Bulk Data Analysis With Optimistic Decompression and Sector Hashing

AAFS Digital & Multimedia Sciences Section
Thursday, February 21, 2013 / 3:45 p.m. - 4:05 p.m.

Simson L. Garfinkel, Kristina Foster, Joel Young

Naval Postgraduate School

Kevin Fairbanks, Johns Hopkins Applied Physics Lab

http://simson.net/

Bulk Data Analysis With Optimistic Decompression and Sector Hashing

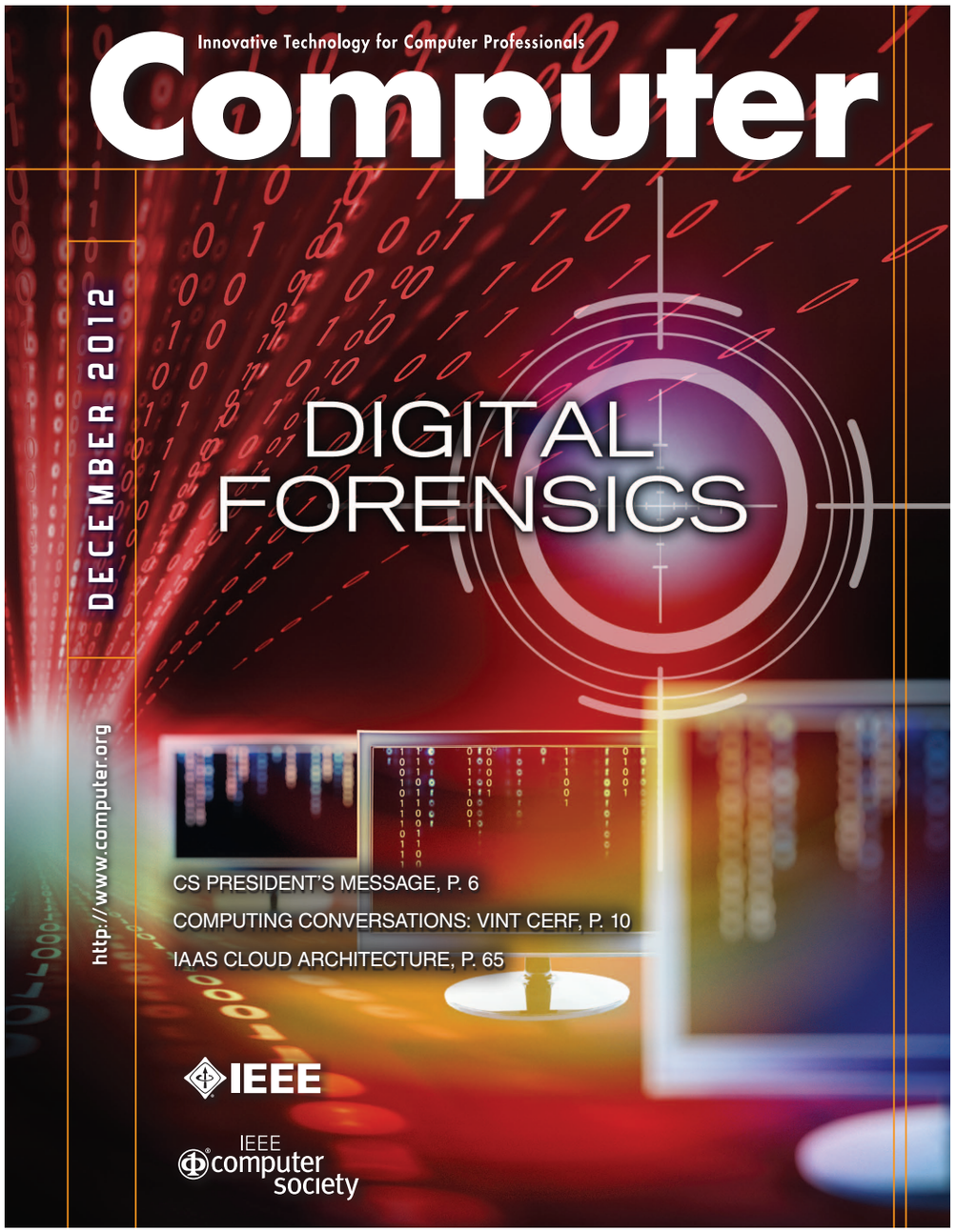
AAFS Digital & Multimedia Sciences Section
Thursday, February 21, 2013 / 3:45 p.m. - 4:05 p.m.

20 Min with Q&A!

Simson L. Garfinkel
Associate Professor, Naval Postgraduate School

http://simson.net/

For information on Distinct Sector Hashing, please see...



COVER FEATURE

Distinct Sector Hashes for Target File Detection

Joel Young, Kristina Foster, and Simson Garfinkel, *Naval Postgraduate School*
Kevin Fairbanks, *Johns Hopkins University*

Using an alternative approach to traditional file hashing, digital forensic investigators can hash individually sampled subject drives on sector boundaries and then check these hashes against a prebuilt database, making it possible to process raw media without reference to the underlying file system.

Forensic examiners frequently search disk drives, cell phones, and even network flows to determine if specific known content is present. For example, a corporate security officer might examine a suspicious employee's laptop for unauthorized documents; law enforcement officers might search a suspect's home computer for illegal pornography; and network analysts might reconstruct Transmission Control Protocol streams to determine if malware was downloaded. In these and many other cases, examiners typically identify files by computing their cryptographic hash—often with MD5 or SHA1 hash algorithms—and then searching a database for the resulting hash value.

Use of hash values for file identification is pervasive in digital forensics—every popular forensics package has built-in support. One of the most widely used databases is the National Software Reference Library (NSRL) Reference Data Set (RDS). Version 2.36, released in March 2012, contains 25,892,924 distinct file hashes (www.nsrl.nist.gov). Other databases are available to customers of specific companies and to law enforcement organizations.

There are many limitations when using file hashes to identify known content. Because changing just a single bit of a file changes its hash, pornographers, malware authors, and other miscreants can evade detection simply by changing a comma to a period or appending a few random bytes to a file. Likewise, hash-based identification will not work if sections of the file are damaged or otherwise unrecoverable. This is especially a problem when large video files are deleted and the operating system reuses a few sectors for other purposes: most of the video is still present on the drive, but recovered video segments will not appear in a database of file hashes.

SECTOR HASHING

We are developing alternative systems for detecting target files in large disk images using cryptographic hashes on sectors of data rather than entire files. Modern file systems align the start of most files with the beginning of a disk sector. Thus, when a megabyte-sized video is stored on a modern hard drive, the first 4 kibibytes are stored in one disk sector, the second 4 KiBytes are stored in another disk sector, typically the adjacent one, and so on. (In our work, we distinguish between power-of-two-based sizes of digital artifacts, such as kibibytes, and power-of-ten-based sizes, such as kilobytes. See the "Decimal versus Binary Prefixes" sidebar for more details.) Furthermore, by sampling randomly chosen sectors from the drive, it is only necessary to read a tiny fraction of the drive to determine with high probability if a target file is present. This enables rapid triage of drive images.

We compare drive sector hashes to a hash database of fixed-sized file fragments, which we call *blocks*. The terms "sector" and "block" are often used incorrectly as syn-

Published by the IEEE Computer Society

Disclaimer

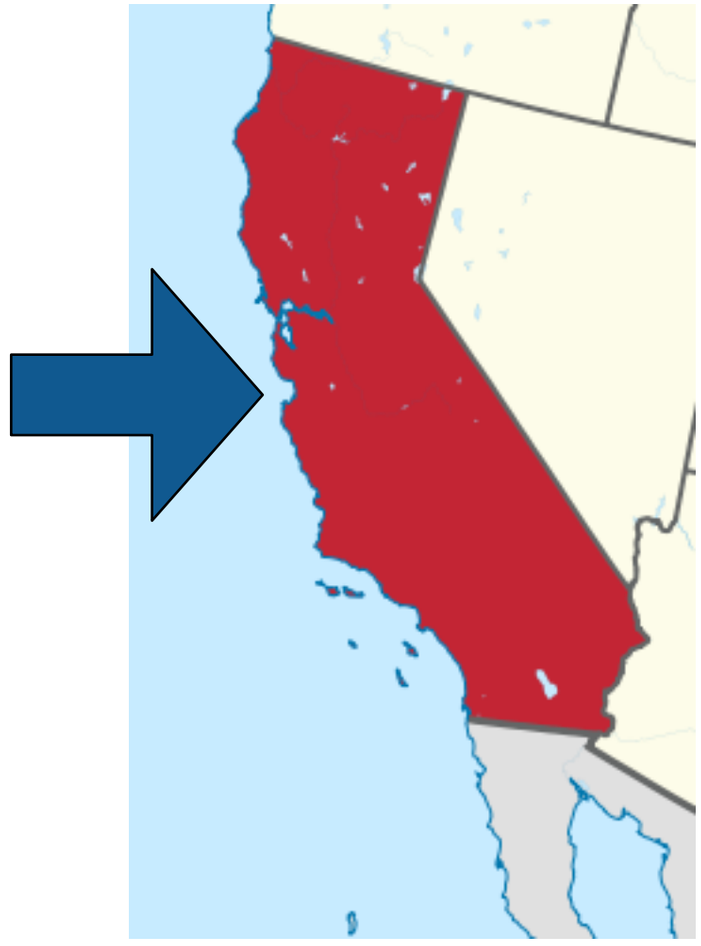
The opinions expressed herein are those of the author(s), and are not necessarily representative of those of the Naval Postgraduate School, the Department of Defense (DOD); or, the United States Army, Navy, or Air Force.

The author has no financial interest in any of the products or technologies described in this presentation.

NPS is the Navy's Research University.

Monterey, CA — 1500 students

- US Military
- Civilian (Scholarship for Service & SMART)
- Foreign Military (30 countries)



National Capital Region (NCR) Office

- 900 N Glebe (Ballston)/Virginia Tech building
  ARLINGTON, VA

My research focus: better tools and algorithms for triage.

Identification of high-value data.

- What is important?
  - *Contacts, calendar, documents?*
  - *Software?*
  - *Geolocation information?*
  - *Temporal / time sequence?*

Correlation — are there copies of the *same* or *similar* information?

- Identify previously unknown *organizations* or *networks*
- Identify data that is *unusual* or *emerging*

Presentation and Integration:

- Make the results *understandable*.
- Effect organizational change through adoption & integration

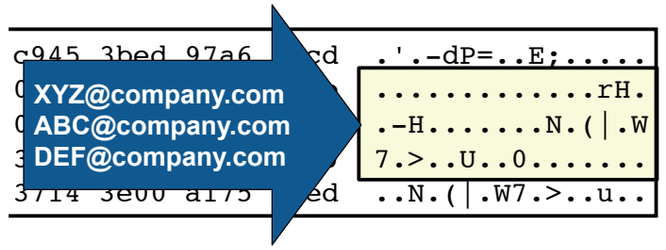
Today's tools frequently miss case-critical data.

Email addresses are typical "features" of forensic interest.

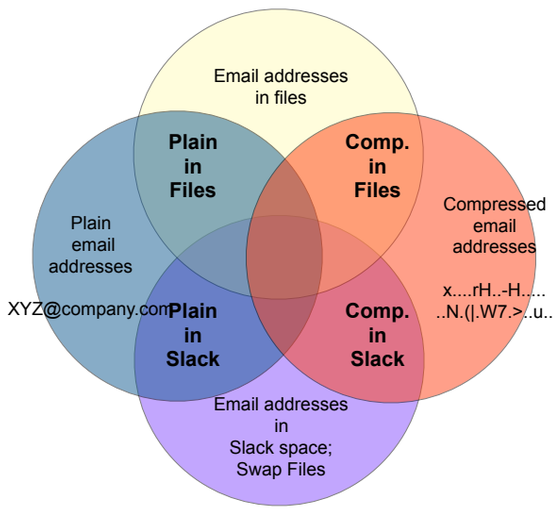
ABC@company.com

Email addresses can be compressed.

Popular forensic tools do not optimistically decompress.



Our study of 1400 drives found thousands of email addresses that were *only in compressed data.*



Email addresses are powerful digital forensic identifiers

Email addresses can reveal:

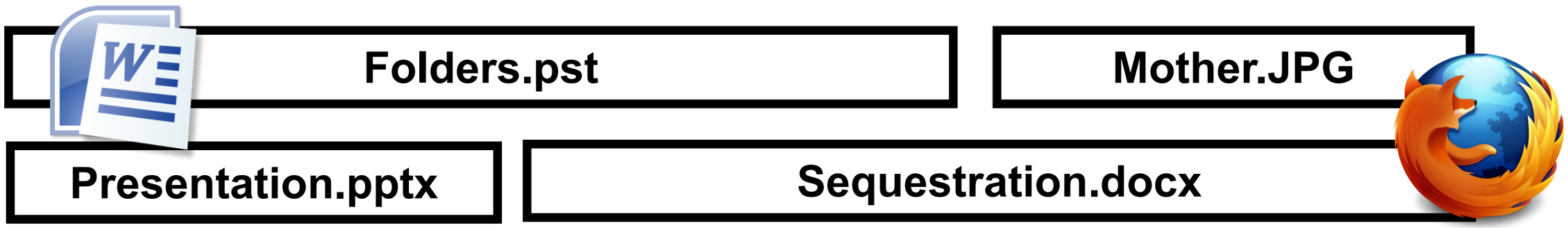
- User(s) of a device
- Associates
- Connections between devices

ABC@company.com
DEF@company.com
XYZ@company.com

HIJ@network.net
KLM@network.net
NOP@network.net
XYZ@company.com

- Today's forensic tools implement two strategies for extracting email addresses.
  1. *Text extraction from files*
  2. *Text extraction from bulk data*

1. Text extraction from files:
File ➳ Text ➳ RegEx ➳ Email Addresses



➳ TEXT TEXT TEXT **XYZ@COMPANY.COM** TEXT TEXT TEXT TEXT ➳ **XYZ@COMPANY.COM**

2. Text extraction from bulk data: [bulk data] ➳ RegEx ➳ Email Addresses

Folders.pst

Mother.JPG

Presentation.pptx

Sequestration.docx

```
a097 83a1 ed96 26a6 3c69 3d0f 750a 2399   ......&.<i=.u.#.
a2b5 bea7 692f 5847 a38a dd53 082c add5   ....i/XG...S.,..
5061 b64c 721d 864b 90b6 b55f bb04 735c   Pa.Lr..K..._..s\
9448 6730 5453 df64 813e b603 5795 2242   .Hg0TS.d.>..W."B
e9c8 7454 7322 7cdc b60e 97af 2f64 2728   ..tTs"|...../d'(
3cfb 84bd 2a84 2dfe 50ea 5935 c349 1513   <XYZ@COMPANY.COM
a9e9 e92c a3f8 6e46 0530 8a88 c7a2 5d2b   ...,..nF.0....]+
d89d 77cc fe1e f637 f3f3 d0af 1b47 c09b   ..w....7.....G..
```

It’s easy to see email addresses in bulk data.

Folders.pst

Mother.JPG

Presentation.pptx

Sequestration.docx

```
a097 83a1 ed96 26a6 3c69 3d0f 750a 2399   ......&.<i=.u.#.
a2b5 bea7 692f 5847 a38a dd53 082c add5   ....i/XG...S.,..
5061 b64c 721d 864b 90b6 b55f bb04 135c   Pa.Lr..K..._..s\
9448 6730 5453 df64 813e b603 5795 1142   .Hg0TS.d.>..W."B
e9c8 7454 7322 7cdc b                     ..tTs"|...../d'(
3cfb 84bd 2a84 2dfe 5                     <XYZ@COMPANY.COM
a9e9 e92c a3f8 6e46 0                     ...,..nF.0....]+
d89d 77cc fe1e f637 f313 d0af 1b47 c90b   ..w....7.....G..
```

XYZ@company.com

Every email address is a sequence of bytes.

A simple email address:

XYZ@company.com

Stored on disk / in memory as 15 bytes:

X Y Z @ c o m p a n y . c o m

Each byte is 8-bits. Range is 0-255

88 89 90 64 99 111 109 112 97 110 121 46 99 111 109

Normally bytes are displayed in hexadecimal notation:

58 59 5a 40 63 6f 6d 70 61 6e 79 2e 63 6f 6d

This is UNICODE

Every email address is a sequence of bytes.

A simple email address:

XYZ@company.com

Stored on disk / in memory as 15 bytes:

X	Y	Z	@	c	o	m	p	a	n	y	.	c	o	m

Each byte is 8-bits. Range is 0-255

88	89	90	64	99	111	109	112	97	110	121	46	99	111	109

Normally bytes are displayed in hexadecimal notation:

58	59	5a	40	63	6f	6d	70	61	6e	79	2e	63	6f	6d

This is UNICODE

Byte sequences can be encoded in many ways.

XYZ@company.com

- Unicode: “XYZ@company.com”

```
58 59 5a 40 63 6f 6d 70 61 6e 79 2e 63 6f 6d
```

- Base 16: “58595a40636f6d70616e792e636f6d0a”

```
3538 3539 3561 3430 3633 3666 3664 3730  58595a40636f6d70
3631 3665 3739 3265 3633 3666 3664 3061  616e792e636f6d0a
```

- Base 64: “WFlaQGNvbXBhbnkuY29tCg===”

```
5746 6c61 5147 4e76 6258 4268 626e 6b75  WFlaQGNvbXBhbnku
5932 3974 4367 3d3d 3d0a                 Y29tCg===.
```

- Compression: echo “XYZ@company.com” | compress | xxd

```
1f9d 9058 b268 0132 e64d 1b38 61dc e471  ...X.h.2.M.8a..q
51b0 8d02                                Q...
```

Compression works by eliminating repeated sequences:

Computers use compression to save memory:

```
5859 5a40 636f 6d70 616e 792e 636f 6d20  XYZ@company.com
4142 4340 636f 6d70 616e 792e 636f 6d20  ABC@company.com
4445 4640 636f 6d70 616e 792e 636f 6d20  DEF@company.com
```

Compressed with “gzip:”

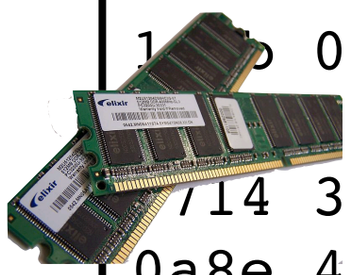
```
1f8b 0800 0000 0000 0203 8b88 8c72 48ce  .............rH.
cf2d 48cc abd4 03d2 0a8e 4ece 287c 1757  .-H.......N.(|.W
3714 3e00 b455 c1c5 3000 0000            7.>..U..0...
```

Compressed email addresses do not “look” like email addresses!

— *Forensic tools must decompress FIRST to identify compressed email addresses.*

It's hard to see compressed email address in bulk data.

```
e327 962d 6450 3d91 c945 3bed 97a6 a4cd   .'.-dP=..E;.....
1[illegible]b 0800 0000 0000 0203 8b88 8c72 48ce   .............rH.
[illegible]8cc abd4 03d2 0a8e 4ece 287c 1757   .-H.......N.(|.W
7714 3e00 b455 c1c5 3000 0000 0000 0000   7.>..U..0.......
0a8e 4ece 287c 1757 3714 3e00 a175 10ed   ..N.(|.W7.>..u..
```



Folders.pst

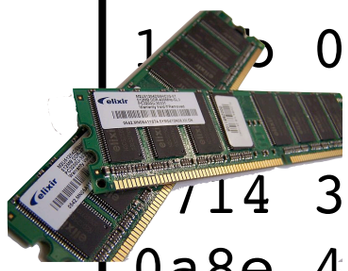
Mother.JPG

Presentation.pptx

Sequestration.docx

```
a097 83a1 ed96 26a6 3c69 3d0f 750a 2399   ......&.<i=.u.#.
a2b5 bea7 692f 5847 a38a dd53 082c add5   ....i/XG...S.,..
5061 b64c 721d 864b 90b6 b55f bb04 735c   Pa.Lr..K..._..s\
9448 6730 5453 df64 813e b603 5795 2242   .Hg0TS.d.>..W."B
e[illegible]8 7454 7322 7cdc b60e 97af 2f64 2728   ..tTs"|...../d'(
[illegible]4bd 2a84 2dfe 50ea 5935 c349 1513   <XYZ@COMPANY.COM
[illegible]e92c a3f8 6e46 0530 8a88 c7a2 5d2b   ...,..nF.0....]+
d89d 77cc fe1e f637 f3f3 d0af 1b47 c09b   ..w....7.....G..
```

It's hard to see compressed email address in bulk data.



```
e327 962d 6450 3d91 c945 3bed 97a6 ...cd   .'.-dP=..E;.....
1...b 0800 0000 0000 0...                  ..............rH.
.....8cc abd4 03d2 0...                    .-H.......N.(|.W
...714 3e00 b455 c1c5 3...                 7.>..U..0.......
0a8e 4ece 287c 1757 3714 3e00 a175 ...ed   ..N.(|.W7.>..u..
```

XYZ@company.com
ABC@company.com
DEF@company.com

Folders.pst

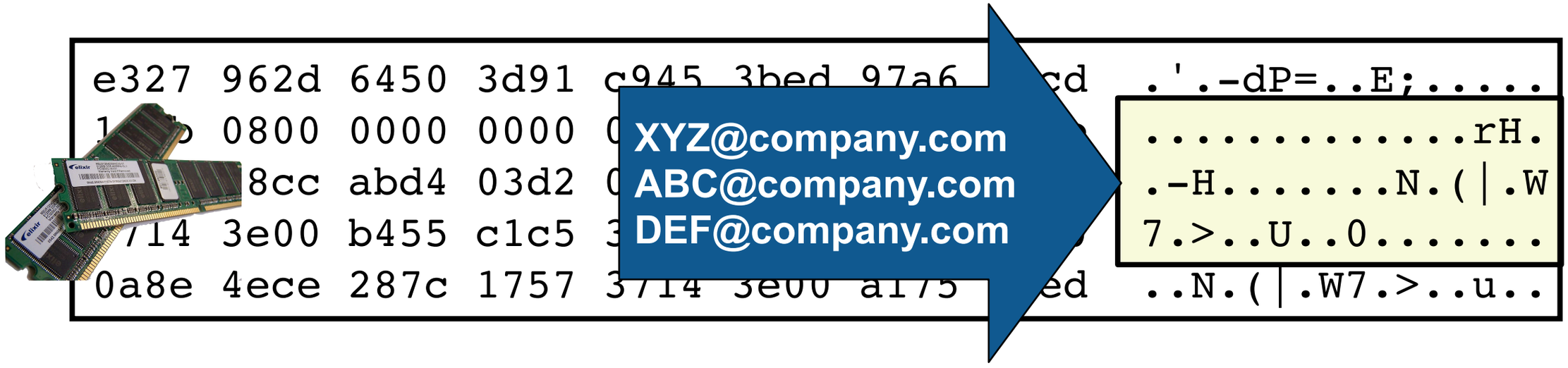
Mother.JPG

Presentation.pptx

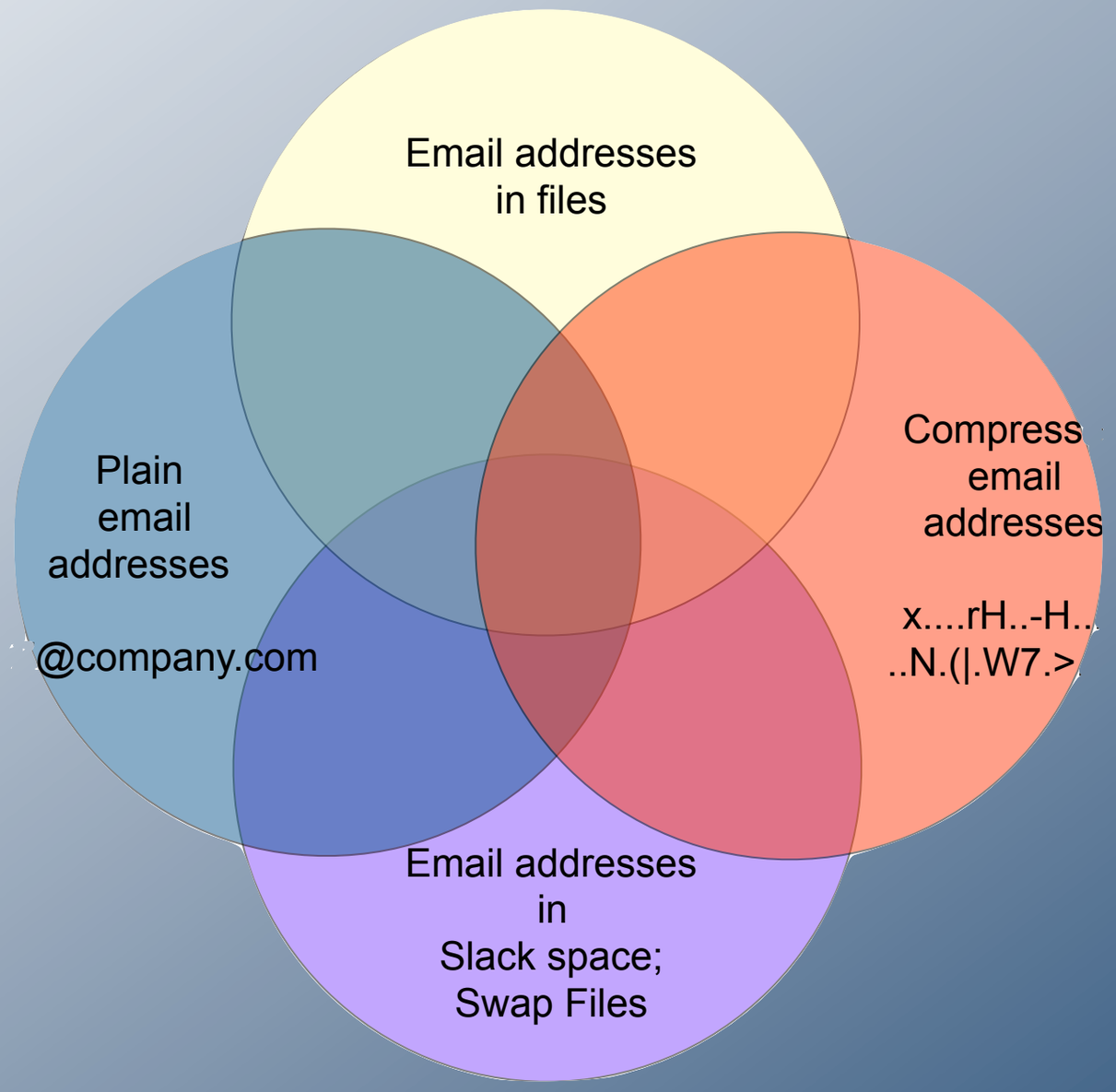
Sequestration.docx

```
a097 83a1 ed96 26a6 3c69 3d0f 750a 2399   ......&.<i=.u.#.
a2b5 bea7 692f 5847 a38a dd53 082c add5   ....i/XG...S.,..
5061 b64c 721d 864b 90b6 b55f bb04 735c   Pa.Lr..K..._..s\
9448 6730 5453 df64 813e b603 5795 2242   .Hg0TS.d.>..W."B
e...8 7454 7322 7cdc b60e 97af 2f64 2728  ..tTs"|...../d'(
....... 4bd 2a84 2dfe 50ea 5935 c349 1513 <XYZ@COMPANY.COM
....9 e92c a3f8 6e46 0530 8a88 c7a2 5d2b  ...,..nF.0....]+
d89d 77cc fe1e f637 f3f3 d0af 1b47 c09b   ..w....7.....G..
```

It’s so hard that none of the commercial digital forensic will show these email addresses.



This is a serious problem.



How big is the problem?

Email addresses can be in files

Files

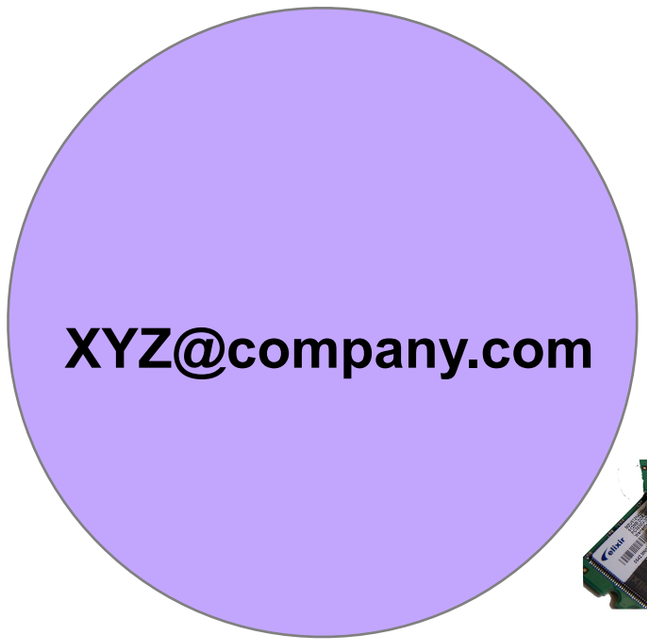
- Documents
- Address book
- Email messages

ABC@company.com
DEF@company.com

Browser Cache:

- Web mail
- Facebook Data

Email addresses can be in non-file disk sectors



Swap Files
Hibernation Files
File fragments

Some may be in *both* files and in non-files.
(A file that's read into RAM before the system hibernates.)

ABC@company.com
DEF@company.com

Alice@net.net

Alice@net.net

XYZ@company.com

Swap Files
Hibernation Files
File fragments

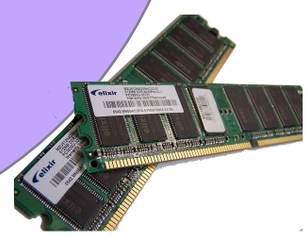
We can use a Venn Diagram to represent email addresses on the media.

ABC@company.com
DEF@company.com

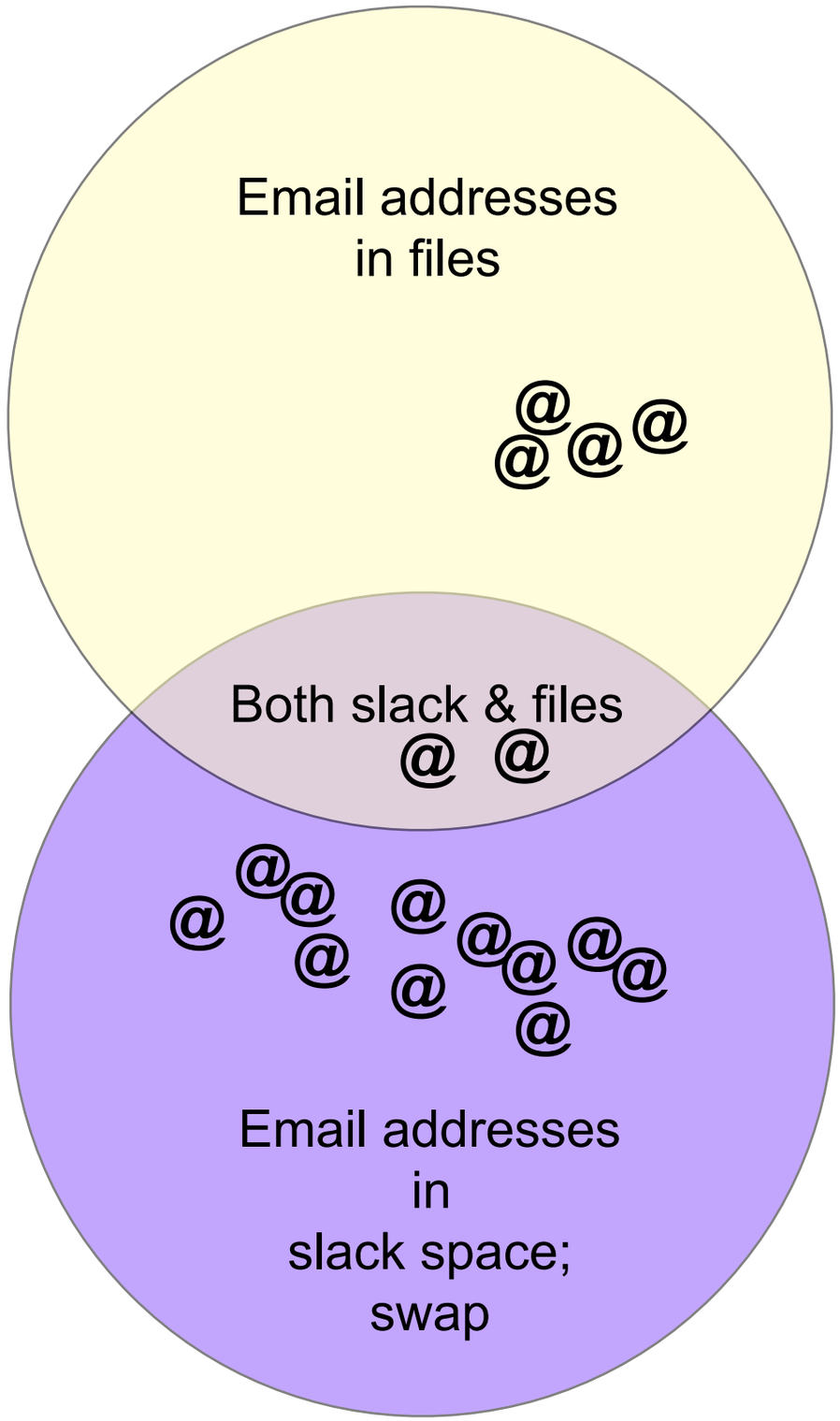
Alice@net.net

XYZ@company.com

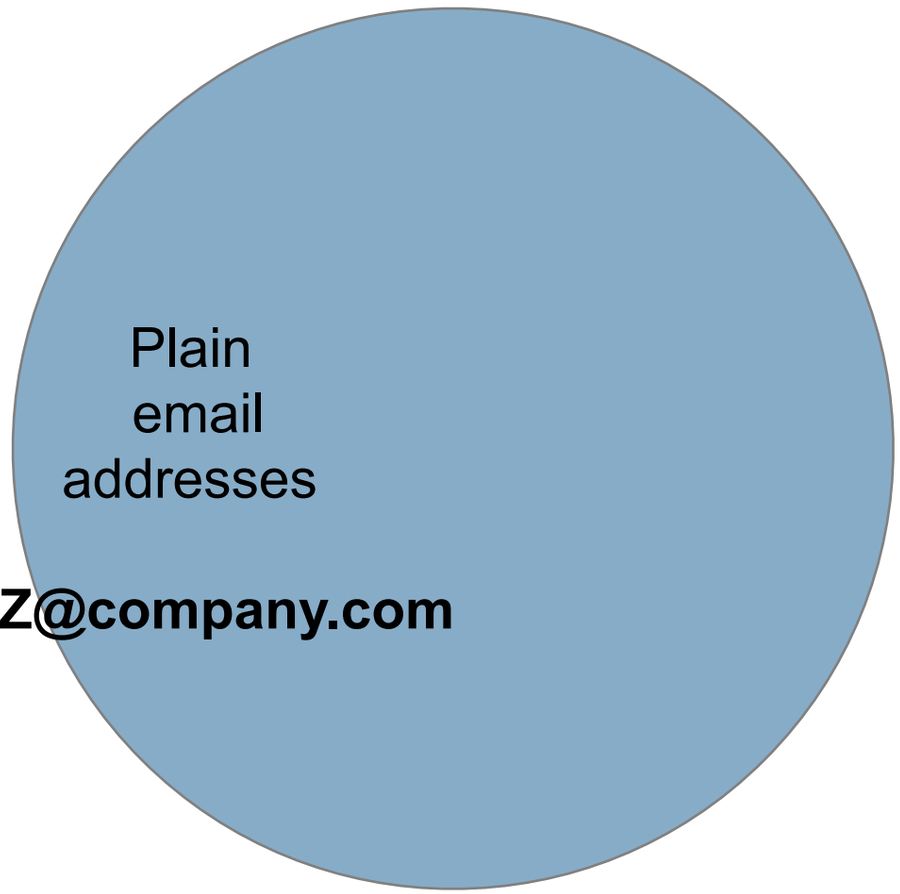
Swap Files
Hibernation Files
File fragments



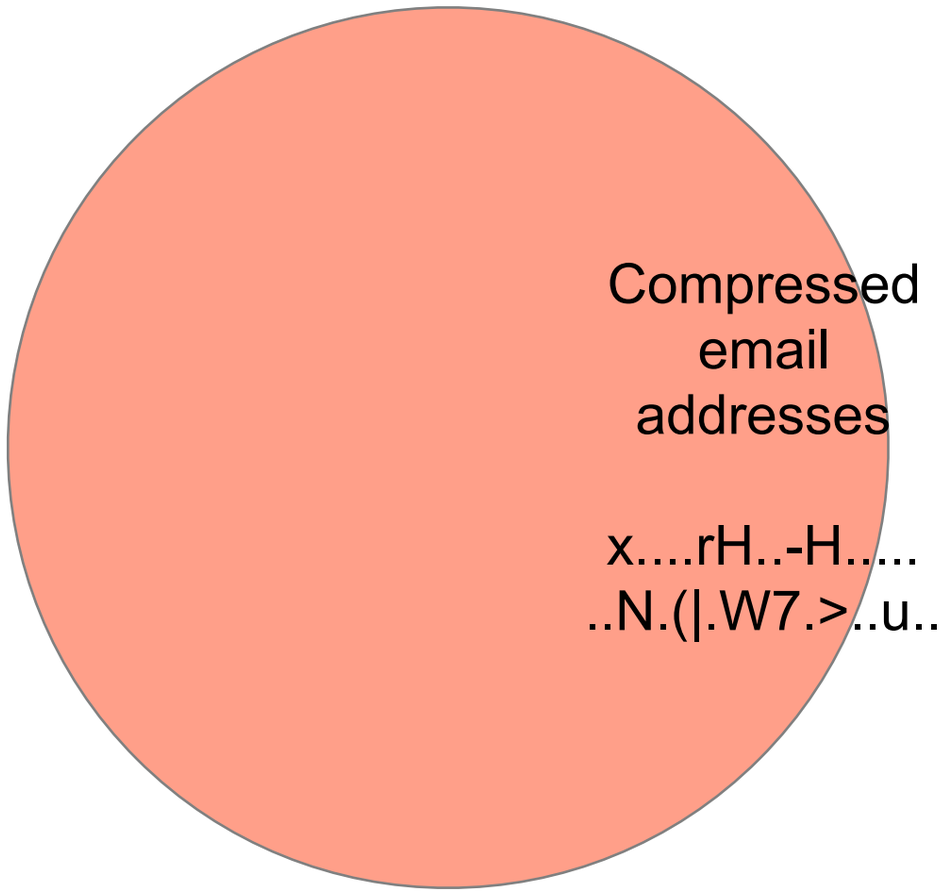
The number of email addresses in each region depends on the media.



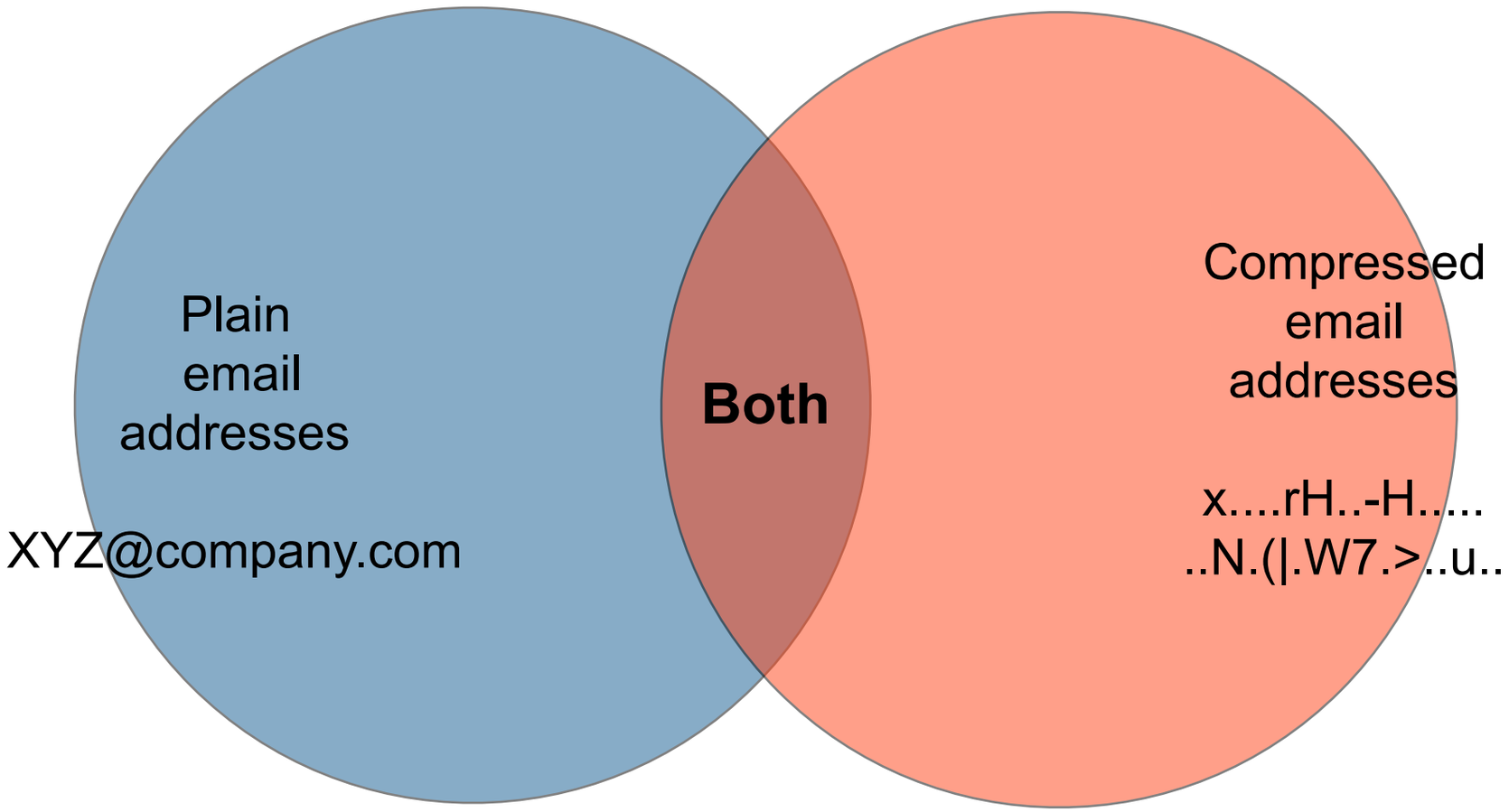
Email addresses can be plain text.
“XYZ@company.com”



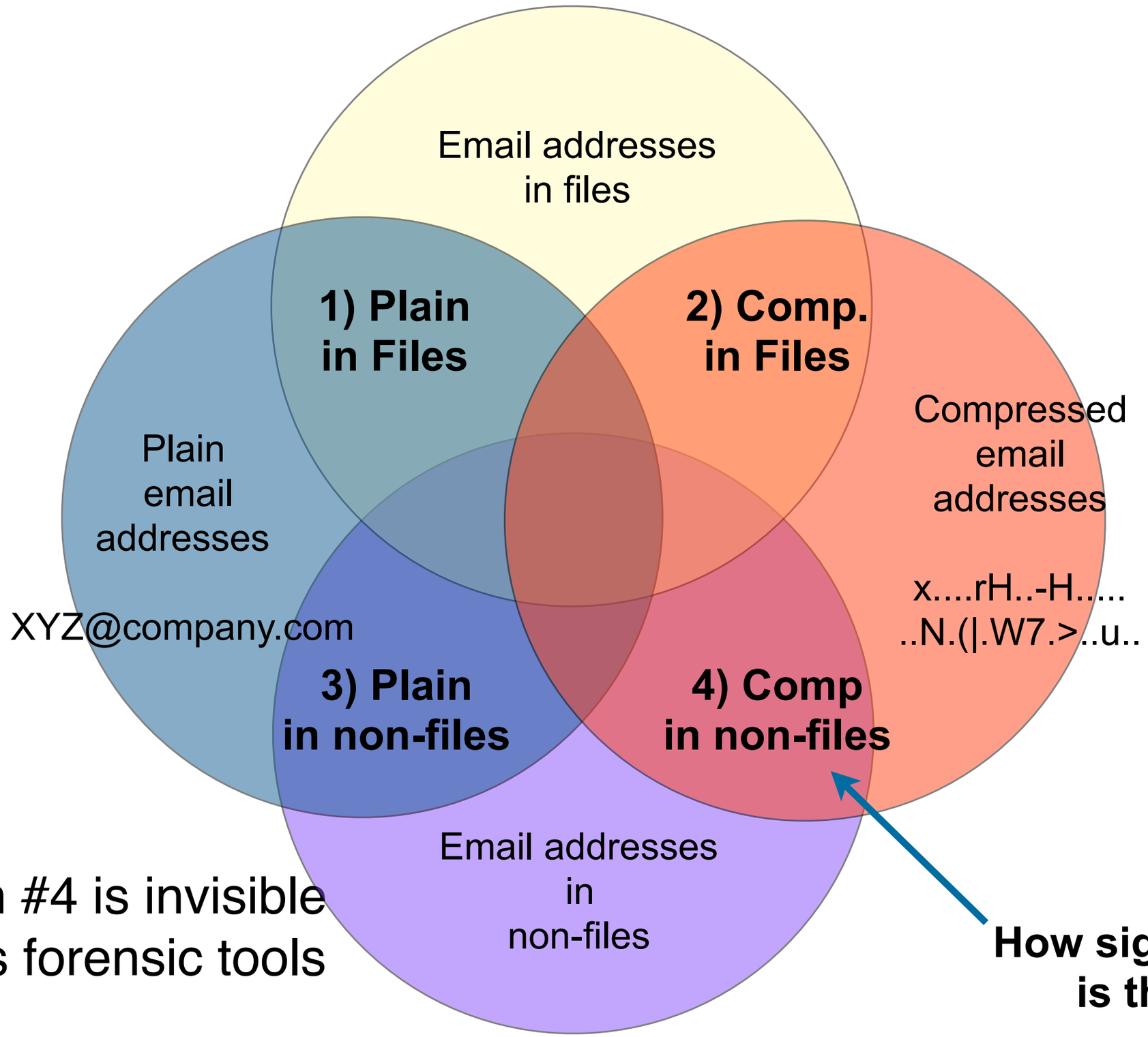
Email addresses can be compressed or encoded. “x....rH..-H.....N.(I.W7.>..u..”



Each email address can be present plain, compressed, or both.



There are four different conditions for an email address on the media.

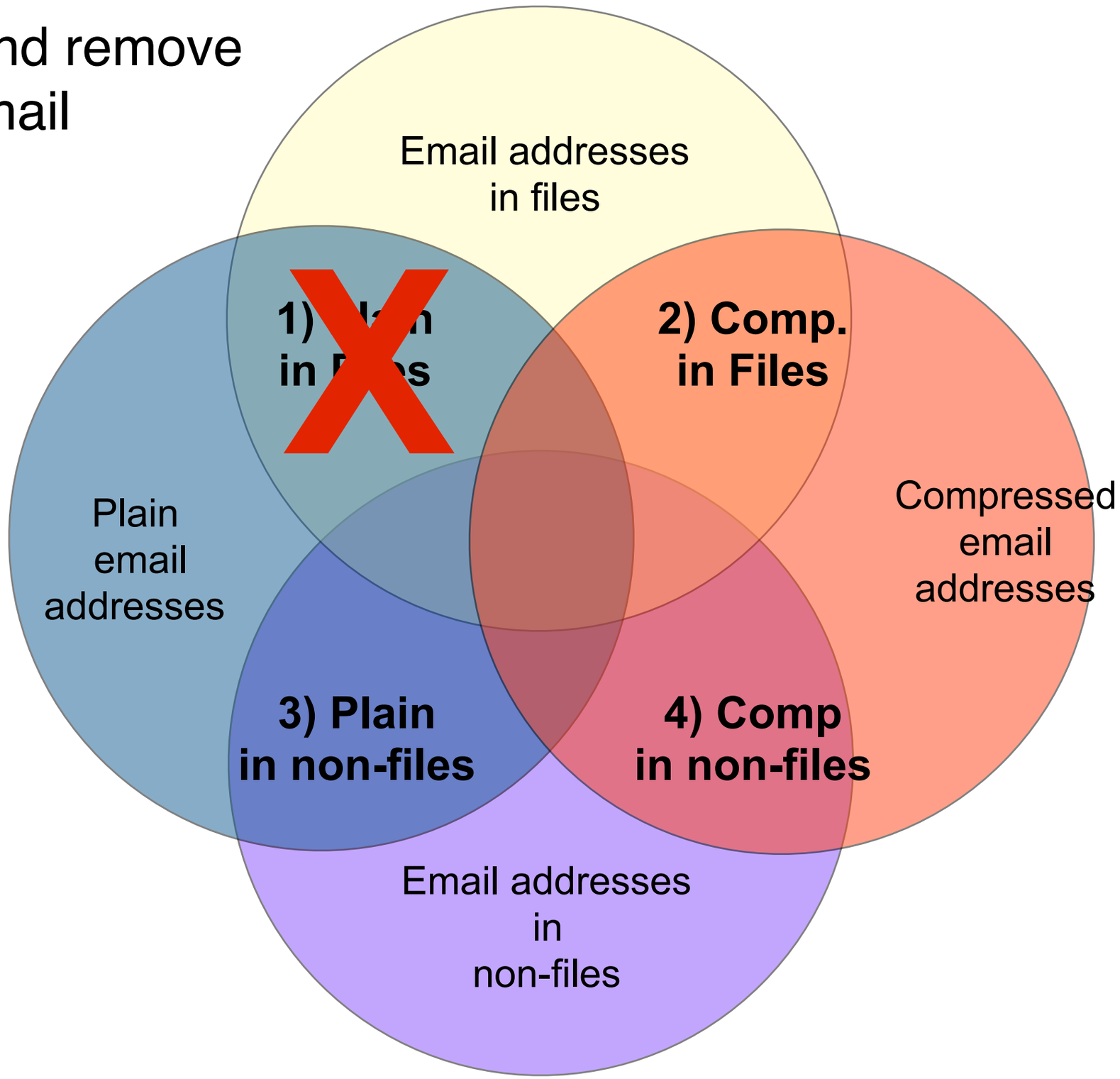


Condition #4 is invisible to today's forensic tools

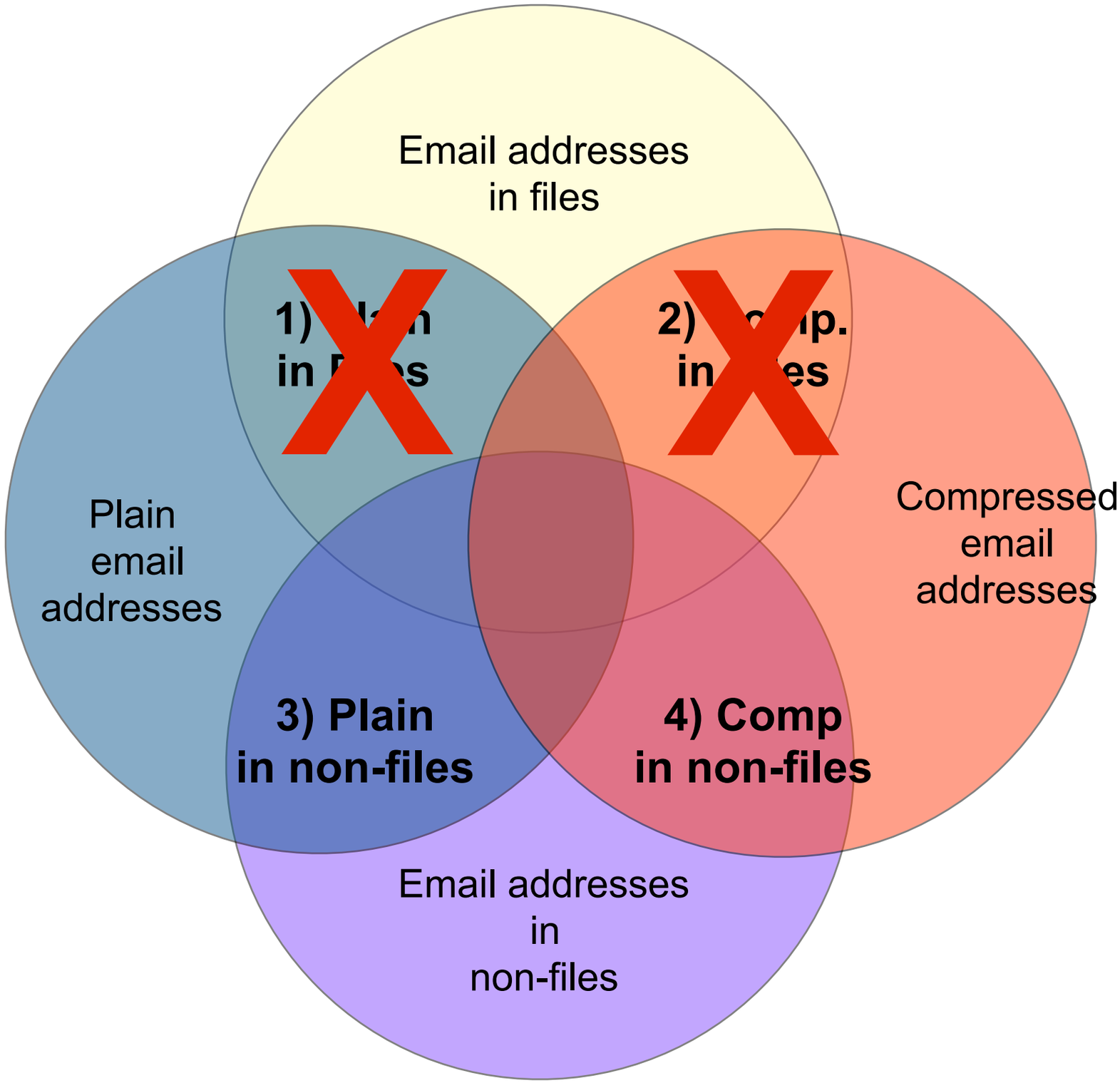
How significant is this?

We devised an experiment to determine the size of condition #4 for a specific drive.

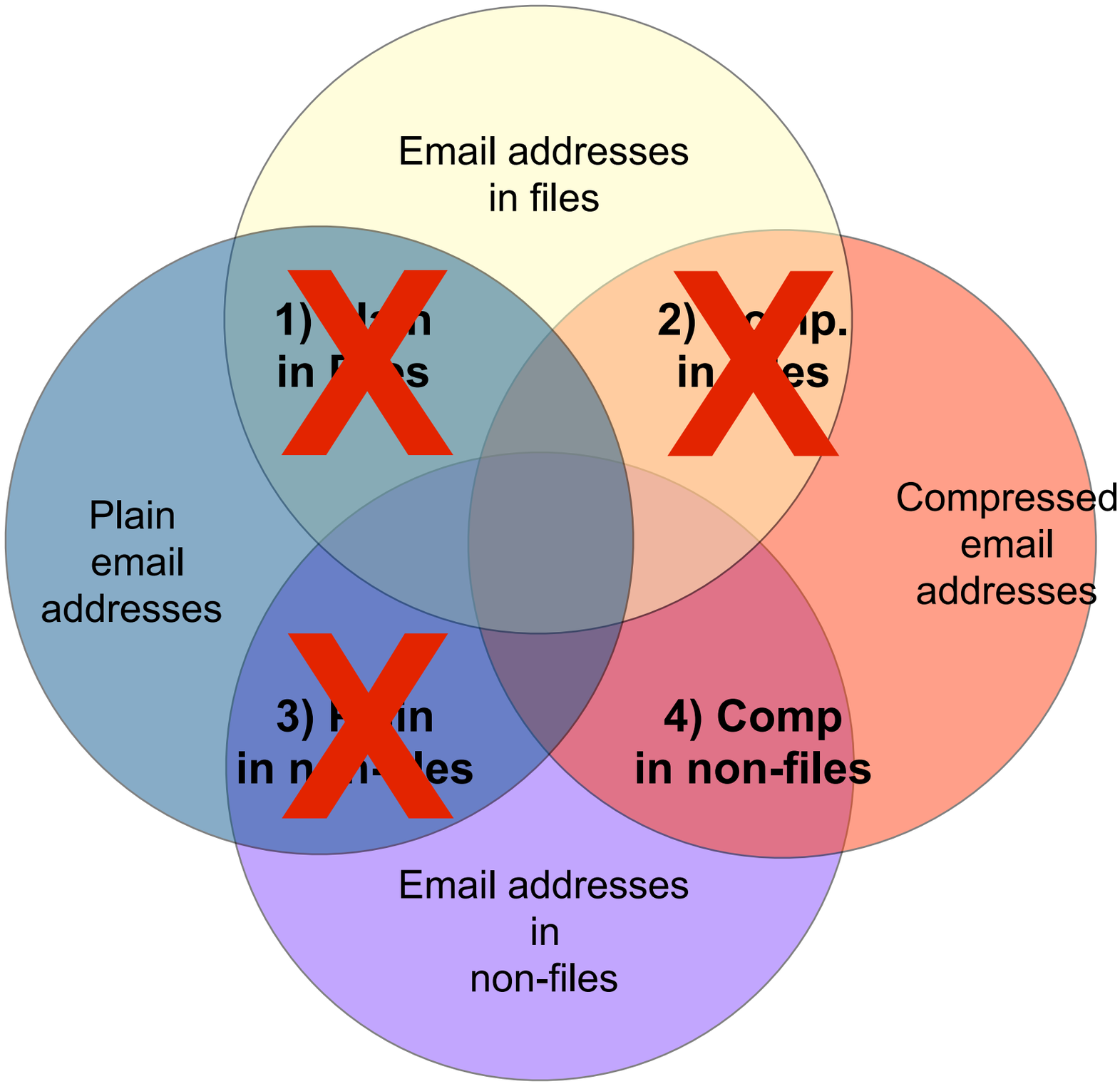
First, find and remove the plain email addresses in files.



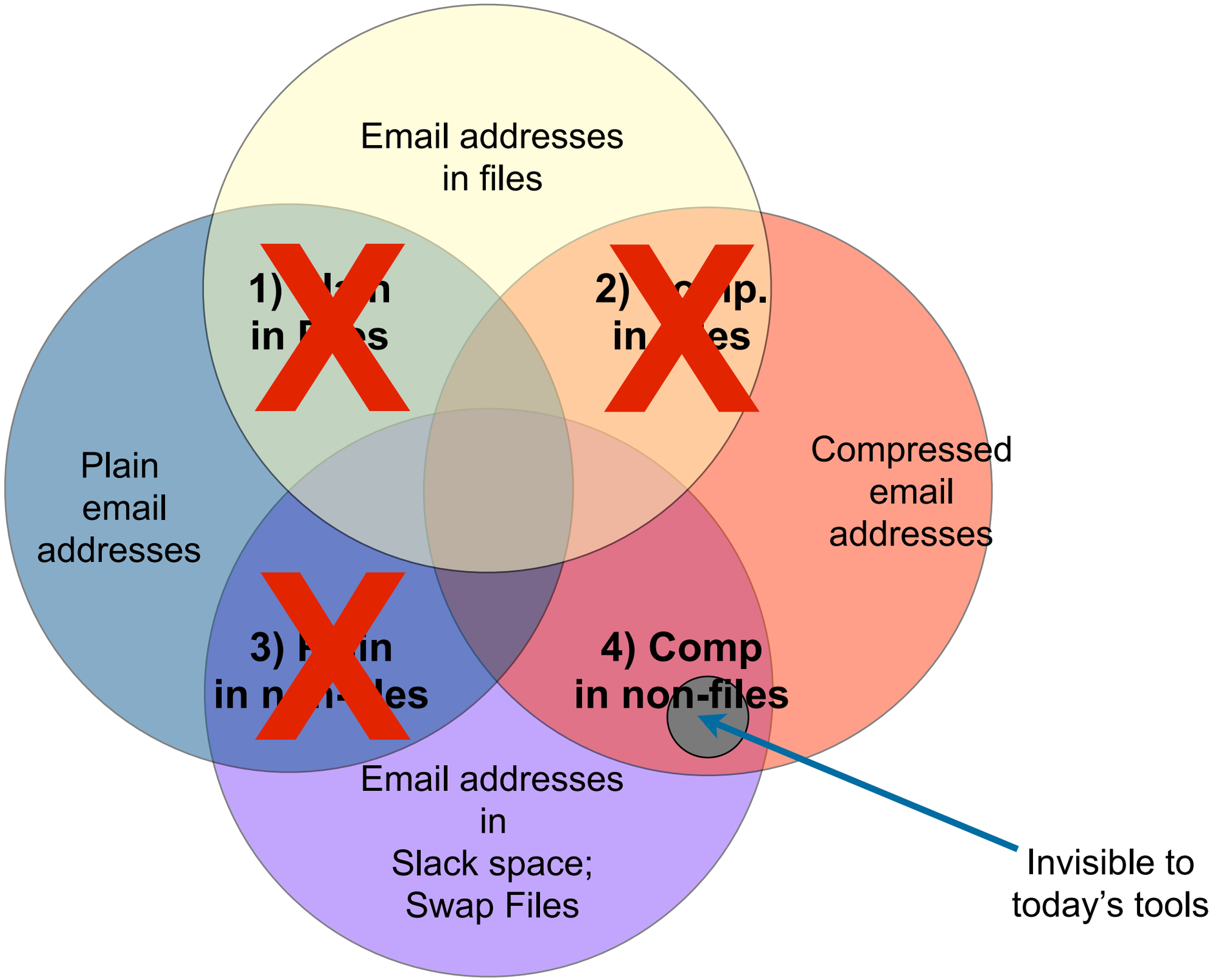
...Remove the addresses compressed and in files....



...Remove email addresses that are not compressed.

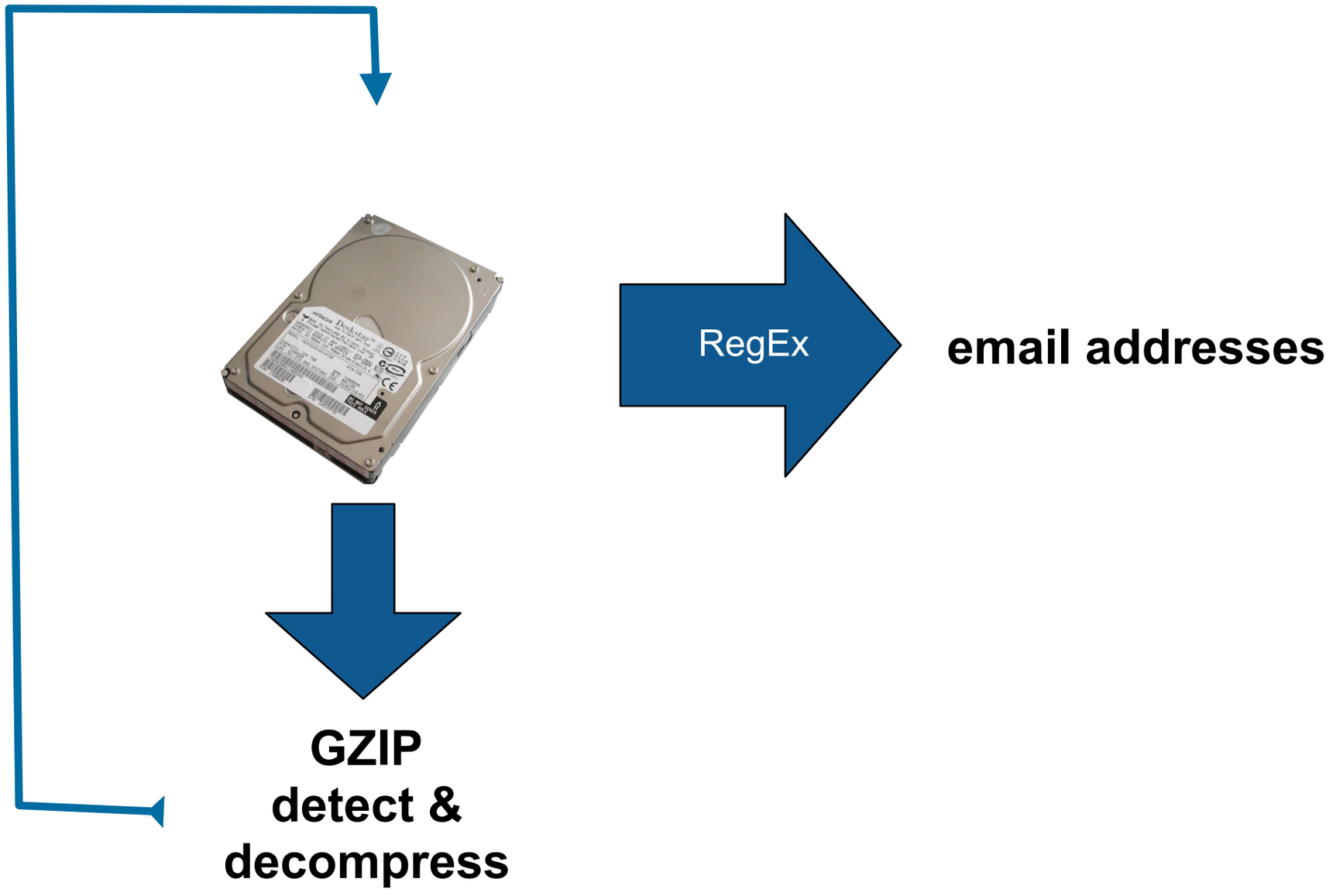


...those that remain are the “invisible” email addresses.



bulk_extractor is an experimental email extraction tool.

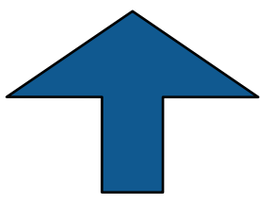
"Digital media triage with bulk data analysis and bulk_extractor,"
Simson L. Garfinkel, *Computers and Security 32 (2013) 56-72*



bulk_extractor can find both plain and compressed text.

“Feature files” contain the extracted email addresses.

```
# UTF-8 Byte Order Marker; see http://unicode.org/faq/utf_bom.html
#
@
...
392175418    WindowsXP@gn.microsoft.com    Name=WindowsXP@gn.microsoft.com\015\012
...
3772517888-GZIP-28322  user@company.com  onterey-<nobr>user@company.com</nobr>
...
```



Offset

Feature

Context

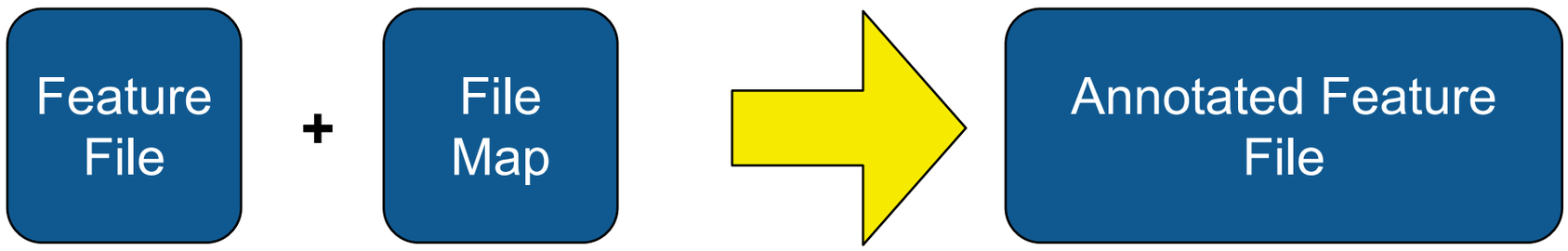
Plain text features have numeric offsets:

`392175418`

Compressed features will indicate the algorithm:

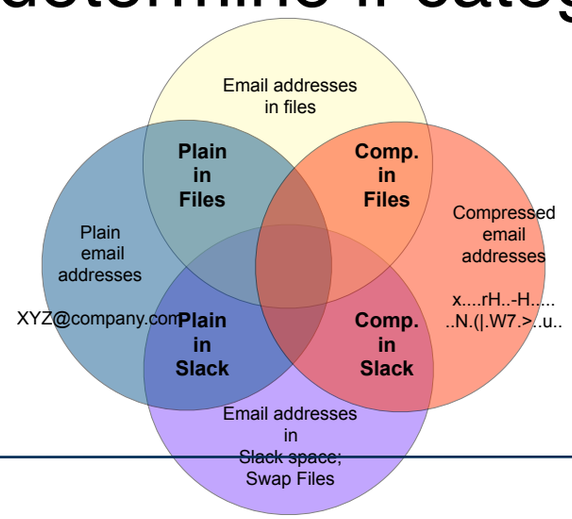
`3772517888-GZIP-28322`

Post-processing with identify_files.py reveals file names




```
Offset:  392175418
Feature: WindowsXP@gn.microsoft.com
Context: \012[User]\015\012Name=WindowsXP@gn.microsoft.com
         \015\012Password=B@ji0
Filename:WINDOWS/system32/oobe/migx25a.dun
MD5:     2b00042f7481c7b056c4b410d28f33cf
```

For each feature, we can determine if category #1, #2, #3 and #4!



bulk_extractor 1.3.2 recognizes a wide variety of features and encoding types:

Feature types:

- Domain Names; Email addresses; URLs, CCNs
- Search terms; Facebook IDs; JSON data
- KML files; EXIF data
- VCARDs
- word search output
- PCAP files; Ethernet Addresses; TCP/IP Connections; etc.
- ELF & PE headers; Windows Prefetch files

```
-rw-r--r--@ 1 simsong  staff       476 Jul  7 23:50 aes_keys.txt
-rw-r--r--@ 1 simsong  staff         0 Jul  7 23:48 alerts.txt
-rw-r--r--@ 1 simsong  staff      2743 Jul  7 23:59 ccn.txt
-rw-r--r--@ 1 simsong  staff       454 Jul  8 00:03 ccn_histogram.txt
-rw-r--r--@ 1 simsong  staff         0 Jul  7 23:48 ccn_track2.txt
-rw-r--r--@ 1 simsong  staff         0 Jul  8 00:03 ccn_track2_histogram.txt
-rw-r--r--@ 1 simsong  staff  23369167 Jul  8 00:03 domain.txt
-rw-r--r--@ 1 simsong  staff    185266 Jul  8 00:03 domain_histogram.txt
-rw-r--r--@ 1 simsong  staff         0 Jul  7 23:48 elf.txt
-rw-r--r--@ 1 simsong  staff   1719842 Jul  8 00:03 email.txt
-rw-r--r--@ 1 simsong  staff     35073 Jul  8 00:03 email_histogram.txt
-rw-r--r--@ 1 simsong  staff     23961 Jul  8 00:00 ether.txt
-rw-r--r--@ 1 simsong  staff       337 Jul  8 00:03 ether_histogram.txt
-rw-r--r--@ 1 simsong  staff  11188830 Jul  8 00:03 exif.txt
-rw-r--r--@ 1 simsong  staff         0 Jul  7 23:48 find.txt
-rw-r--r--@ 1 simsong  staff      1112 Jul  8 00:01 gps.txt
-rw-r--r--@ 1 simsong  staff         0 Jul  7 23:48 hex.txt
-rw-r--r--@ 1 simsong  staff     95835 Jul  8 00:03 ip.txt
-rw-r--r--@ 1 simsong  staff     11603 Jul  8 00:03 ip_histogram.txt
-rw-r--r--@ 1 simsong  staff   2025702 Jul  8 00:03 json.txt
-rw-r--r--@ 1 simsong  staff         0 Jul  7 23:48 kml.txt
-rw-r--r--@ 1 simsong  staff    194991 Jul  8 00:03 packets.pcap
-rw-r--r--@ 1 simsong  staff     21343 Jul  8 00:03 report.xml
-rw-r--r--@ 1 simsong  staff   3782598 Jul  8 00:03 rfc822.txt
-rw-r--r--@ 1 simsong  staff    213746 Jul  8 00:03 tcp.txt
-rw-r--r--@ 1 simsong  staff     61255 Jul  8 00:03 tcp_histogram.txt
-rw-r--r--@ 1 simsong  staff     59469 Jul  8 00:03 telephone.txt
-rw-r--r--@ 1 simsong  staff      6612 Jul  8 00:03 telephone_histogram.txt
-rw-r--r--@ 1 simsong  staff  67205326 Jul  8 00:03 url.txt
-rw-r--r--@ 1 simsong  staff         0 Jul  8 00:03 url_facebook-id.txt
-rw-r--r--@ 1 simsong  staff   5706665 Jul  8 00:03 url_histogram.txt
-rw-r--r--@ 1 simsong  staff         0 Jul  8 00:03 url_microsoft-live.txt
-rw-r--r--@ 1 simsong  staff      8504 Jul  8 00:03 url_searches.txt
-rw-r--r--@ 1 simsong  staff    151673 Jul  8 00:03 url_services.txt
-rw-r--r--@ 1 simsong  staff         0 Jul  7 23:48 vcard.txt
-rw-r--r--@ 1 simsong  staff  18549729 Jul  8 00:03 windirs.txt
-rw-r--r--@ 1 simsong  staff  29051041 Jul  8 00:03 winpe.txt
-rw-r--r--@ 1 simsong  staff   1984759 Jul  8 00:03 winprefetch.txt
-rw-r--r--@ 1 simsong  staff  34128889 Jul  8 00:03 zip.txt
```

Encoding Types:

- ZIP; GZIP; Windows Hibernation
- BASE16, BASE64

Some drives have a lot of compressed data

This drive contains a GZIP stream in a Windows Hibernation File.

```
...
…6464-HIBER-49691-GZIP-1526   groups-noreply@linkedin.com    3d\134"groups-noreply@linkedin.com
…6464-HIBER-49691-GZIP-2018   m*****************@gmail.com   3d\134"m*****************@gmail.co
…6464-HIBER-49691-GZIP-2128   sur*******1@gmail.com          3d\134"sur*******1@gmail.com\134"\
…6464-HIBER-49691-GZIP-2625   *******.consultancy@gmail.com 3d\134"*******.consultancy@gmail.c
…6464-HIBER-49691-GZIP-2736   sur*******1@gmail.com          3d\134"sur*******1@gmail.com\134"\
…6464-HIBER-49691-GZIP-3186   san****@***********.com        \134" "san****@***********.com\134"\134u
…6464-HIBER-49691-GZIP-3685   Careers@******bank.com         3d\134"Careers@******bank.com\134"
…6464-HIBER-49691-GZIP-4124   par****@team******.com         3d\134"par****@team******.com\134"
…6464-HIBER-49691-GZIP-4149   u003epar****@team******.com    \134u003epar****@team******.com\13
…6464-HIBER-49691-GZIP-4607   d****.*****@gmail.com          3d\134"d****.*****@gmail.com\134"\
…6464-HIBER-49691-GZIP-4631   u003ed****.*****@gmail.com     \134u003ed****.*****@gmail.com\134
…6464-HIBER-49691-GZIP-5114   raj******@bsnl.in              3d\134"raj******@bsnl.in\134"\134u
…6464-HIBER-49691-GZIP-5558   kiran.***@****technology.com   3d\134"kiran.***@****technology.co
…6464-HIBER-49691-GZIP-5671   sur*******1@gmail.com          3d\134"sur*******1@gmail.com\134"\
...
```

- JSON object downloaded from Facebook by compressed HTTP
- In RAM, written to HIBER on disk when the system went into sleep.

We ran bulk_extractor and identify_filenames.py on drive IN10-0138 and examined the email encodings:

Emails seen	count	1) Plain in Files	2) Comp. in Files	3) Plain in non-files	4) Comp in non-files
Cleartext		358	--	5341	--
All Comp		--	9	--	135
GZIP	50	13	1	22	14
HIBER	39	6	1	27	5
HIBER-GZIP	23			21	2
PDF	88	1		9	78
ZIP	28	2	5	3	18
ZIP-PDF	18				18

135 out of 5700 email addresses are invisible to existing tools.

Many of these email addresses are significant

Example email addresses (sanitized)

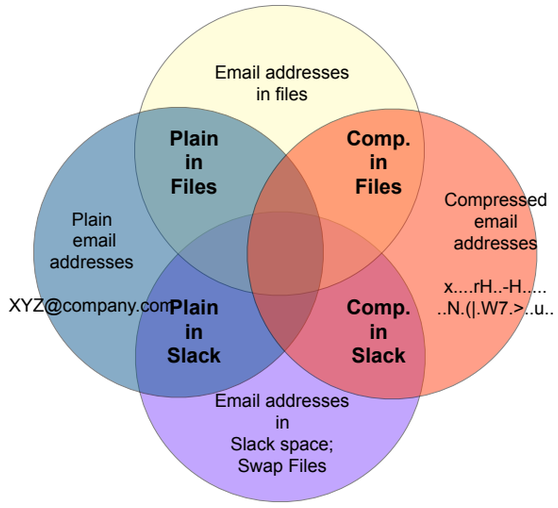
Encoding	Email Address (*Sanitized)	Note
GZIP	****@*****.dk	PII
ZIP	******@desktopsidebar.com	PII
HIBER	ntIV@std.do	false positive
ZIP	****************@digital.com	source code?
ZIP	pcg@goof.com	ECGS Compiler
ZIP	andrew@northwindtraders.com	MS Office Sample
ZIP	ActiveSh@eet.Na	false positive
GZIP	linux-ntfs-dev@lists.sourceforge.net	mailing list

Questions:

- How common are compressed email addresses in unallocated space?
- Is this technique worth the effort?

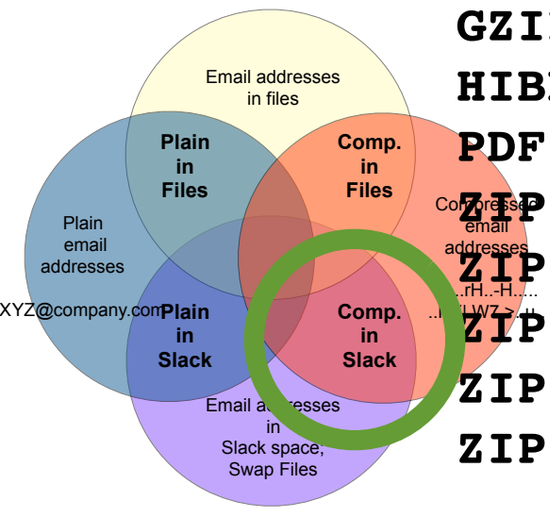
Analysis of 1,646 disk images (Including email addresses present in cleartext)

Coding	Drives	Emails	avg	max	σ
(CLEARTEXT)	949	2,043,168	2,152	178,073	8,890
ZIP	426	86,259	202	59,369	2,887
GZIP	261	79,351	304	9,111	1,035
GZIP-GZIP	17	12,676	745	11,845	2,778
PDF	186	2,569	13	238	30
HIBER	85	1,481	17	220	43
ZIP-ZIP	74	470	6	48	8
ZIP-GZIP	18	307	17	132	31
BASE64	56	250	4	50	7
ZIP-PDF	28	125	4	18	4
ZIP-BASE64-GZIP	2	65	32	38	5
BASE64-GZIP	2	65	32	38	5
GZIP-GZIP-GZIP	4	58	14	38	14
GZIP-ZIP	7	54	7	30	9
GZIP-BASE64	7	44	6	11	3
GZIP-PDF	5	38	7	30	11
GZIP-GZIP-BASE64	2	38	19	30	11
ZIP-BASE64	5	30	6	13	5
GZIP-GZIP-ZIP	1	12	12	12	0
ZIP-ZIP-ZIP	4	10	2	6	2
HIBER-GZIP	1	2	2	2	0
BASE64-GZIP-GZIP	2	2	1	1	0
ZIP-BASE64-GZIP-GZIP	2	2	1	1	0
ZIP-ZIP-PDF	1	1	1	1	0



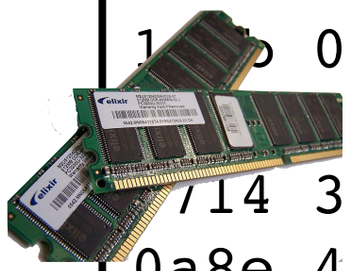
Analysis of 1,646 disk images

Coding	Drives	Emails	avg	max	σ
1) Plain in files	739	81,920	110	4,206	253
2) Comp in files	355	19,711	55	5,454	388
3) Plain in non-files	860	1,956,059	2,274	178,073	9,248
4) Comp in non-files	474	165,481	349	59,376	2,889
BASE64 Comp	54	219	4	50	7
BASE64-GZIP Comp	2	64	32	37	5
GZIP Comp	234	66,195	282	9,103	981
GZIP-BASE64 Comp	7	44	6	11	3
GZIP-GZIP Comp	15	12,663	844	11,845	2,944
GZIP-GZIP-BASE64 Comp	2	38	19	30	11
GZIP-GZIP-GZIP Comp	4	58	14	38	14
GZIP-GZIP-ZIP Comp	1	12	12	12	0
GZIP-PDF Comp	5	38	7	30	11
GZIP-ZIP Comp	6	49	8	30	9
HIBER Comp	79	1,433	18	217	44
PDF Comp	162	2,352	14	238	31
ZIP Comp	388	85,252	219	59,369	3,025
ZIP-BASE64 Comp	5	30	6	13	5
ZIP-BASE64-GZIP Comp	2	65	32	38	5
ZIP-GZIP Comp	14	261	18	132	34
ZIP-PDF Comp	26	115	4	18	4



- Conclusion: This is a big deal! Lots of email addresses are being missed.

Some drives have more than TEN THOUSAND email addresses that are compressed and not in a file.



```
e327 962d 6450 3d91 c945 3bed 97a6 [illegible]cd   .'.-dP=..E;.....
[illegible]0 0800 0000 0000 0[illegible]  .............rH.
[illegible]8cc abd4 03d2 0[illegible]  .-H.......N.(|.W
[illegible]714 3e00 b455 c1c5 3[illegible]  7.>..U..0.......
0a8e 4ece 287c 1757 3714 3e00 a175 [illegible]ed   ..N.(|.W7.>..u..
```

XYZ@company.com
ABC@company.com
DEF@company.com

Folders.pst

Mother.JPG

Presentation.pptx

Sequestration.docx

```
a097 83a1 ed96 26a6 3c69 3d0f 750a 2399   ......&.<i=.u.#.
a2b5 bea7 692f 5847 a38a dd53 082c add5   ....i/XG...S.,..
5061 b64c 721d 864b 90b6 b55f bb04 735c   Pa.Lr..K..._..s\
9448 6730 5453 df64 813e b603 5795 2242   .Hg0TS.d.>..W."B
[illegible]8 7454 7322 7cdc b60e 97af 2f64 2728   ..tTs"|...../d'(
[illegible]4bd 2a84 2dfe 50ea 5935 c349 1513   <XYZ@COMPANY.COM
[illegible]e92c a3f8 6e46 0530 8a88 c7a2 5d2b   ...,..nF.0....]+
d89d 77cc fe1e f637 f3f3 d0af 1b47 c09b   ..w....7.....G..
```

(Compressed email in files are also ignored...)

"Digital media triage with bulk data analysis and bulk_extractor,"
Simson L. Garfinkel, *Computers and Security 32 (2013) 56-72*

email address	Application (encoding)	*strings* & *grep*	EnCase	BE
plain_text@textedit.com	Apple TextEdit (UTF-8)	✓	✓	✓
plain_text_pdf@textedit.com	Apple TextEdit print-to-PDF (/FlateDecode)			✓
rtf_text@textedit.com	Apple TextEdit (RTF)	✓	✓	✓
rtf_text_pdf@textedit.com	Apple TextEdit print-to-PDF (/FlateDecode)			✓
plain_utf16@textedit.com	Apple TextEdit (UTF-16)		✓	✓
plain_utf16_pdf@textedit.com	Apple TextEdit print-to-PDF (/FlateDecode)			✓
pages@iwork09.com	Apple Pages '09	✓	✓	✓
pages_comment@iwork09.com	Apple Pages (comment) '09			✓
keynote@iwork09.com	Apple Keynote '09			✓
keynote_comment@iwork09.com	Apple Keynote '09 (comment)			✓
numbers@iwork09.com	Apple Numbers '09			✓
numbers_comment@iwork09.com	Apple Numbers '09 (comment)			✓
user_doc@microsoftword.com	Microsoft Word 2008 (Mac) (.doc file)	✓	✓	✓
user_doc_pdf@microsoftword.com	Microsoft Word 2008 (Mac) print-to-PDF			
user_docx@microsoftword.com	Microsoft Word 2008 (Mac) (.docx file)			✓
user_docx_pdf@microsoftword.com	Microsoft Word 2008 (Mac) print-to-PDF (.docx file)			
xls_cell@microsoft_excel.com	Microsoft Word 2008 (Mac)	✓	✓	✓
xls_comment@microsoft_excel.com	Microsoft Word 2008 (Mac)			✓
xlsx_cell@microsoft_excel.com	Microsoft Word 2008 (Mac)			✓
xlsx_cell_comment@microsoft_excel.com	Microsoft Word 2008 (Mac) (Comment)			✓
doc_within_doc@document.com	Microsoft Word 2007 (OLE .doc file within .doc)	✓	✓	✓
docx_within_docx@document.com	Microsoft Word 2007 (OLE .doc file within .doc)	✓	✓	✓
ppt_within_doc@document.com	Microsoft PowerPoint and Word 2007 (OLE .ppt file within .doc)	✓	✓	✓
pptx_within_docx@document.com	Microsoft PowerPoint and Word 2007 (OLE .pptx file within .docx)			✓
xls_within_doc@document.com	Microsoft Excel and Word 2007 (OLE .xls file within .doc)	✓	✓	✓
xlsx_within_docx@document.com	Microsoft Excel and Word 2007 (OLE .xlsx file within .docx)			✓
email_in_zip@zipfile1.com	text file within ZIP			✓
email_in_zip_zip@zipfile2.com	ZIP'ed text file, ZIP'ed			✓
email_in_gzip@gzipfile.com	text file within gzip			✓
email_in_gzip_gzip@gzipfile.com	gzip'ed text file, gzip'ed			✓

21 out of 30 compressed email addresses in test files were ignored.

There are many sources of compressed data. Today's tools ignore these data when not in files.

Documents:

- Microsoft Office (.docx, .xlsx, .pptx); PDF files (text is compressed)
- Browser Cache (downloads are compressed)

Archives:

- ZIP files; GZIP (GZ) files

System Resources:

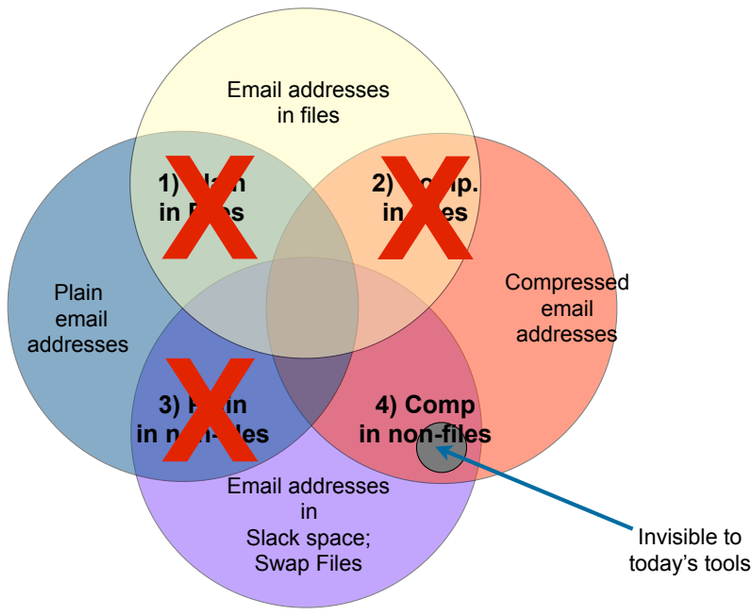
- Hibernation files & file fragments

If forensic examiners miss an email address:

- A perpetrator or an accomplice may not be identified
- Media may not be associated with a crime

In summary: Compressed emails in non-file space are being systematically ignored. It's a serious problem.

Important, relevant data is hidden by today's tools.



I demonstrated the extent of the problem with:

- bulk_extractor, a high-performance stream-based feature extractor
  - *https://github.com/simsong/bulk_extractor (dev tree)*
  - *http://digitalcorpora.org/downloads/bulk_extractor (downloads)*
  - *http://www.sciencedirect.com/science/article/pii/S1674048120014 72 (paper)*
  - *http://simson.net/clips/academic/2013.COSE.bulk_extractor.pdf*
- Real Data Corpus:
  - *http://digitalcorpora.org/*