenOffice doc.

Malicious:

- Gifar: avatar GIF with appended Java archive.
- CVE-2017-13156 Janus:DEX+APK



Covered by Mitra & tools

Mitra https://github.com/corkami/mitra

Identify file types, make space, combine and adjust data.

Should keep the files valid (UAYOR):

no guarantee, no deep parsing: just a minimal implementation.

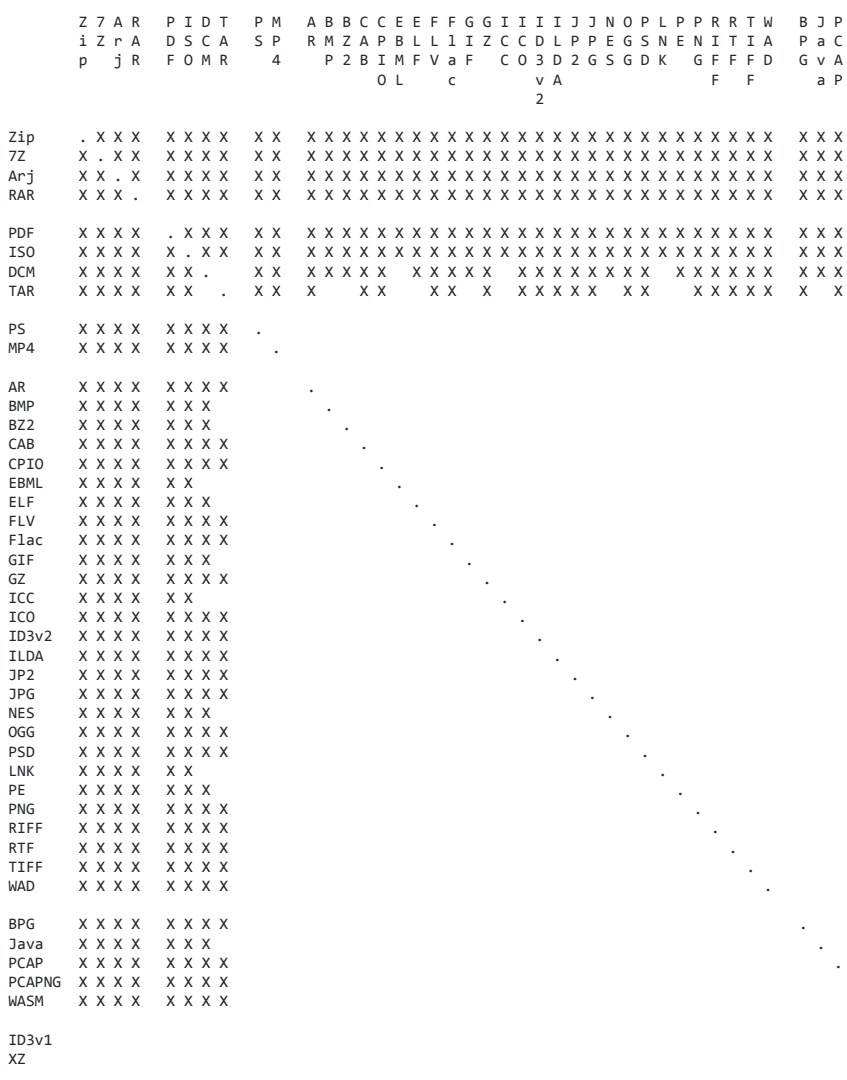
```
$ mitra.py dicom.dcm png.png
dicom.dcm
File 1: DICOM / Digital Imaging and Communications in Medicine
png.png
File 2: PNG / Portable Network Graphics

Zipper Success!
Zipper: interleaving of File1 (type DCM) and File2 (type PNG)
```

Named after Mithridates
(a famous polyglot)

Combinations

Many formats combinations are supported by Mitra.

Easy to extend (no need for full support).



Fooling type identification

Mock files

Mocky: Mitra-based mock signatures patching

A 190-in-1 yet empty file

The file is **mostly empty!**
It only contains magics to fake file types.

https://github.com/corkami/pocs/tree/master/polymocks

00	.M	.Z	60	EA	.j	.P	01	07	19	04	00	10	.S	.N	.D	.H
10	.N	.R	.O	.0	DC	A7	C4	FD	5D	1C	9E	A3	.R	.E	.~	.^
20	.N	.X	.S	.B	18	28	6F	01	.P	.K	03	04	.P	.T	.M	.F
30	.S	.y	.m	.E	.x	.e	.7	.z	BC	AF	27	1C	.S	.O	.N	.G
40	7F	10	DA	BE	00	00	CD	21	.P	.K	01	02	.S	.C	.R	.S
50	.R	.a	.r	.!	^Z	07	01	00	.L	.R	.Z	.I	.P	.L	.O	.T
60	.%	.%	.8	.4	.R	.a	.r	.!	^Z	07	00	00	00	.M	.A	.P
70	.	.(	FD	.7	.z	.X	.Z	00	04	22	4D	18	03	21	4C	18
80	.D	.I	.C	.M	.%	.P	.D	.F	.-	.1	..	.4	.	.o	.b	.j
…	…															

Many magics are at the start of the file.

This file is simultaneously detected as:

- DOS EXE, COM and MBR
- Zoo, ARJ, VirtualBox, MS Compress, 3DS
- ISO, RAW ISO, Nero, PhotoCD
- FastTracker, ScreamTracker, Adlib tracker, Polytracker, SoundFX
- Apple, IBM, HP, Linux, Ultrix, Raid, ODS, Nintendo, Kodak
- EZD, CCP4, Plot84, MAR, Dicom

…

output from
`file --keep-going`

```
multi: Windows Program Information File for \030(o\001
- MAR Area Detector Image,
- Linux kernel x86 boot executable RW-rootFS,
- ReiserFS V3.6
- Files-11 On-Disk Structure (ODS-52); volume label is '            '
- DOS/MBR boot sector
- Game Boy ROM image (Rev.00) [ROM ONLY], ROM: 256Kbit
- Plot84 plotting file
-  DOS/MBR boot sector
- DOSFONT2 encrypted font data
- Kodak Photo CD image pack file , landscape mode
- SymbOS executable v., name: HNRO0\334\247\304\375]\034\236\243
- ISO 9660 CD-ROM filesystem data (raw 2352 byte sectors)
- Nero CD image at 0x4B000 ISO 9660 CD-ROM filesystem data
- High Sierra CD-ROM filesystem data
- Old EZD Electron Density Map
- Apple File System (APFS), blocksize 24061976
- Zoo archive data, modify: v78.88+
- Symbian installation file
- 4-channel Fasttracker module sound data Title: "MZ`\352\210\360'\315!"
- Scream Tracker Sample  adlib drum mono 8bit unpacked
- Poly Tracker PTM Module Title: "MZ`\352\210\360'\315!"
- SNDH Atari ST music
- SoundFX Module sound file
- D64 Image
- Nintendo Wii disc image: "NXSB\030(o\0
- Nintendo 3DS File Archive (CFA)
- Unix Fast File system [v1] (litt
- Unix Fast File system [v2] (litt
- Unix Fast File system [v2] (littl
- ISO 9660 CD-ROM filesystem data (
- F2FS filesystem, UUID=00000000-0000-0000-0000-000000000000, volume name ""
- DICOM medical imaging data
- Linux kernel ARM boot executable zImage (little-endian)
- CCP4 Electron Density Map
- Ultrix core file from 'X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVI...
- VirtualBox Disk Image (MZ`\352\210\360'\315!), 5715999566798081280 bytes
- MS Compress archive data
- AMUSIC Adlib Tracker MS-DOS executable, MZ for MS-DOS COM executable for DOS
- JPEG 2000 image
- ARJ archive data
- unicos (cray) executable
- IBM OS/400 save file data
- data
```

```
0       0x0     Gameboy ROM,, [ROM ONLY], ROM: 256Kbit
80      0x50    RAR archive data, version 5.x
88      0x58    lrzip compressed data
89      0x59    rzip compressed data - versi
114     0x72    xz compressed data
120     0x78    LZ4 compressed data
...
```

output (150 sigs) from `Binwalk`

```
$ mocky.py --combined input/jpg.jpg
Filetype: JFIF / JPEG File Interchange Format
Parasite-combined sig(s): unicos / Symbian / snd / wdk / SoundFont / icc / VICAR / netbsd_ktraceS / SoundFX / VirtualBox /
ScreamTracker / Plot84 / ezd / dicom / Tar(checksum) / ds / CCP4 / DRDOS / pif / mbr
25676
> Combined Mock: mA-jpg.jpg
```

Add any possible signature with Mocky

```
$ file mA-jpg.jpg
mA-jpg.jpg: tar archive
```

<- FILE sees it as a TAR file!
(valid TAR signature + checksum)

Many detected file types

```
$ file mA-jpg.jpg --keep-going --raw
mA-jpg.jpg: tar archive
- DR-DOS executable (COM)
- JPEG image data, baseline, precision 8, 104x56, components 1
- Windows Program Information File for acsp`
- VICAR label file
- DOS/MBR boot sector
- Nintendo DS ROM image: "������" (SNDH, Rev.107) (homebrew)
- Plot84 plotting file
-  DOS/MBR boot sector
- sfArk compressed Soundfont
- Old EZD Electron Density Map
- Symbian installation file
- Scream Tracker Sample mono 8bit
- SNDH Atari ST music
- SoundFX Module sound file
- DICOM medical imaging data
- CCP4 Electron Density Map
- VirtualBox Disk Image (������), 5715999566798081280 bytes
- unicos (cray) executable
- data
```

Still a perfectly valid JPEG!
(with an extra COMment segment stuffed with signatures)

```
$ identify -verbose ./mA-jpg.jpg
Image:
  Filename: ./mA-jpg.jpg
  Format: JPEG (Joint Photographic Experts Group JFIF format)
  Mime type: image/jpeg
  Class: PseudoClass
  Geometry: 104x56+0+0
  Resolution: 36x36
  Print size: 2.88889x1.55556
  Units: PixelsPerCentimeter
  Colorspace: Gray
[...]
```

Easy polymock crafting with Mocky

Near-polyglots

Def: polyglots with some contents
that is replaced by an external operation.

(the smaller the better)

Ex: Crypto-polyglots

```
    B  M 3C 00 00 00 00 00 00 00 20 00 00 00 0C 00
00: 89  P  N  G \r \n ^Z \r 00 00 00 2C  c  O  M  M
10: 00 00 0D 00 07 00 01 00 01 00 FF FF FF 00 00 00
20: 00 00 00 00 65 40 00 00 55 40 00 00 67 60 00 00
30: 57 50 00 00 65 60 00 00 00 00 00 00 00 00 00 00
40: 1D 44 05 DC 00 00 00 0D  I  H  D  R 00 00 00 0D
50: 00 00 00 07 01 03 00 00 00 E9 BE 55 59 00 00 00
60: 06  P  L  T  E FF FF FF 00 00 00 55 C2 D3 7E 00
70: 00 00 1B  I  D  A  T 08 1D 63 00 82 54 03 86 70
80: 07 86 F4 02 06 F7 00 06 57 03 06 06 06 00 21 1A
90: 03 10 32 6A 0B 48 00 00 00 00  I  E  N  D AE 42
A0: 60 82
```

```
B  M 3C 00 00 00 00 00 00 00 20 00 00 00 0C 00
```


```
89  P  N  G \r \n ^Z \n 00 00 00 2C  c  O  M  M
```


```
mitra.py bmp.bmp png.png --overlap
```

Generates `O(10-40)-PNG[BMP]{424D3C0000000000000020000000C00}.1965e270.png.bmp`

A BMP/PNG NEAR POLYGLOT, WITH 16 BYTES OF OVERLAP

A valid BMP is AES-CBC encrypted as a PNG with a special IV to encrypt the first block as expected (AngeCryption).

AngeCryption works with
ECB, CBC, CFB, OFB

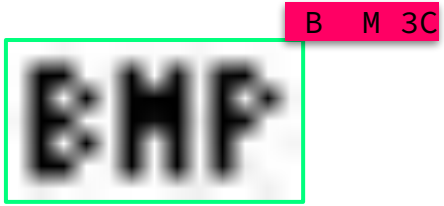
```
00:  B  M 3C 00 00 00 00 00 00 00 20 00 00 00 0C 00
10: 00 00 0D 00 07 00 01 00 01 00 FF FF FF 00 00 00
20: 00 00 00 00 65 40 00 00 55 40 00 00 67 60 00 00
30: 57 50 00 00 65 60 00 00 00 00 00 00 00 00 00 00
40: 00 A1 3B E2 E0 64 F0 A7 AE 5E 21 64 BC 44 5F 09
50: E3 67 D3 10 19 AF 09 F1 99 1A 33 B3 BF 28 EF 9E
60: 71 3D 87 79 EC 73 A9 60 82 74 1B EB 08 B4 4E B7
70: E5 9E 16 A9 CE BC 1B 71 99 E7 F8 E8 FA 8C C0 6C
80: 6B 85 4B 56 73 7D 22 BD 46 DE AC 3F BF EE 8B 96
90: AB 74 55 5F 21 B7 10 1B D6 96 18 45 6E E5 B0 3C
A0: 7C 22 99 87 EA FE 1F 4D FF C8 52 C0 24 C7 AD A8
```

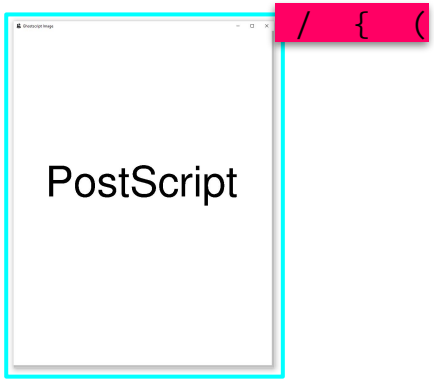
AES-CBC →

```
89  P  N  G \r \n ^Z \n 00 00 00 30  c  O  M  M
71 2F D8 C7 79 C1 EB CF 63 B0 22 2B 0A 6D E3 2D
24 49 57 B1 9B BB C2 FA 94 8A 8C 53 9E A1 30 63
30 C9 41 75 EA AF 75 EE 95 7C 57 E9 16 4F F7 3B
1D 44 05 DC 00 00 00 0D  I  H  D  R 00 00 00 0D
00 00 00 07 01 03 00 00 00 E9 BE 55 59 00 00 00
06  P  L  T  E FF FF FF 00 00 00 55 C2 D3 7E 00
00 00 1B  I  D  A  T 08 1D 63 00 82 54 03 86 70
07 86 F4 02 06 F7 00 06 57 03 06 06 06 00 21 1A
03 10 32 6A 0B 48 00 00 00 00  I  E  N  D AE 42
60 82 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```


```
mitra/utils/cbc$ angecrypt.py "O(10-40)-PNG[BMP]{424D3C000000000000000200000000C00}.1965e270.png.bmp" bmp-png.cbc
```

```
     B  M 3C
00:  /  {  ( 00 00 00 00 00 00 00 20 00 00 00 0C 00
10:  00 00  0D 00 07 00 01 00 01 00 FF FF FF 00 00 00
20:  00 00 00 00 65 40 00 00 55 40 00 00 67 60 00 00
30:  57 50 00 00 65 60 00 00 00 00 00 00  )  }  %  !
40:   P  S \r \n  /  N  i  m  b  u  s  S  a  n  s  -
50:   R  e  g  u  l  a  r     1  0  0     s  e  l  e
60:   c  t  f  o  n  t \r \n  7  5     4  0  0     m
70:   o  v  e  t  o \r \n  (  P  o  s  t  S  c  r  i
80:   p  t  )     s  h  o  w \r \n  s  h  o  w  p  a
90:   g  e \r \n  s  t  o  p \r \n 00 00 00 00 00 00
```






```
mitra.py postscript.ps bmp.bmp --overlap
```

Generates O(3-3c)-PS[BMP]{424D3C}.209881aa.ps.bmp

A BMP/PS NEAR POLYGLOT WITH 3 BYTES OF OVERLAP

Both files are decrypted via GCM from the same ciphertext but via different keys.
The nonce is bruteforced to generate the right overlap with either key.

TimeCryption works with
CTR, OFB, GCM, GCM-SIV, OCB3

ciphertext

Key$_1$ Key$_2$

00:	B	M	3C	00	00	00	00	00	00	00	20	00	00	00	0C	00
10:	00	00	0D	00	07	00	01	00	01	00	FF	FF	FF	00	00	00
20:	00	00	00	00	65	40	00	00	55	40	00	00	67	60	00	00
30:	57	50	00	00	65	60	00	00	00	00	00	00	B7	EB	32	E8
40:	16	D6	9E	76	AC	20	9C	8C	9F	06	6F	55	3F	96	0E	09
50:	04	24	41	5D	22	7C	A6	E5	0E	AC	ED	1C	04	65	BE	E6
60:	E8	AB	E4	D2	C6	B6	CD	9F	AB	85	E1	CE	03	C5	A5	85
70:	70	B5	09	EB	EB	CB	D1	2F	7C	4D	B0	09	35	38	D9	B7
80:	82	31	BB	87	96	22	C8	4E	C0	EC	89	C3	CB	97	63	D3
90:	A0	28	47	5B	71	C2	95	EC	12	E2	52	B0	6F	B1	EE	61

/	{	(	07	3A	14	40	E5	3E	EC	AE	A2	AD	87	AA	38
11	C4	5D	5A	35	2D	EB	EC	47	CC	A7	B5	63	22	90	B7
5F	D7	41	7B	FD	6D	53	DB	78	9F	AA	A6	2B	22	61	AD
BB	38	48	4A	5C	A7	D5	E4	63	4F	4D	7B	)	}	%	!
P	S	\r	\n	/	N	i	m	b	u	s	S	a	n	s	-
R	e	g	u	l	a	r		1	0	0		s	e	l	e
c	t	f	o	n	t	\r	\n	7	5		4	0	0		m
o	v	e	t	o	\r	\n	(	P	o	s	t	S	c	r	i
p	t	)		s	h	o	w	\r	\n	s	h	o	w	p	a
g	e	\r	\n	s	t	o	p	\r	\n	00	00	00	00	00	00


```
mitra/utils/gcm$ meringue.py "O(3-3c)-PS[BMP]{424D3C}.209881aa.ps.bmp" bmp-ps.gcm
```

Risk: unexpected decryption

The same encrypted content can also be decrypted with authentication with another key.

Store CleanFile encrypted via GoodKey.
When BadKey is added to the KeyRing,
CleanCipher gets decrypted as BadFile with authentication.

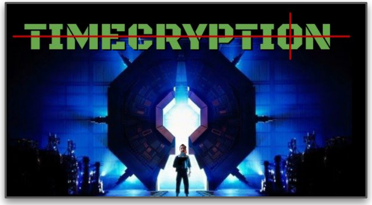
Near polyglots

AngeCryption: ECB CBC CFB **OFB**

TimeCryption: CTR **OFB** GCM OCB$_3$ GCM-SIV

A bit complex, but powerful when mixed with cryptography.

May require some bruteforcing.




```
                                                              Variable    Unsupported
                                                               offset      parasite
Minimal start offset
         1 2 4 8     9  16  20  23  28  34  40  64  94  132    12  28
                      12          26  32  36      68  112 226    16

         P P J F M T F W G P R I R B C I P C J P E A P I I J    W B O      B E G L N
         S E P l P I L A Z N I D T M P L S A P C L R C C C a    A P G      Z B I N E
             G a 4 F V D   G F 3 F P I D D B 2 A F   A O C v    S G G      2 M F K S
               c   F         F v     O A       P     P     a    M            L
                               2               N
                                               G

1* PS    . M A ? ? ? ? ? ? A ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?    ? ? ?      ? ? ? ? ?
2^ PE    M . A A A A A A A A A A A A A A A A A A ! ! ! ! ! !    M M M      ! ! ! ! !
4+ JPG   A A . A A A A A A A A A A A A A A A A A A A A A A A    A A A      A A A A A
.  .
.  .
```

2022: MD5 is a form of art

Hash collisions

"Classic" crypto attacks,
enhanced by file format tricks.

https://github.com/corkami/collisions/

```
$ ./gz.py libjpeg-turbo-2.1.3.tar.gz tiff-4.4.0rc1.tar.gz
libjpeg-turbo-2.1.3.tar.gz (2260756 bytes): split in 78 members
tiff-4.4.0rc1.tar.gz (2841082 bytes): split in 78 members
Success!
22fb3b1171cc1bb9969b093e77f69e7c
coll-1.gz => libjpeg-turbo-2.1.3.tar.gz
coll-2.gz => tiff-4.4.0rc1.tar.gz
```

In less than 1s...

```
$ tar tvf coll-1.gz
drwxrwxr-x root/root         0 2022-02-25 19:53 libjpeg-turbo-2.1.3/
-rw-rw-r-- root/root     24927 2022-02-25 19:53 libjpeg-turbo-2.1.3/BUILDING.md
[...]
-rw-rw-r-- root/root     10840 2022-02-25 19:53 libjpeg-turbo-2.1.3/wrppm.c
-rw-rw-r-- root/root      7483 2022-02-25 19:53 libjpeg-turbo-2.1.3/wrtarga.c
```

```
$ tar tvf coll-2.gz
drwxrwxr-x even/even         0 2022-05-20 18:13 tiff-4.4.0/
-rw-rw-r-- even/even      1146 2021-03-05 14:01 tiff-4.4.0/COPYRIGHT
[...]
-rw-rw-r-- even/even      1520 2022-02-19 16:33 tiff-4.4.0/contrib/addtiffo/Makefile.am
-rw-rw-r-- even/even     20907 2022-05-20 18:11 tiff-4.4.0/contrib/addtiffo/Makefile.in
-rw-rw-r-- even/even     33511 2022-05-20 18:11 tiff-4.4.0/Makefile.in
```

Instant MD5 collisions of:

JPG, PNG, GIF, GZIP, PE, MP4, JPEG2000, PDF, DOCX/PPTX/XSLX, EPUB, 3MF, XPS...

Not possible for: ELF, Mach-O, Java Class, TAR, ZIP...

Computing MD5 collisions...

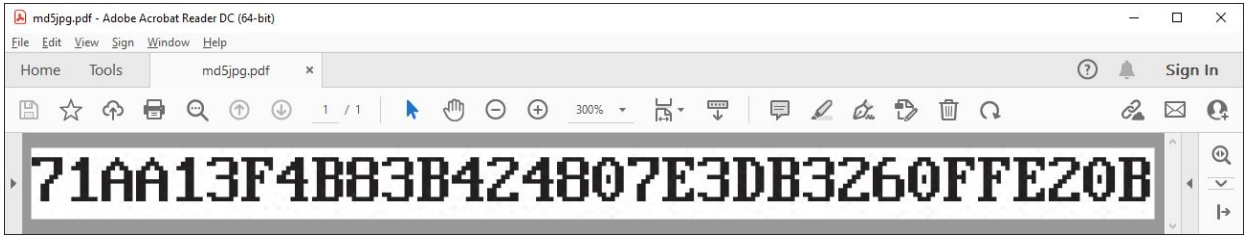
Mako's "Toy MD5 Collider" for the Mega Drive

dd49d7eb...

1988: Sega Megadrive
16bits @ 7.6 MHz

1992: MD5

...on a MegaDrive



A PDF hashquine

Hashquines

Files showing their own MD5 (PDF, PNG, GIF, PS, TIFF)

A GIF hashquine

https://github.com/corkami/collisions/blob/master/hashquines/README.md

Monomorph: any payload, same hash

32768 Md5 collisions to encode any 4 kb payload.

```
$ hello
Hello World!
$ hashquine
My MD5 is: 3cebbe60d91ce760409bbe513593e401
$ md5sum *
3cebbe60d91ce760409bbe513593e401  bind_tcp
3cebbe60d91ce760409bbe513593e401  hashquine
3cebbe60d91ce760409bbe513593e401  hello
3cebbe60d91ce760409bbe513593e401  rickroll
```

https://github.com/DavidBuchanan314/monomorph

Worried about hash collisions?

DetectColl can detect any MD5 or SHA1 hash collision.

Flame's unique collision.

```
$ detectcoll flame.der
Found collision in block 11:
   dm: dm4=80000000 dm11=ffff8000 dm14=80
   ihv1=1ba33aac3a7f9ed70aec349b40390e85
     ihv2=9ba33aac3c7f60ee8cebf69bc2391085
```

Newest SHA1 ones: Shambles

```
$ detectcoll 13-shambles1.bin
  Found collision in block 9 using Dv
     dm: dm0=f4000002 dm1=3ffffff0 dm2=6c0000
  dm7=abffffec dm8=f4000002 dm9=c0000010 dm10=93fff
  dm15=a8000010
     ihv1=72d42d69a661589d73fc20173d1dce014c7813bc
     ihv2=72d43f9ba661592f73fc20173d1dce03cc7813bc
```

Structure heuristics can also help to pre-filter files.

https://github.com/corkami/collisions/blob/master/README.md#detection

You've been warned...

Use MD5 at your own risks!

It's trivial and instant
to craft colliding files
with arbitrary contents.

And it's a fun toy.

What about SHA1?

All these formats attacks are already possible with SHA1!

MD5 and SHA1/2 enforce similar file constraints.

SHA1 computations are already documented and implemented, but still **too expensive to run** ($11k-45k per format).

No such computations for SHA2 yet.

Conclusion

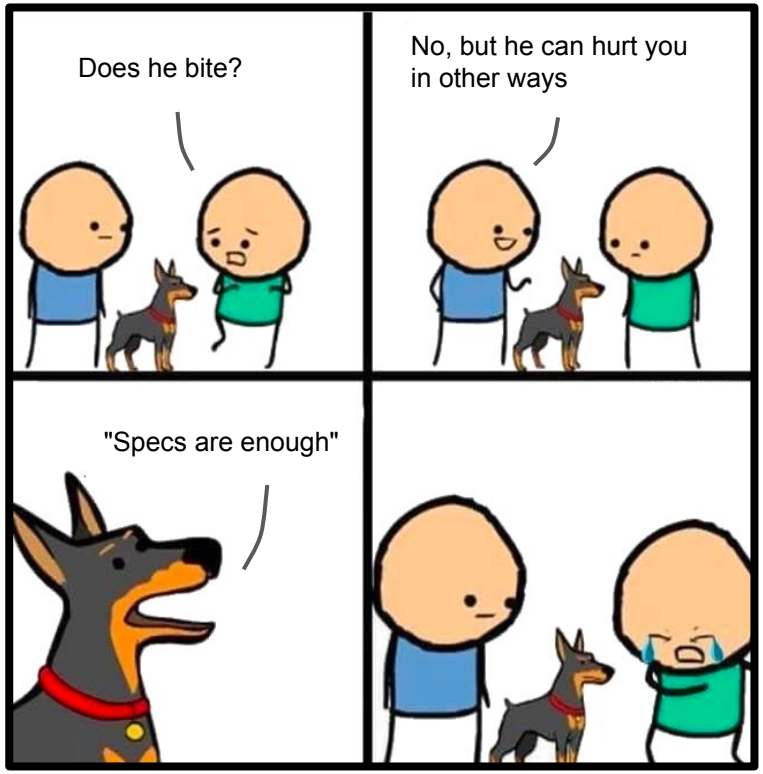
It was just an overview...

File formats pose many challenges

Many formats are a big mess of "standards" together:
A growing technical debt.

New hacks appear for various reasons:
the landscape becomes even more complex.

Hash abuses become more risky.
-> time-consuming to detect or to upgrade to SHA2/3.



Thank you!

Questions / feedback ?

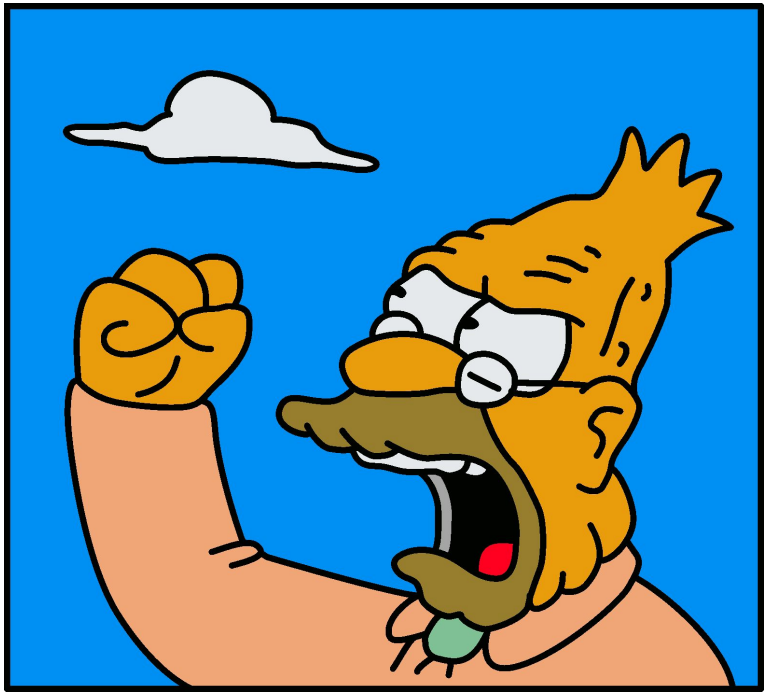
ANGE ALBERTINI
reverse engineering
VISUAL DOCUMENTATIONS
@angealbertini
ange@corkami.com
http://www.corkami.com

Bonus

OldManYellsAt.*

My own redrawing, available as:
- PDF
- indexed PNG
- optimized SVG

Feel free to convert
to your "favorite" file format!



https://github.com/corkami/pics/blob/master/tracing/README.md