Digital Forensics Innovation: Searching A Terabyte of Data in 10 minutes

Simson L. Garfinkel
Associate Professor, Naval Postgraduate School

Jan 18, 2013

http://simson.net/

https://domex.nps.edu/deep/

NPS is the Navy's Research University.

Monterey, CA — 1500 students

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

Graduate Schools of Operational & Information Sciences (GSOIS)

- Computer Science
- Defense Analysis
- Information Sciences
- Operations Research
- Cyber Academic Group

National Capital Region (NCR) Office

- 900 N Glebe (Ballston)/Virginia Tech building

The Digital Evaluation and Exploitation (DEEP) Group: Research in “trusted” systems and exploitation.

“Evaluation”

- Trusted hardware and software
- Cloud computing

“Exploitation”

- MEDEX — “Media” — Hard drives, camera cards, GPS devices.
- CELEX — Cell phone
- DOCEX — Documents
- DOMEX — Document & Media Exploitation

Current Partners:

- Law Enforcement (FBI & Local)
- DHS (HSARPA; Video Games & Insider Threat)
- NSF (Courseware development)
- DoD

Digital information is pervasive in today's society. Attorneys, judges and juries are not digital experts.

Many potential sources of digital evidence:

- Laptops; Cell Phones; Email messages

Devices that might have evidence

?

Court proceedings

Many possible goals:

- Establish possession of contraband information (child pornography, credit card #s)
- Recover stolen information
- Document a conspiracy (stock fraud; murder-for-hire)

The digital forensics process makes *digital evidence* available for [legal] decisions

Preparation: policy, training & tools

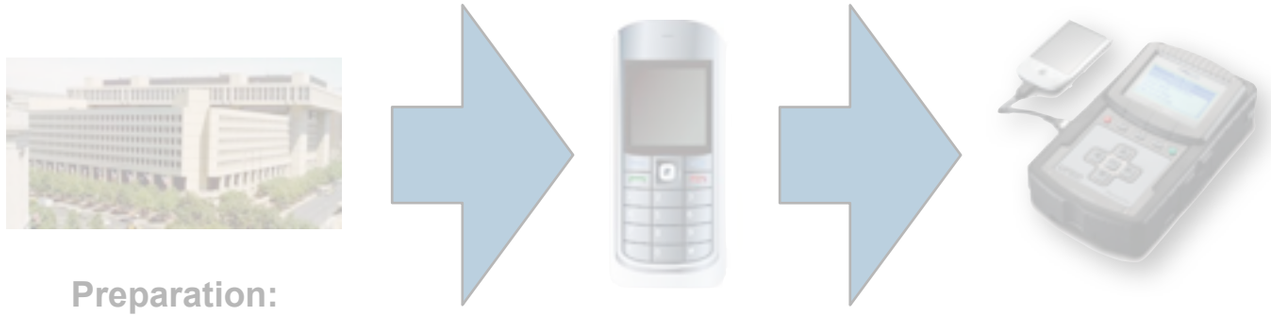
Collect & preserve evidence

Extract preserved data

Analysis

Reporting & Testimony

Most digital forensics focuses on the first half.



Preparation: policy, training & tools

Collect & preserve evidence

Extract preserved data

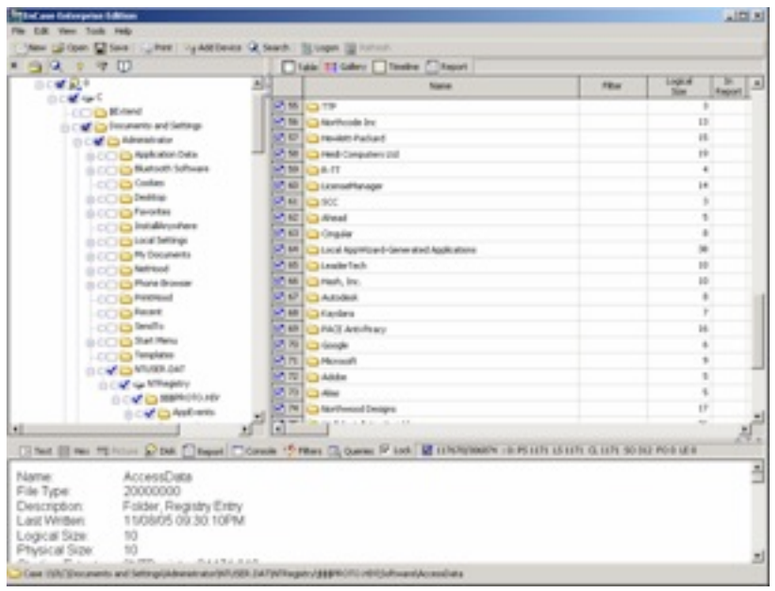
Training the force

“Write Blocker”

Hard Drive from desktop

“Imaging” tools

Reverse engineering to understand data structures



EnCase Forensic

My focus is developing better analysis approaches

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*

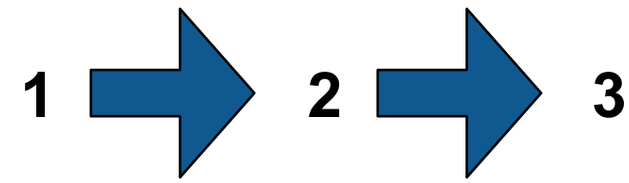
Presentation and Integration:

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

Three principles underly my research:

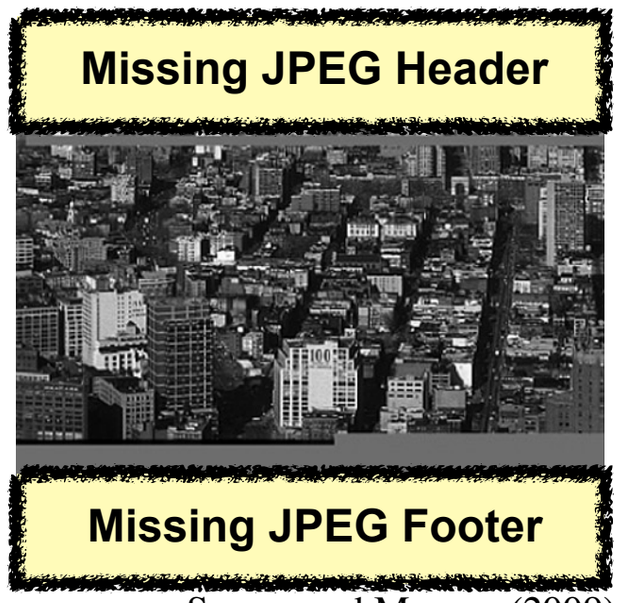
1. Automation is essential.

- Today most forensic analysis is done manually.
- We are developing techniques & tools to allow automation.



2. Concentrate on the invisible.

- It's *easy* to wipe a computer….
  — *but targets don't erase what they can't see.*
- So we look for:
  —*Deleted and partially overwritten files.*
  —*Fragments of memory in swap & hibernation.*
  —*Tool marks.*



Sencar and Memon (2009)

3. Large amounts of data is essential.

- Most research is based on search & recognition
  —*10x the data produces 10x the false-positives*
- We develop algorithms that work *better* with more data.

We do science with “real data.”

The Real Data Corpus (70TB)

- Disks, camera cards, & cell phones purchased on the secondary market.
- Most contain data from previous users.
- Mostly acquire outside the US:
  - —*Canada, China, England, Germany, France, India, Israel, Japan, Pakistan, Palestine, etc.*
- Thousands of devices (HDs, CDs, DVDs, flash, etc.)

The problems we encounter obtaining, curating and exploiting this data mirror those of national organizations

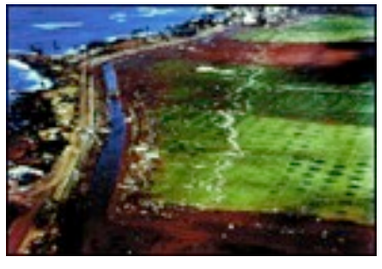
—*Garfinkel, Farrell, Roussev and Dinolt, Bringing Science to Digital Forensics with Standardized Forensic Corpora, DFRWS 2009*
http://digitalcorpora.org/

We manufacture data that can be freely redistributed.

Files from US Government Web Servers (500GB)

- ≈1 million heterogeneous files
  - —*Documents (Word, Excel, PDF, etc.); Images (JPEG, PNG, etc.)*
  - —*Database Files; HTML files; Log files; XML*
- Freely redistributable; Many different file types
- This database was surprising difficulty to collect, curate, and distribute:
  - —*Scale created data collection and management problems.*
  - —*Copyright, Privacy & Provenance issues.*

Advantage over flickr & youtube: persistence & copyright

<abstract>NOAA's National Geophysical Data Center (NGDC) is building high-resolution digital elevation models (DEMs) for select U.S. coastal regions. … </abstract>

<abstract>This data set contains data for birds caught with mistnets and with other means for sampling Avian Influenza (AI)….</abstract>

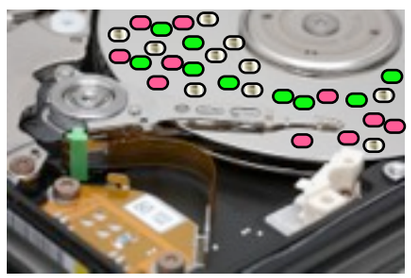
This talk presents today's digital forensic challenges and presents a research project that helps address them.

Introducing digital forensics

✔

Today's digital forensics challenges

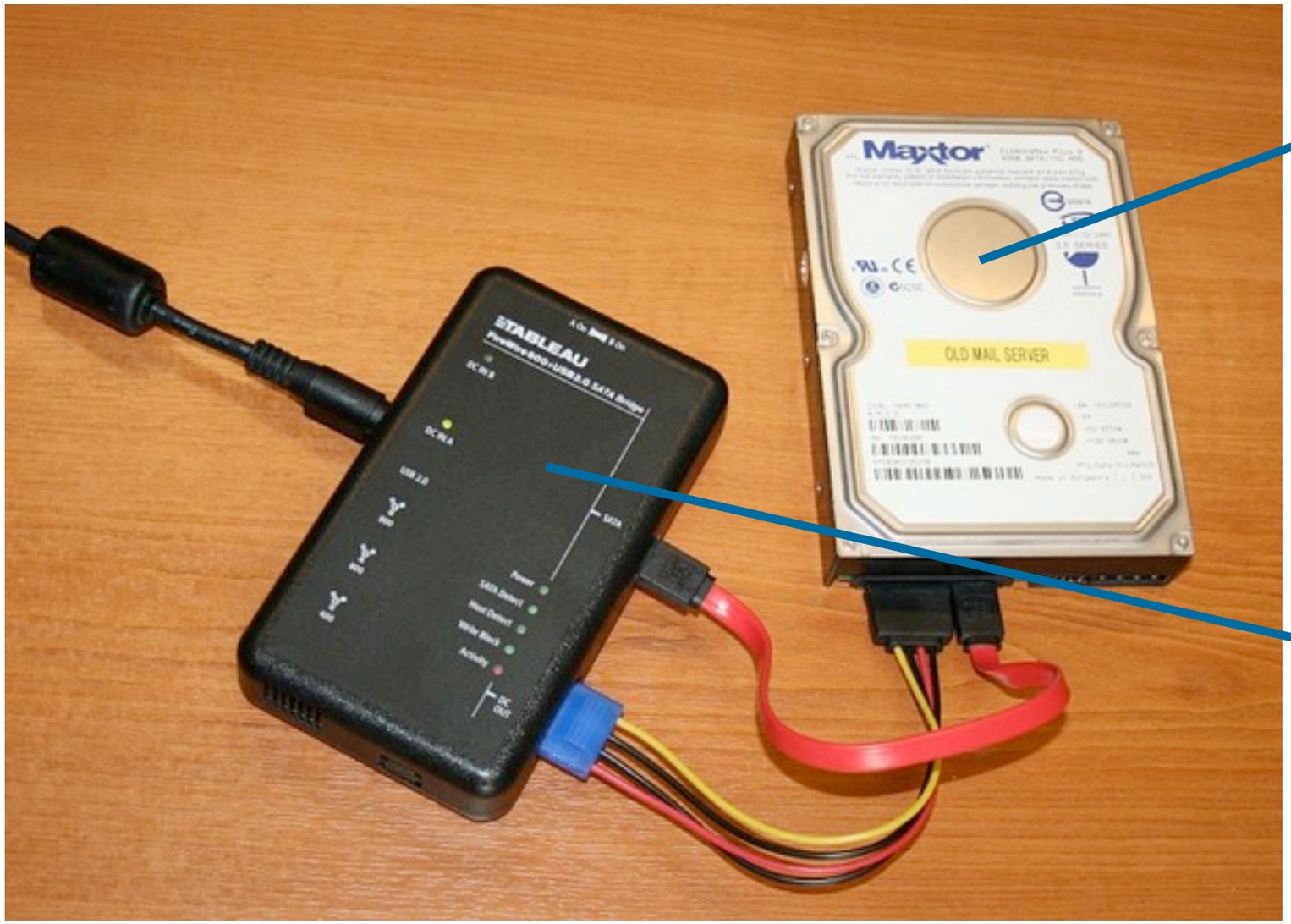
Random sampling for high speed forensics



Challenges Facing Digital Forensics

Extracting digital evidence was simple five years ago

"Imaging tools" extracted data without modification.



"Write Blocker" prevents accidental overwriting.

Original device stored in evidence locker.

Forensic copy ("disk image") stored on a storage array.

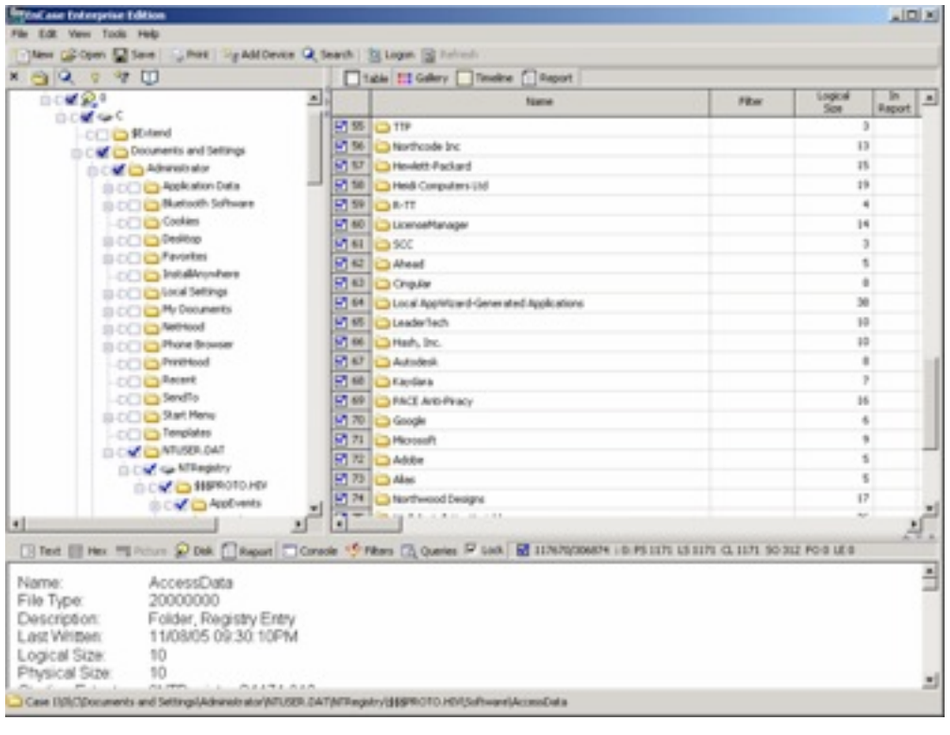
Analyzing digital evidence was simple five years ago

Commercial tools extracted *files* from disk images

- Display of *allocated* & *deleted* files.
- String search
- File extraction
- File “carving”
- Examining disk sectors

Job of analyst:

- Find interesting data
- Report it



Today much of the work is with cell phones. Every one is different.

Operating system:

- Android? iPhone? Blackberry? Feature Phone?

Access to the data:

- PIN lock?
- Encrypted Storage?
- Stored locally or in the cloud?

Applications:

- Built-in? Downloaded from “App Store”?
- Custom-written?
- Self-destruct / remote wipe?
- Malware?

Human Language: English? Korean? Chinese?

Digital forensics is fundamentally different from other kinds of scientific exploration...

There are five key challenges that we face...

Diversity is the fundamental challenge of DF

DF must analyze any OS, application, protocol, encryption, etc...








"Analyze any data that might be found on any computer."

Diversity of devices, diversity over time

Today's DF tools must process:

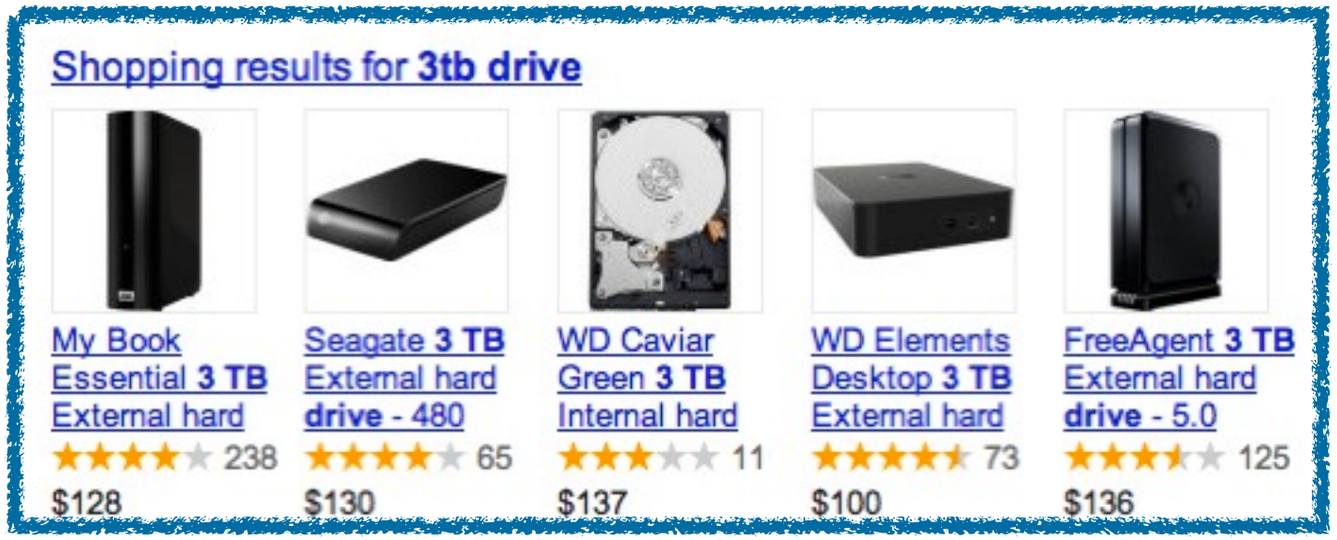
- Today's computers / phones / cameras
  - —*Because some criminals like to buy what's new!*
- Yesterday's computers / phones / cameras
  - —*Because criminals are using old devices too!*

Implications for DF users and developers:

- Upgrade DF software as soon as possible.
- DF software will become geometrically more complicated over time....
  - —*... or DF software will adapt on the fly to new data formats and representations.*
  - —*automated code analysis; pattern matching; hidden Markov models; etc.*

Data scale is a never-ending problem

Every year we have more data to analyze



Moore's law helps the adversary as much as us!

- We are using top-of-the-line system to analyze top-of-the-line systems
- We need to analyze in days what a subject spent weeks, months or years assembling

—*We will never outpace the performance curve.*

We must adopt "big data solutions"

Human capital is a challenge — especially in DF

Users (examiners, analysts):

- Overwhelmingly in law enforcement.
- Little or no background in CS or IS
- Deadline-driven; over-worked
- Knowledgable users tend to focus in just one particular area.

  —*Result: It takes two years to train most DF examiners.*

Researchers and Developers:

- Data diversity means developers need to know the whole stack

  —*opcodes & Unicode ⇒ OS & Apps ⇒ networking, encryption, etc.*

- Scale issues means developers need to know HPC:

  —*threading, systems engineering, supercomputing, etc.*

- Result:

  —*It's hard to find qualified developers*

  —*Developers must be generalists*

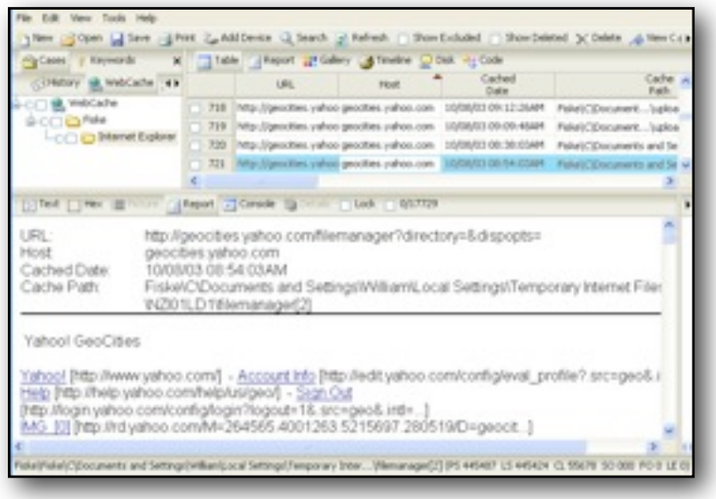
The “CSI Effect” creates unrealistic expectations.

TV digital forensics:

- Every investigator is trained on every tool.
- Correlation is easy and instantaneous.
- There are no false positives.
- Overwritten data can be recovered.
- Encrypted data can usually be cracked.
- It is impossible to delete anything.

The reality:

- Overwritten data *cannot* be recovered
- Encrypted data usually can't be decrypted
- Forensics rarely answers questions or establishes guilt
- Tools crash a lot



EnCase

Result:

—DF is a difficult process that looks easy

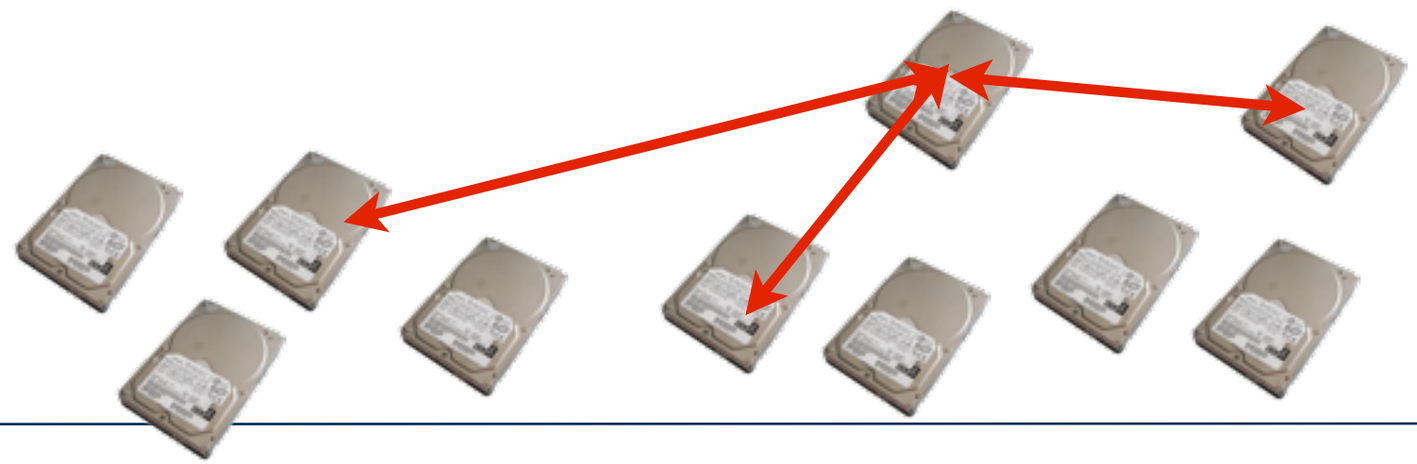
DF must respond with new science.

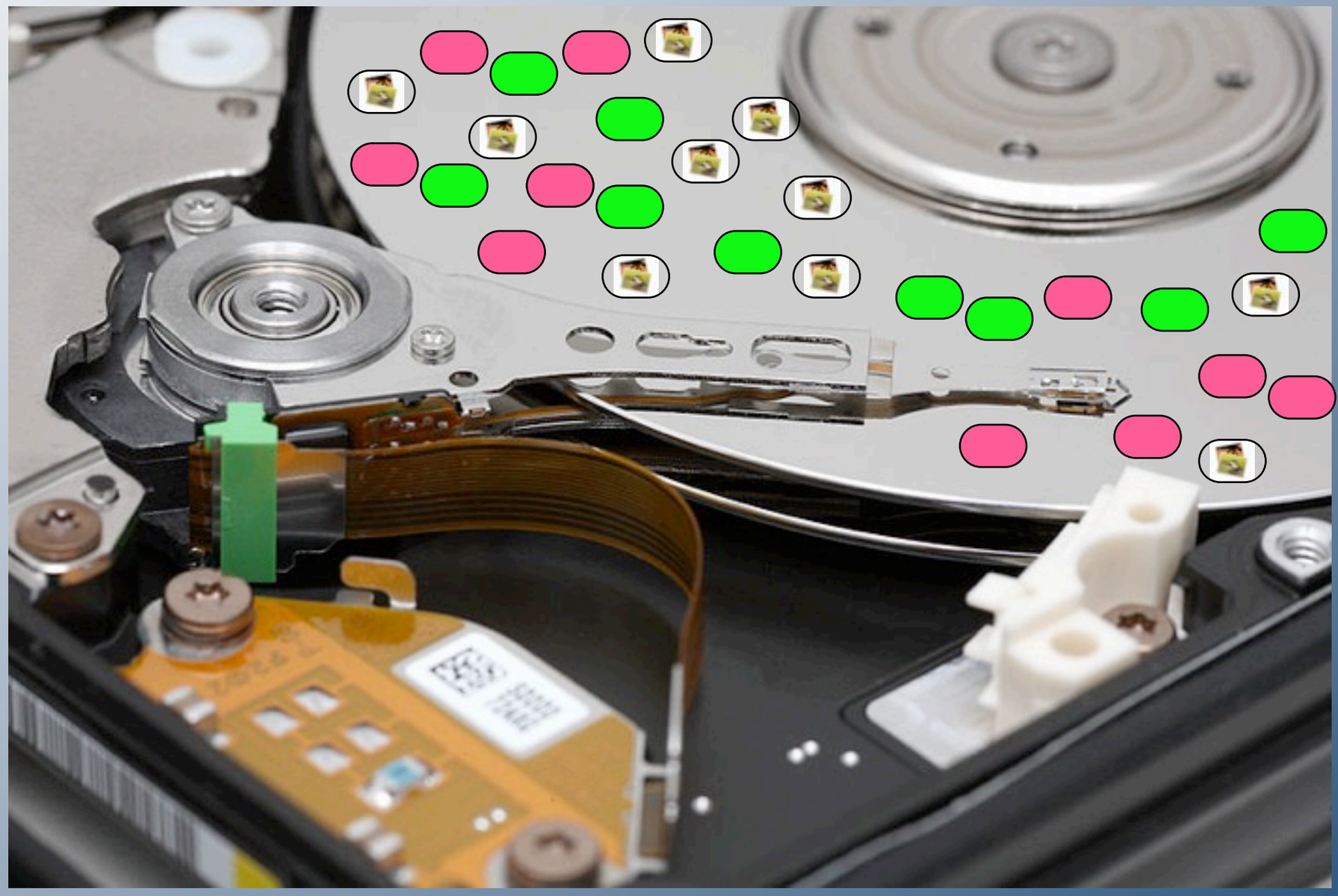
Current approaches don't scale.

- User spent *years* assembling email, documents, etc
- Analysts have days or hours to process it
- Police analyze top-of-the-line systems
  —*with top-of-the-line systems*
- National Labs have large-scale server farms
  —*to analyze huge collections*

Our new algorithms must leverage our advantage: massive data

- Outlier detection and correlation
- Operate autonomously on incomplete, heterogeneous datasets
- Automatically calibrate; have no false positives





High speed forensic analysis with random sampling

Traditionally forensic analysis was leisurely.
Today much analysis is under time pressure.

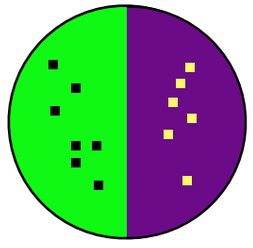
US agents encounter hard drives at border crossings...

US agents might have a need to search a room of computers:

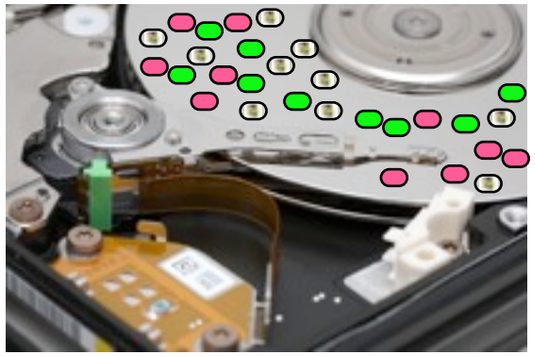
What can we learn about a 1TB drive in five minutes?

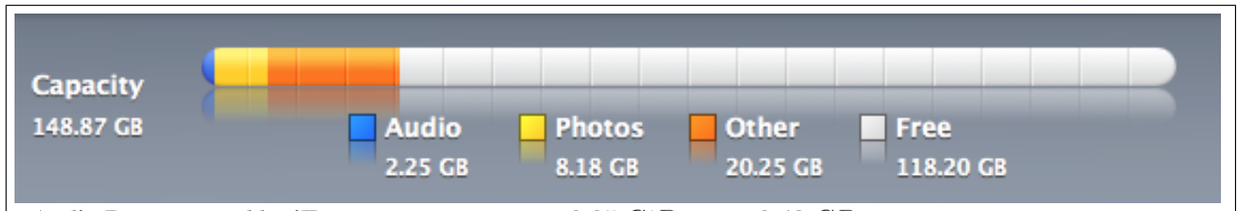
Random sampling is a powerful tool for analyzing data

Simple random sampling can determine % free space



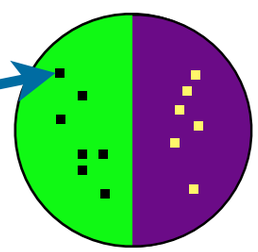
Data characterization can determine the *kind* of stored data





Audio Data reported by iTunes:	2.25 GiB	2.42 GB	
MP3 files reported by file system:		2.39 GB	
Estimated MP3 usage with random sampling :		2.49 GB	10,000 random samples
		2.71 GB	5,000 random samples

Sector hashing can identify *specific target files*



It takes 3.5 hours to read a 1TB hard drive.

In 5 minutes you can read:

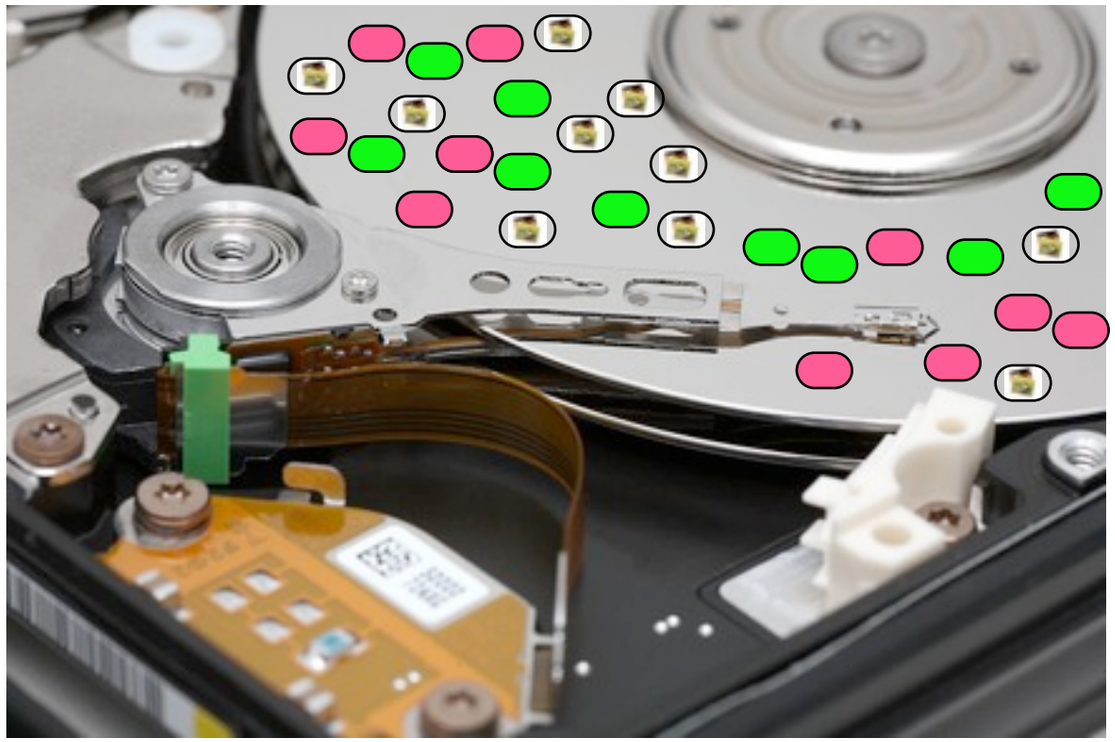
- 36 GB in one strip
- 100,000 randomly chosen 64KiB strips (assuming 3 msec/seek)

			
Minutes	208	5	5
Data	1 TB	36 GB	6.5 GB
# Seeks	1	1	100,000
% of data	100%	3.6%	0.65%

The statistics of a *randomly chosen sample* predict the *statistics of a population.*

US elections can be predicted by sampling thousands of households:

Hard drive contents can be predicted by sampling thousands of sectors:

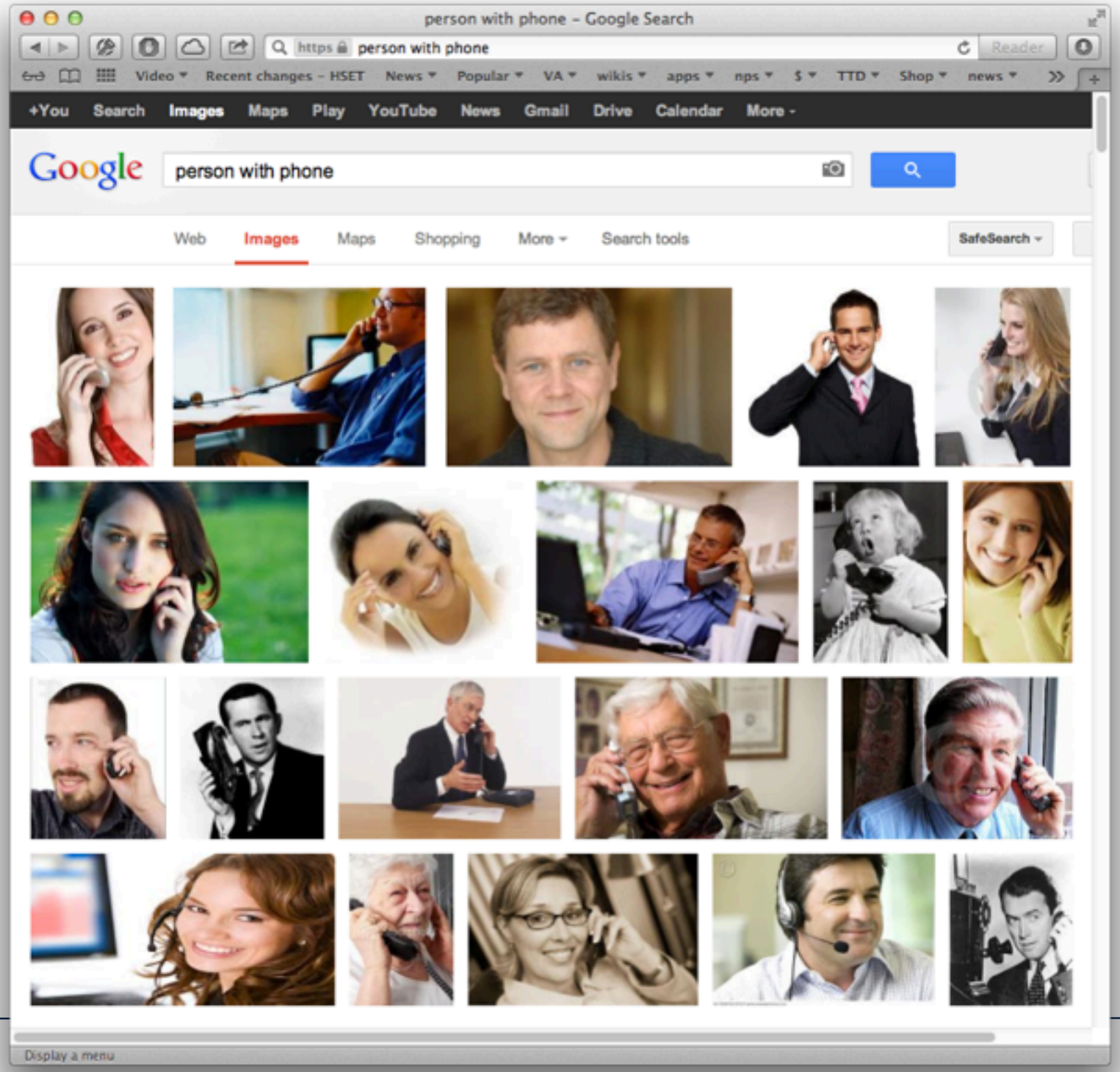


The challenge is identifying *likely voters.*

The challenge is *identifying the sector* content that is sampled.

Challenge for political polls: interpreting each phone call

"On Tuesday, how will you vote for governor?"



Challenge for forensic sampling: interpreting each sector

"What data do you have?"

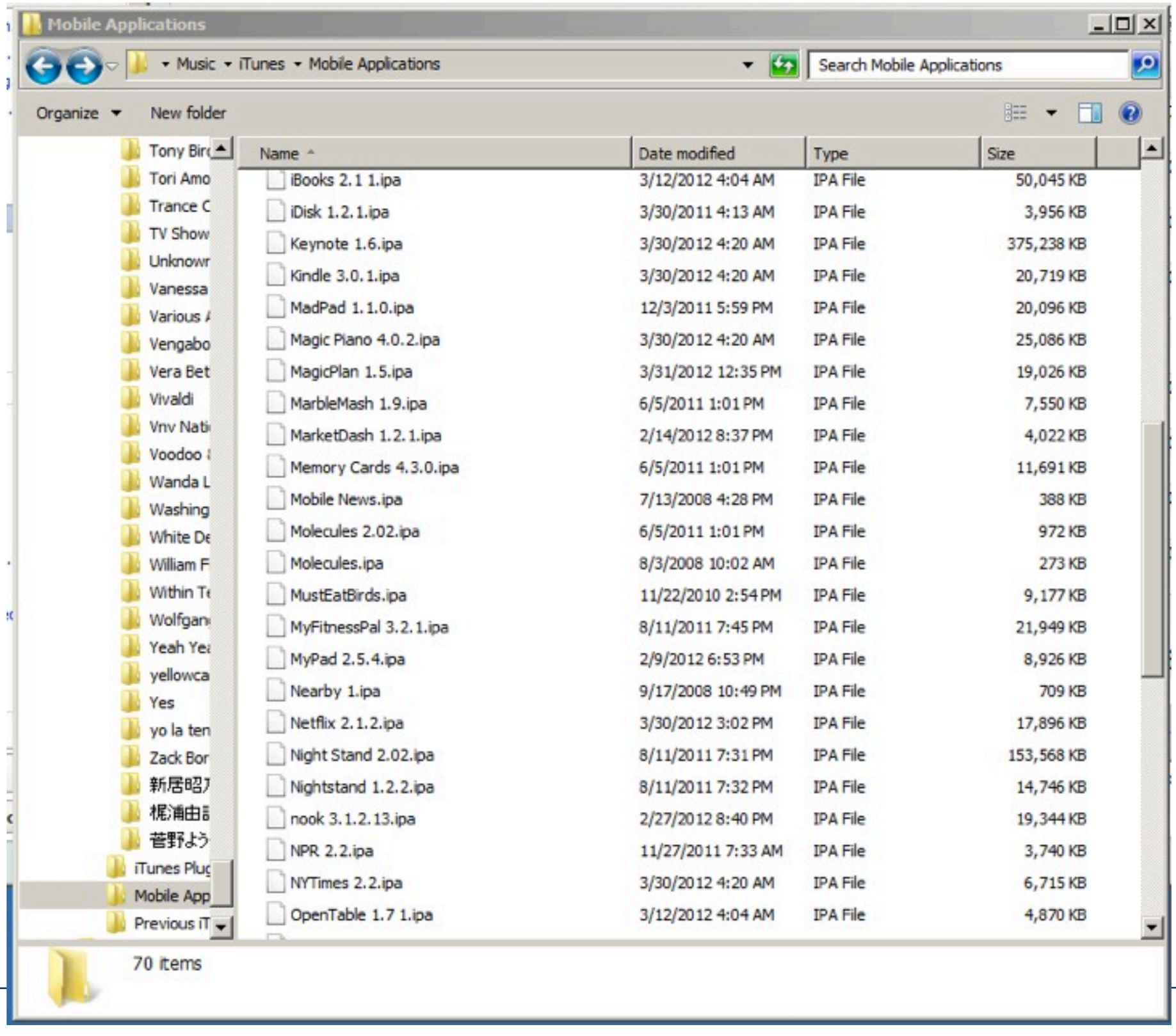
- Easy:

```
0000000: ffd8 ffe0 0010 4a46 4946 0001 0201 0048  ......JFIF.....H
0000010: 0048 0000 ffe1 1d17 4578 6966 0000 4d4d  .H......Exif..MM
0000020: 002a 0000 0008 0007 0112 0003 0000 0001  .*..............
0000030: 0001 0000 011a 0005 0000 0001 0000 0062  ...............b
0000040: 011b 0005 0000 0001 0000 006a 0128 0003  ...........j.(..
0000050: 0000 0001 0002 0000 0131 0002 0000 001b  .........1......
0000060: 0000 0072 0132 0002 0000 0014 0000 008d  ...r.2..........
0000070: 8769 0004 0000 0001 0000 00a4 0000 00d0  .i..............
0000080: 0000 0048 0000 0001 0000 0048 0000 0001  ...H.......H....
0000090: 4164 6f62 6520 5068 6f74 6f73 686f 7020  Adobe Photoshop
00000a0: 4353 2057 696e 646f 7773 0032 3030 353a  CS Windows.2005:
00000b0: 3035 3a30 3920 3136 3a30 313a 3432 0000  05:09 16:01:42..
00000c0: 0000 0003 a001 0003 0000 0001 0001 0000  ................
00000d0: a002 0004 0000 0001 0000 00c8 a003 0004  ................
00000e0: 0000 0001 0000 0084 0000 0000 0000 0006  ................
00000f0: 0103 0003 0000 0001 0006 0000 011a 0005  ................
```

- Hard:

```
000a000: 0011 fa71 57f4 6f5f ddff 00bd 15fb 5dfd  ...qW.o_......].
000a010: a996 0fc9 dff1 ff00 b149 e154 97f4 efd5  .........I.T....
000a020: e3f5 7f47 71df 8ffb d5d7 da9e d87f c12f  ...Gq........../
000a030: f8ff 00d8 b1f4 b1f8 ff00 c57e ab7a ff00  ..........~.z..
```

We think of computers as devices with *files*.



Data on computers is stored in fixed-sized sectors.

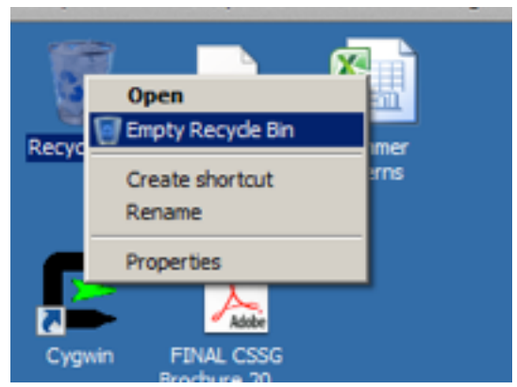
Data in a sector can be resident:

Resident Data

user files
email messages
[temporary files]

Files can be “deleted” but the data remains:

Deleted Data

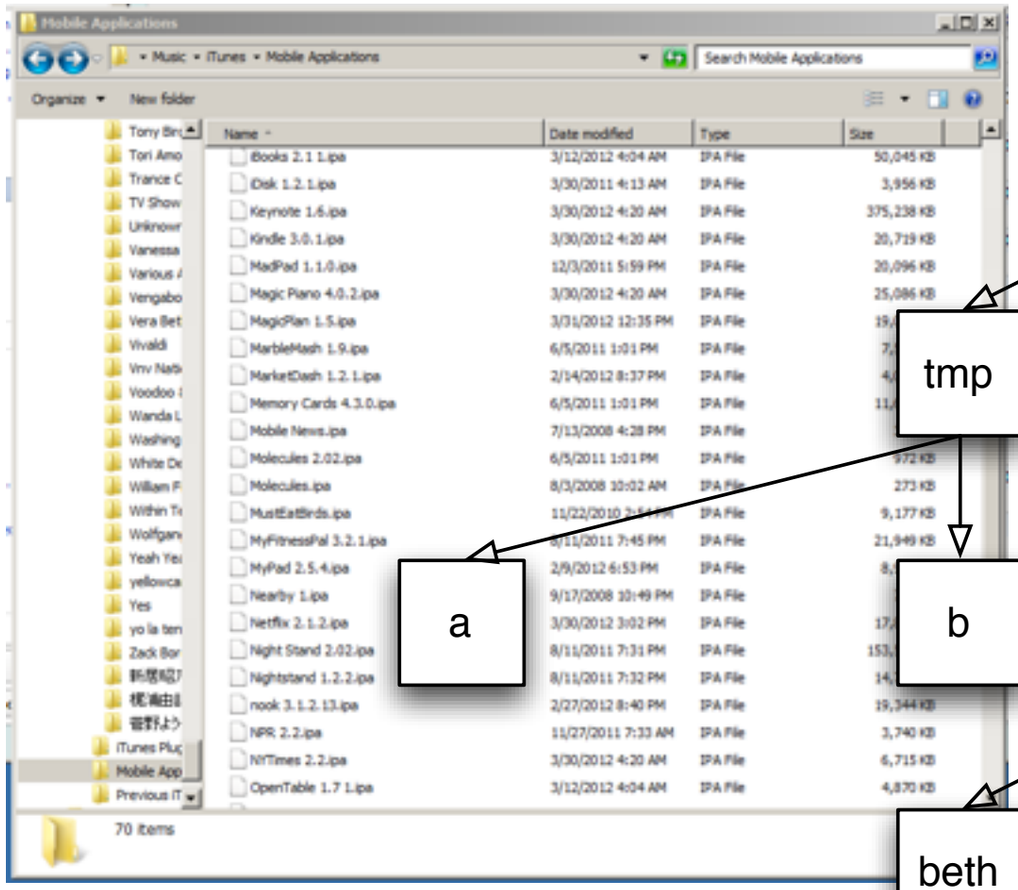


Sectors can be wiped clean:

No Data

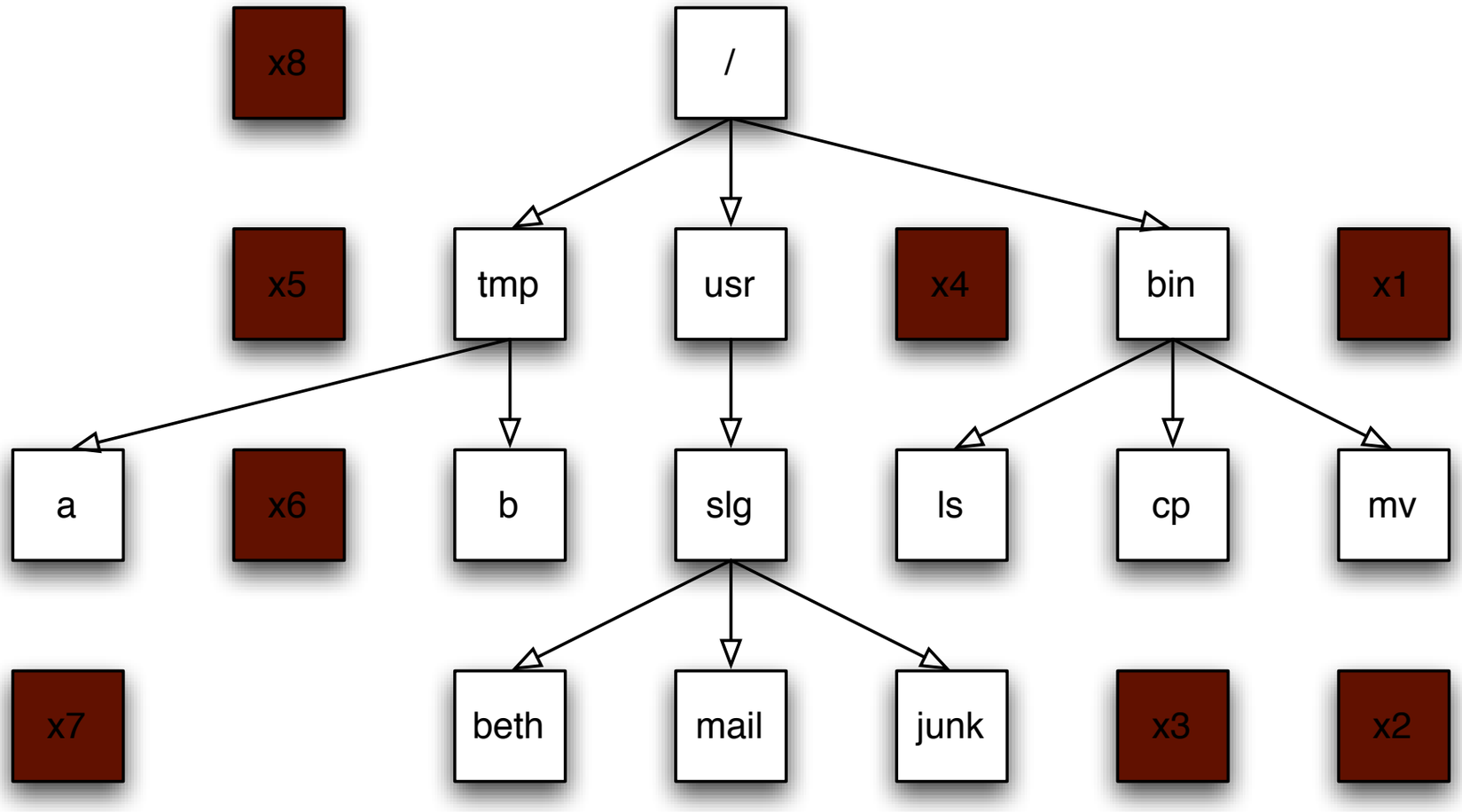
blank sectors

Resident data is the data you see from the root directory. e.g. “allocated” files.



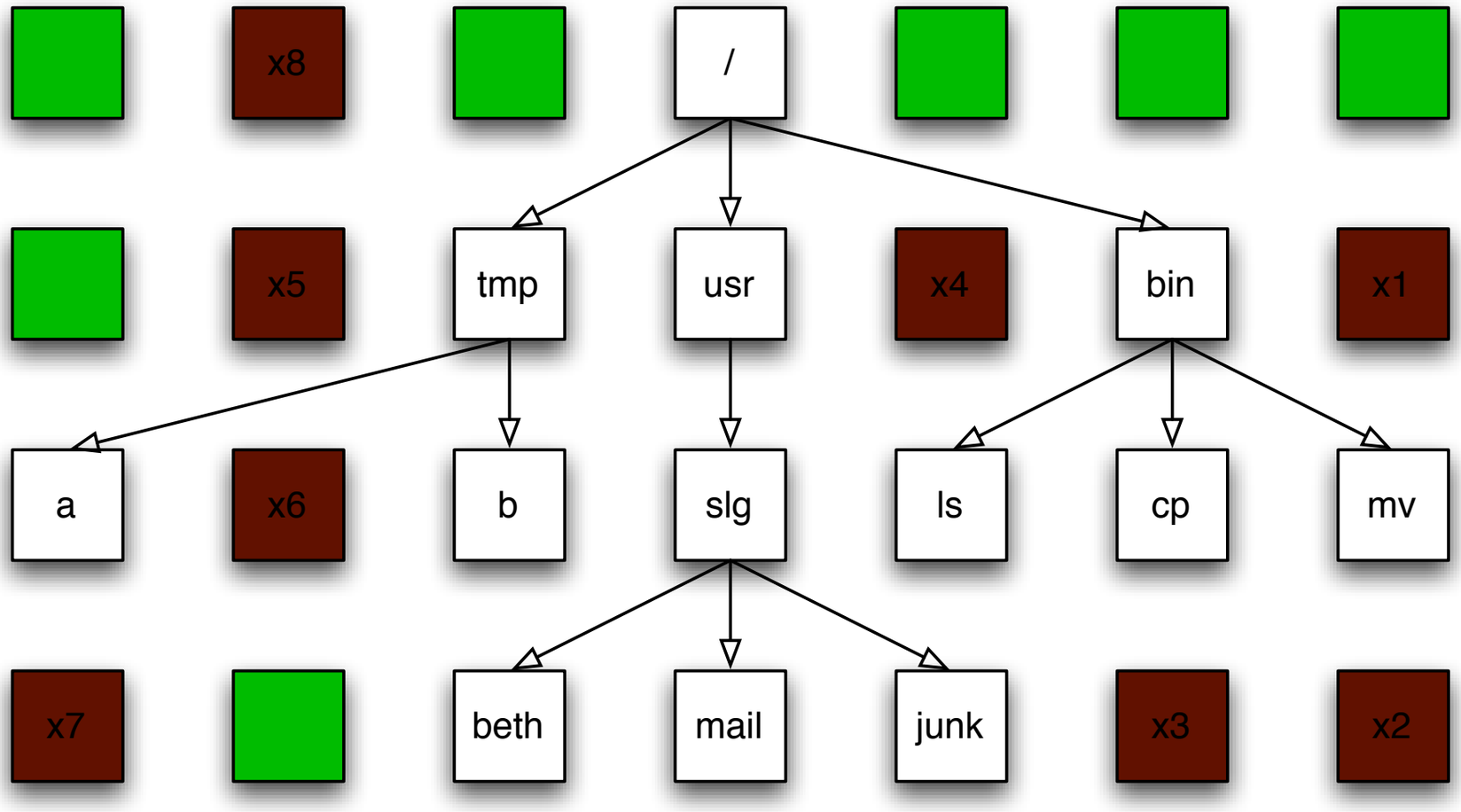
Resident Data

“Deleted data” is on the disk, but can only be recovered with forensic tools.



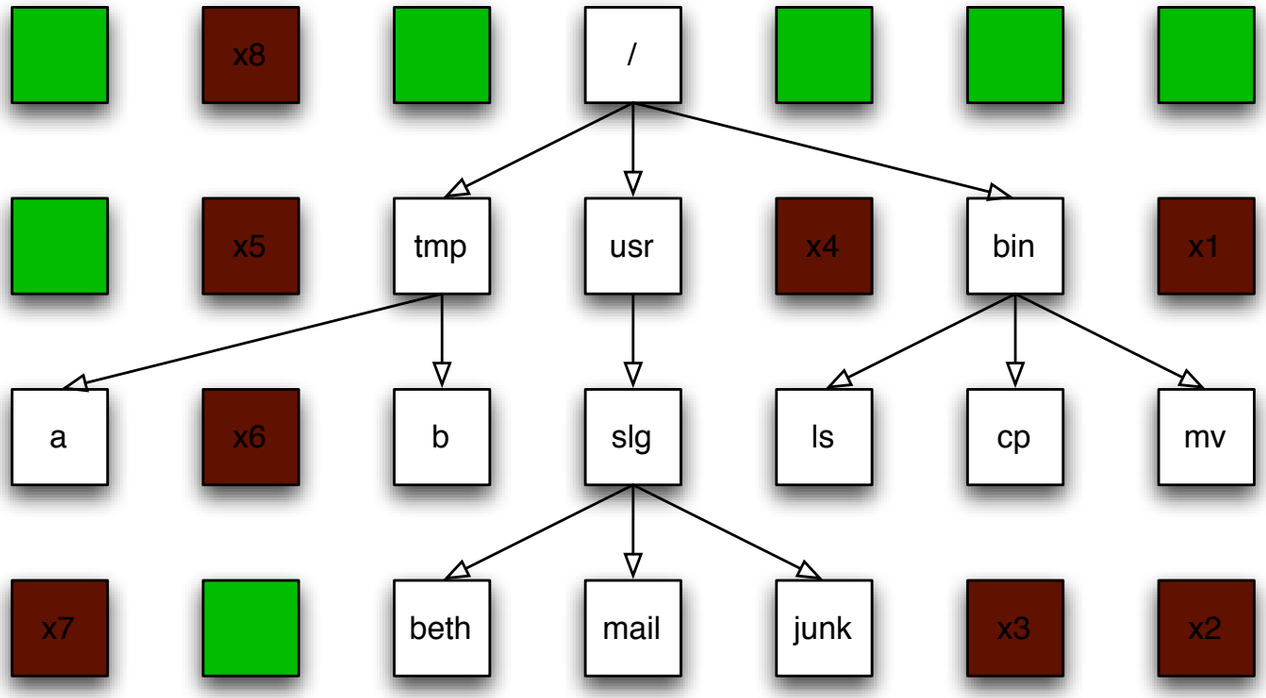
Deleted Data

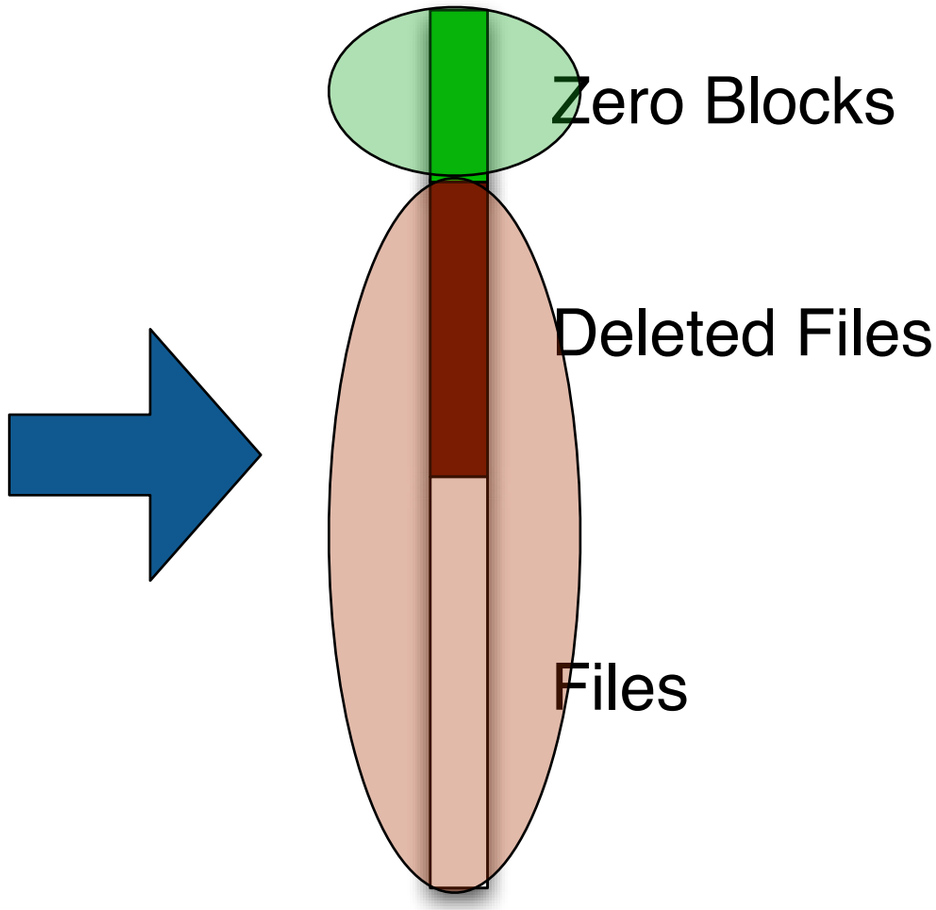
Some sectors are blank.
They have “No data.”



No Data

Sampling can't distinguish *allocated* from *deleted* data.



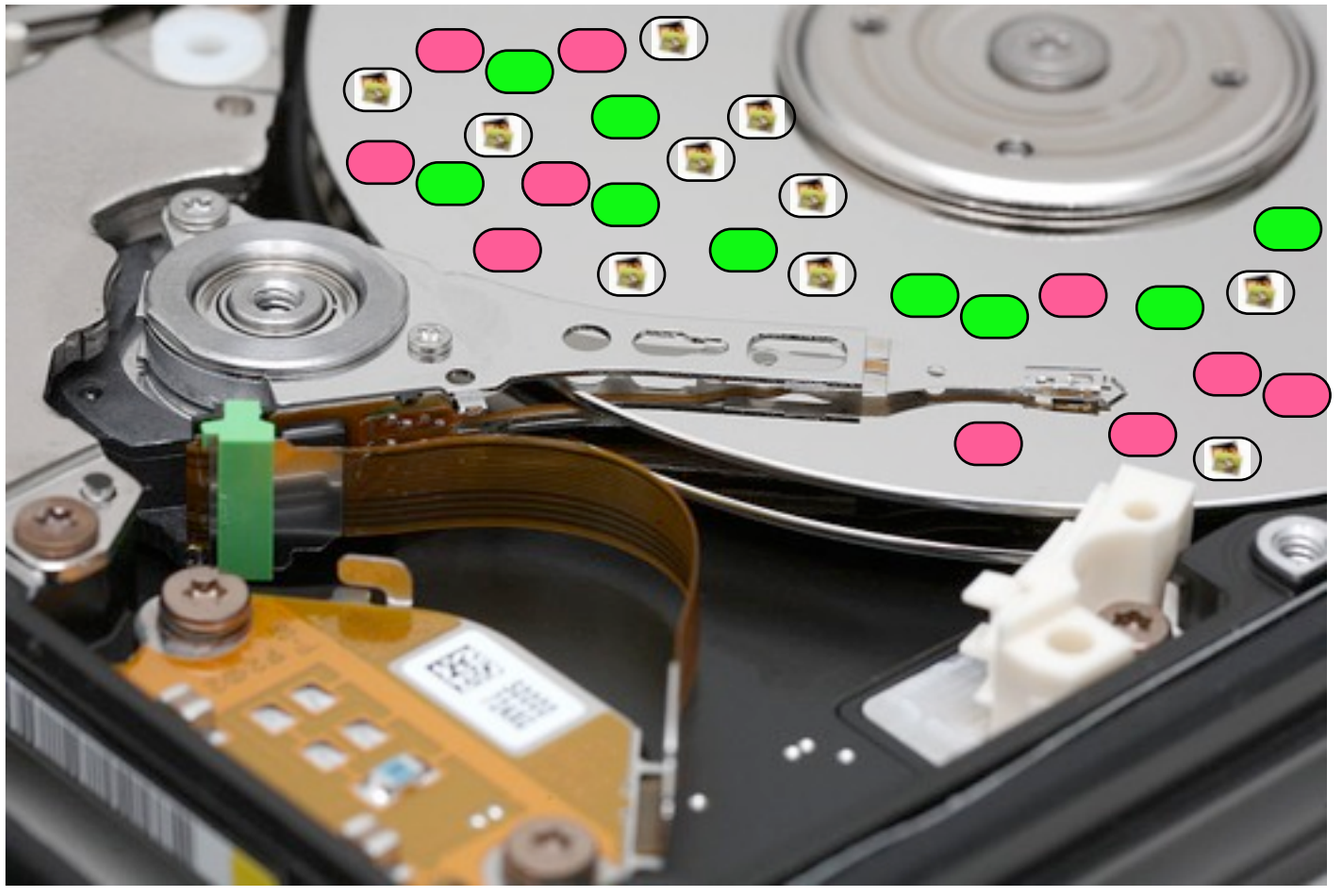


Sampling can tell us about the content of the data

Sampling can tell us the proportion of...

—*blank sectors; video; HTML files; other data types...*

—*data with distinct signatures...*

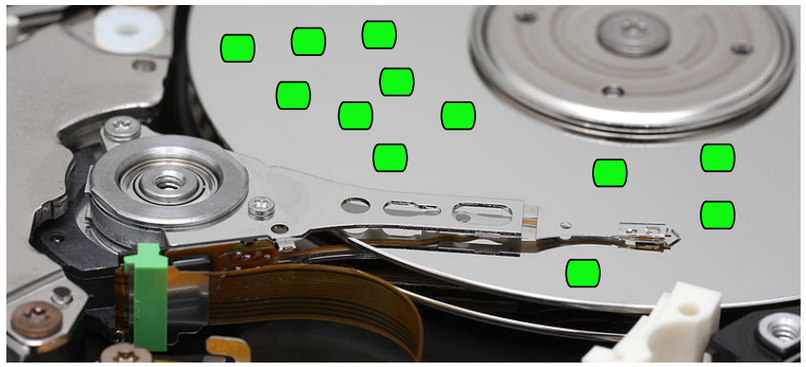


...provided we can identify it

Simplify the problem.
Can we use statistical sampling to verify wiping?

Many organizations discard used computers.

Can we verify if a disk is properly wiped in 5 minutes?



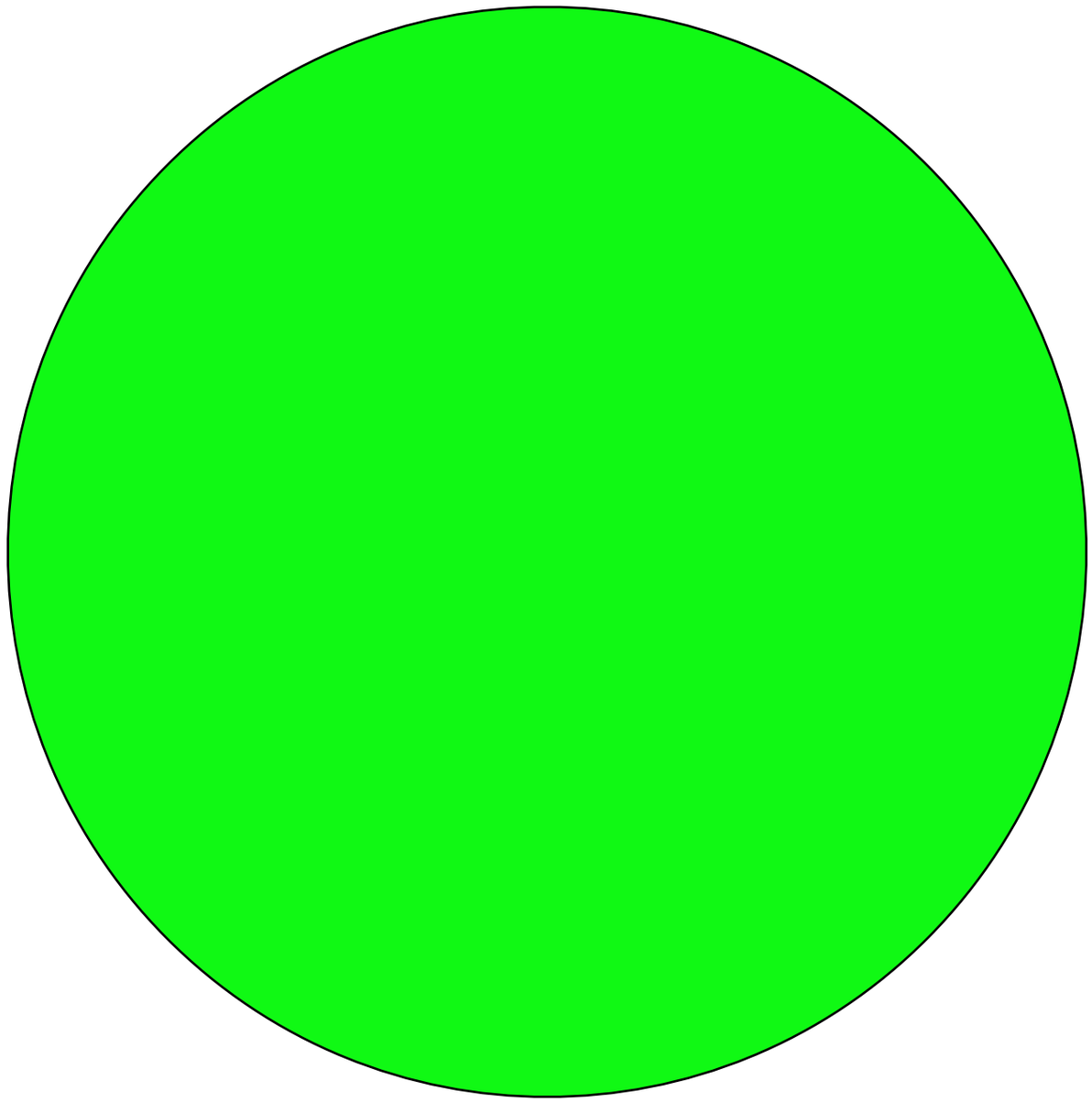
Simple solution:

- 1. Read a random sector

  —*If there is data, the drive is not wiped.*

- 2. Repeat until satisfied.

A 1TB drive has 2 billion sectors.
What if we read 10,000 and they are all blank?

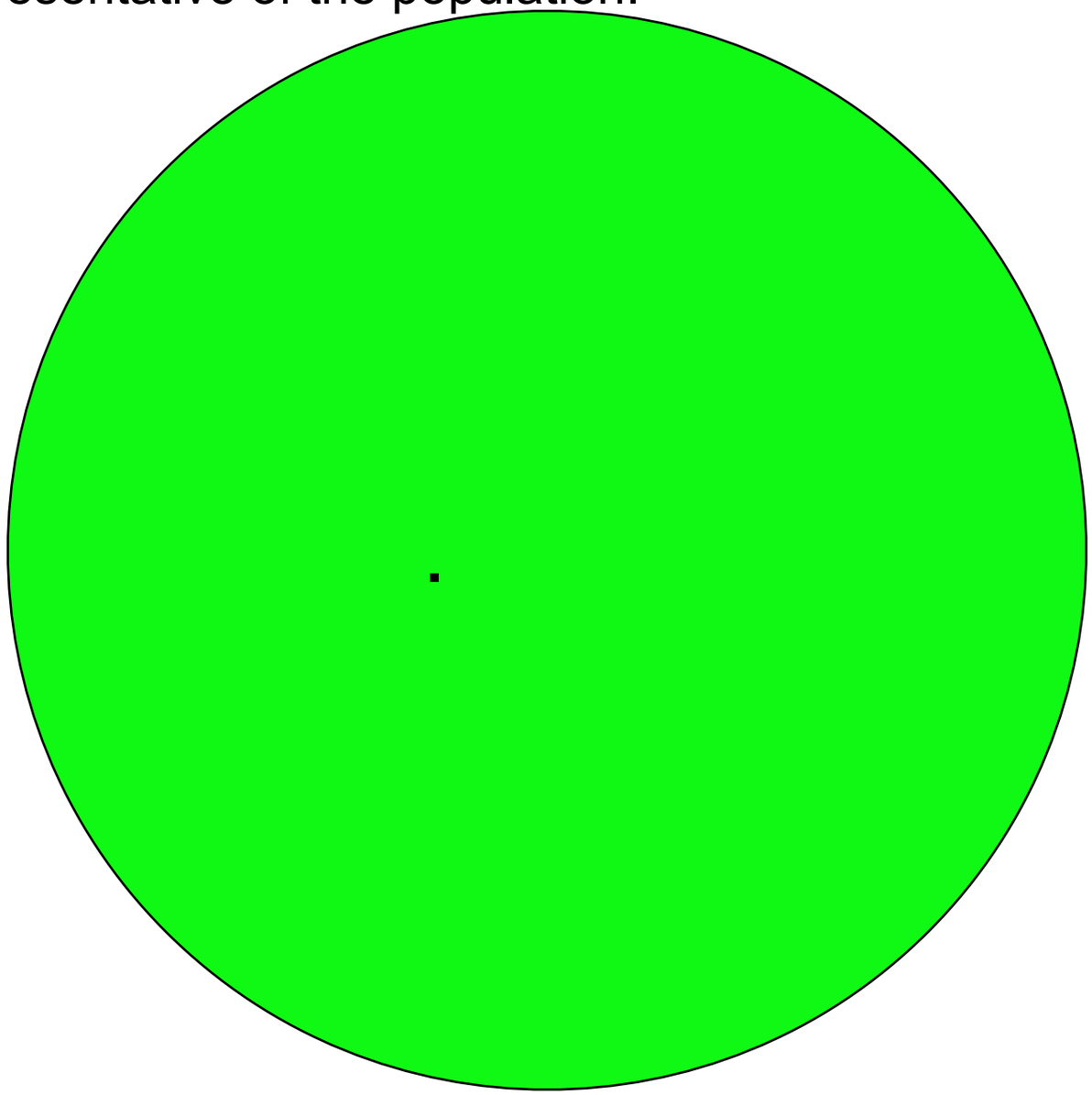


Chances are good that they are all blank.

Random sampling *won't* find a single written sector.

If the disk has 1,999,999,999 blank sectors (1 with data)

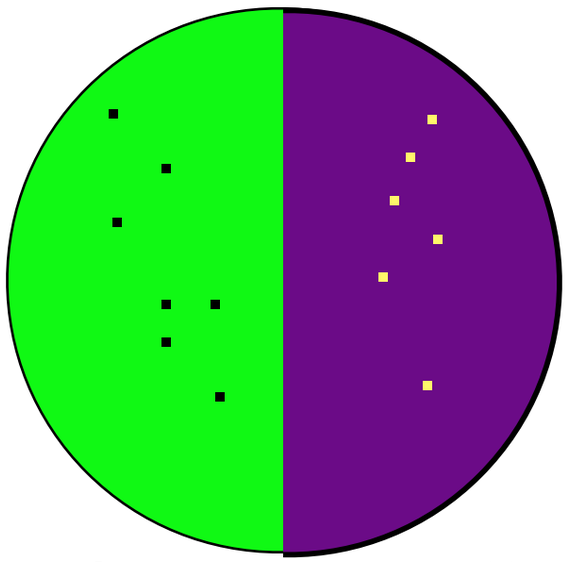
- The sample is representative of the population.



We will only find that 1 sector with exhaustive search.

If half of the sectors are blank...

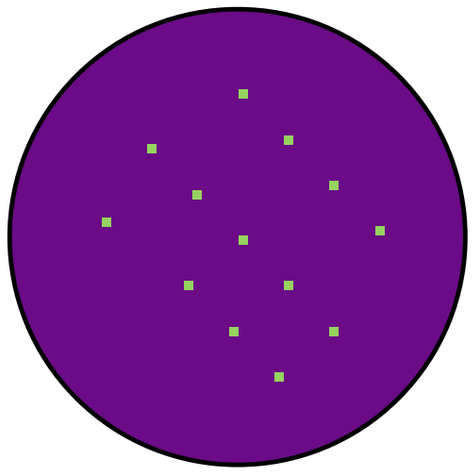
Sectors	Blank	Data
Sampled	5,000 (50%)	5,000 (50%)
Total:	1,000,000,000 (50%)	1,000,000,000 (50%)



The distribution of the data *does not matter* if sampling is random.

What if the the sampled sectors *are the only blank sectors?*

Sectors	Blank	Data
Sampled	10,000 (100%)	0 (0%)
Total:	10,000 (0.0005%)	1,999,990,000 (99%)



If the the only sectors read are blank...

—*We are incredibly unlucky.*

—*Somebody has hacked our random number generator!*

This is an example of the "urn" problem from statistics

Assume a 1TB disk has 10MB of data.

- 1TB = 2,000,000,000 = 2 Billion 512-byte sectors!
- 10MB = 20,000 sectors

Read just 1 sector; the odds that it is blank are:

$$\frac{2,000,000,000 - 20,000}{2,000,000,000} = .99999$$

Read 2 sectors. The odds that both are blank are:

$$\left(\frac{2,000,000,000 - 20,000}{2,000,000,000}\right)\left(\frac{1,999,999,999 - 20,000}{2,000,000,000}\right) = .99998$$

first pick **second pick**

Odds we may have missed something

The more sectors picked, the less likely we are to miss the data….

$$P(X = 0) = \prod_{i=1}^{n} \frac{((N - (i-1)) - M)}{(N - (i-1))} \qquad (5)$$

Sampled sectors	Probability of not finding data
1	0.99999
100	0.99900
1000	0.99005
10,000	0.90484
100,000	0.36787
200,000	0.13532
300,000	0.04978
400,000	0.01831
500,000	0.00673

Table 1: Probability of not finding any of 10MB of data on a 1TB hard drive for a given number of randomly sampled sectors. Smaller probabilities indicate higher accuracy.

Non-null data		Probability of not finding data
Sectors	Bytes	with 10,000 sampled sectors
20,000	10 MB	0.90484
100,000	50 MB	0.60652
200,000	100 MB	0.36786
300,000	150 MB	0.22310
400,000	200 MB	0.13531
500,000	250 MB	0.08206
600,000	300 MB	0.04976
700,000	350 MB	0.03018
1,000,000	500 MB	0.00673

Table 2: Probability of not finding various amounts of data when sampling 10,000 disk sectors randomly. Smaller probabilities indicate higher accuracy.

—Pick 500,000 random sectors

—If are all NULL, the disk has p=(1-.00673) chance of having 10MB of non-NULL data

—The disk has a 99.3% chance of having less than 10MB of data

In practice, we use a modified algorithm...

Sample with 64KiB “blocks” instead of 512-byte sectors.

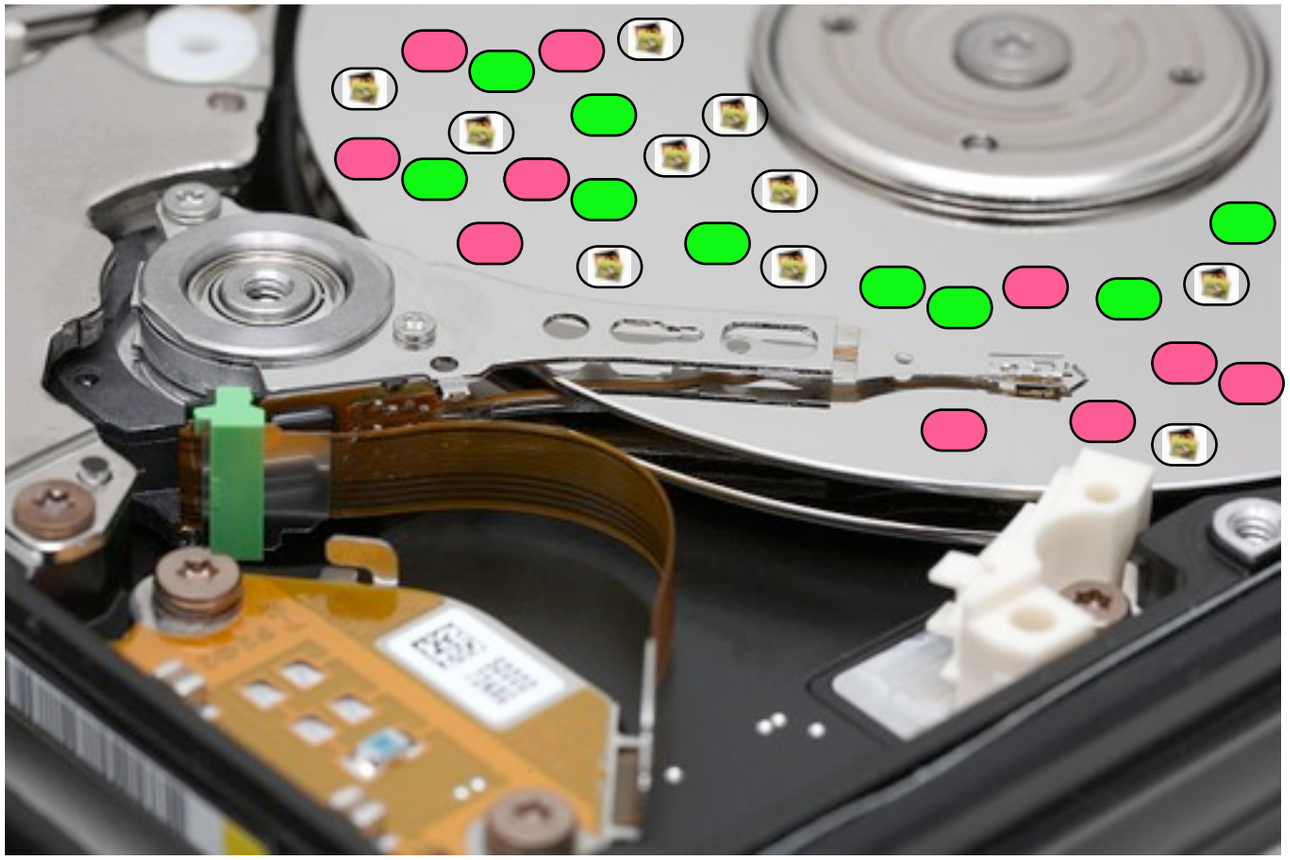
- It takes the same amount of time to read 65,536 bytes as 512 bytes
- Analyze 64KiB block with a 4KiB sliding window
- On a 1TB drive, there are 15,258,789 64KiB sections

Identify data “type”

- Blank
- JPEG
- Video
- Encrypted

Update results in real-time

- Provides immediate feedback
- Catches important data faster
- Stop when analyst is satisfied.

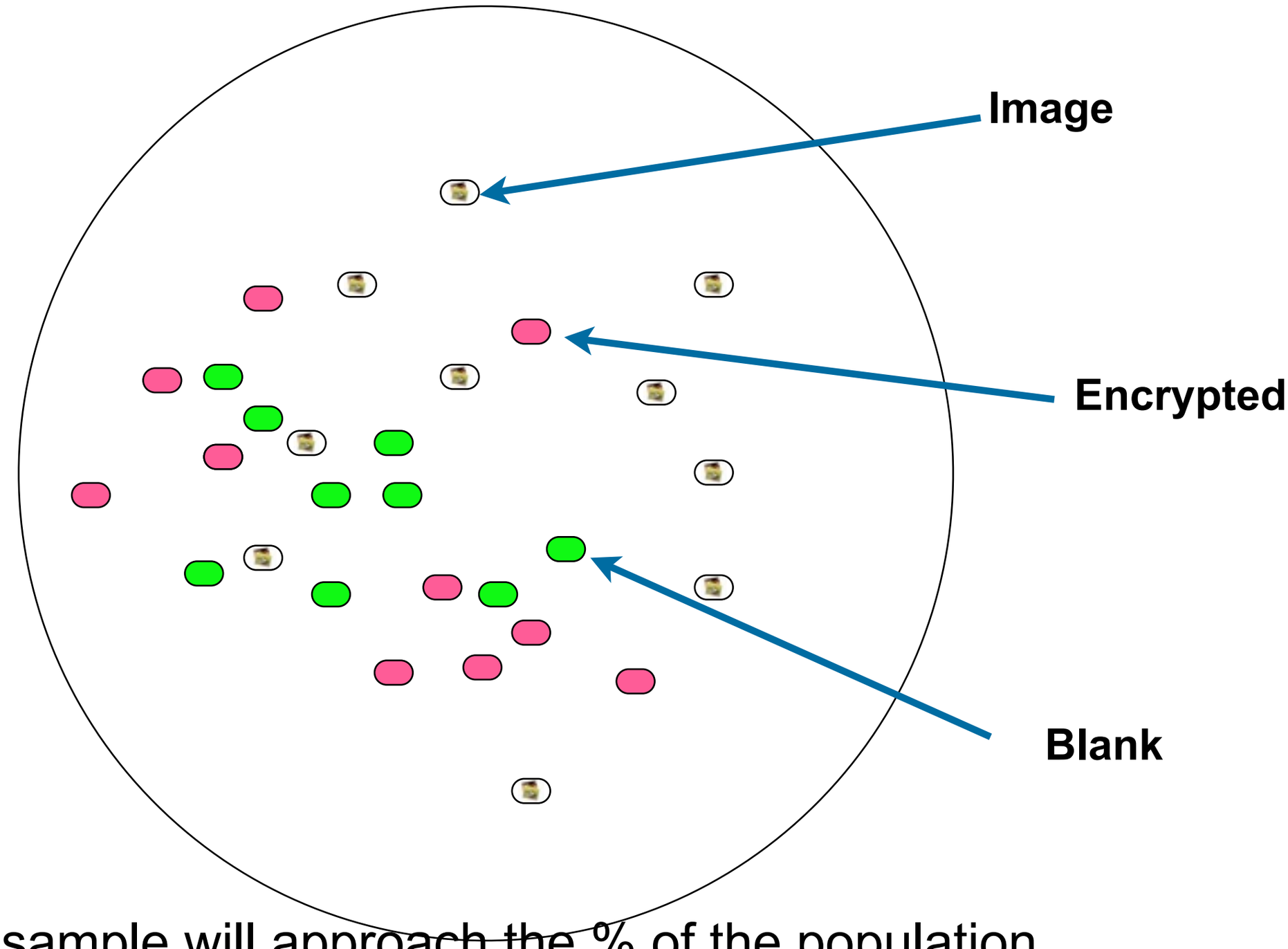


We used this technique to calculate the size of the TrueCrypt volume on this iPod.

It takes 3+ hours to read all the data on a 160GB iPod.

- Apple bought very slow hard drives.

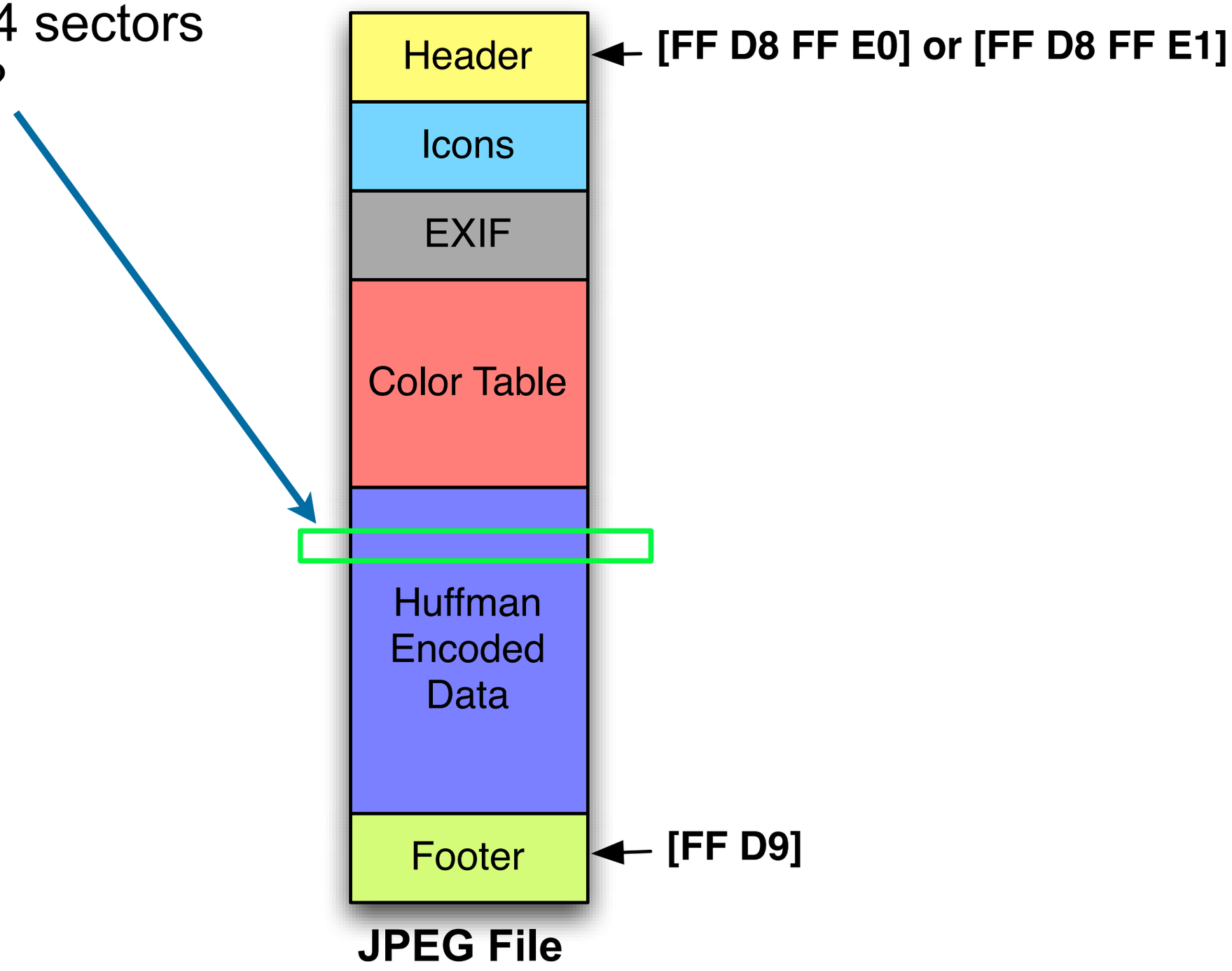
We got a statistically significant sample in two minutes.



The % of the sample will approach the % of the population.

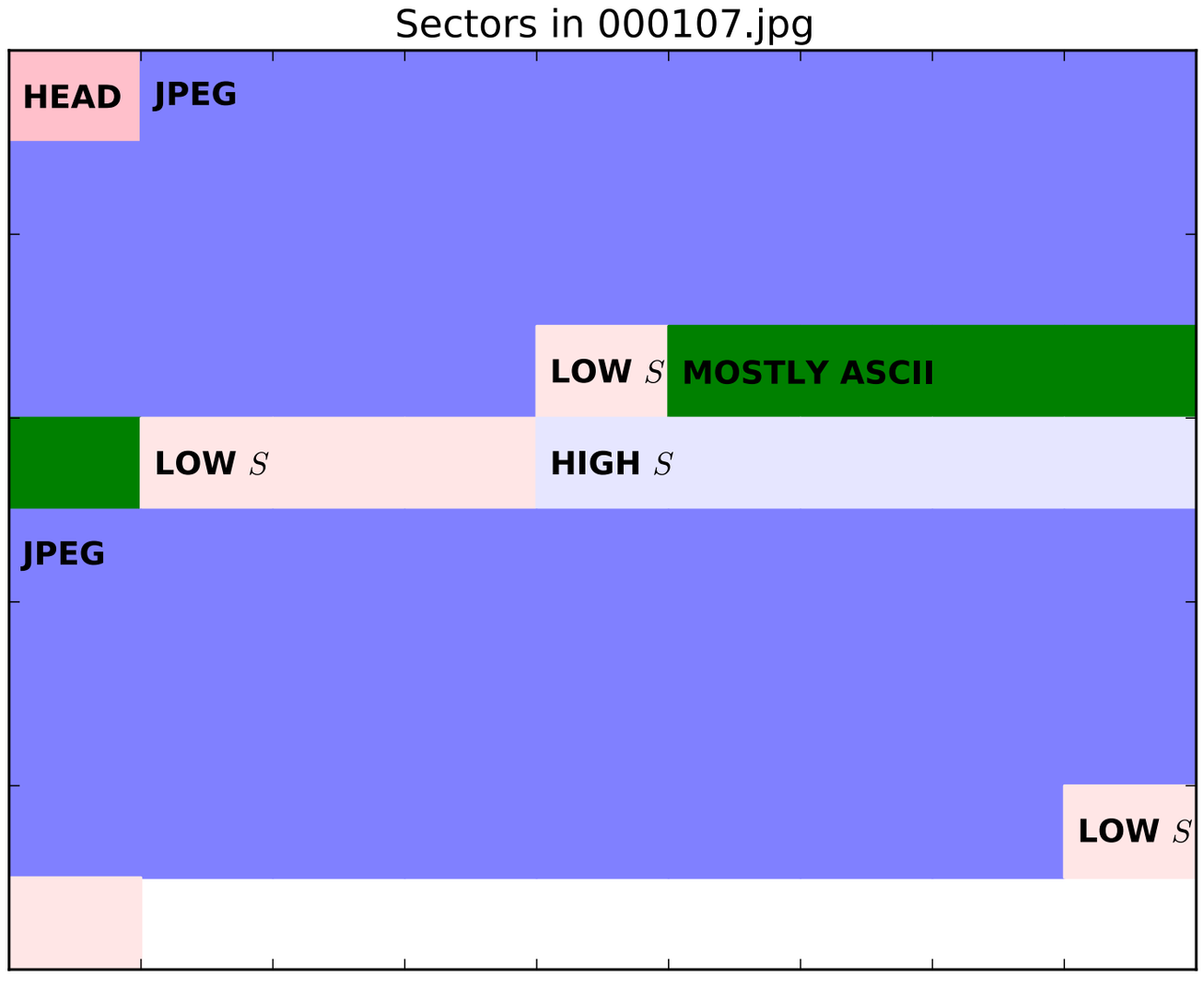
The challenge: identifying a file “type” from a fragment.

Can you identify a JPEG file from reading 4 sectors in the middle?



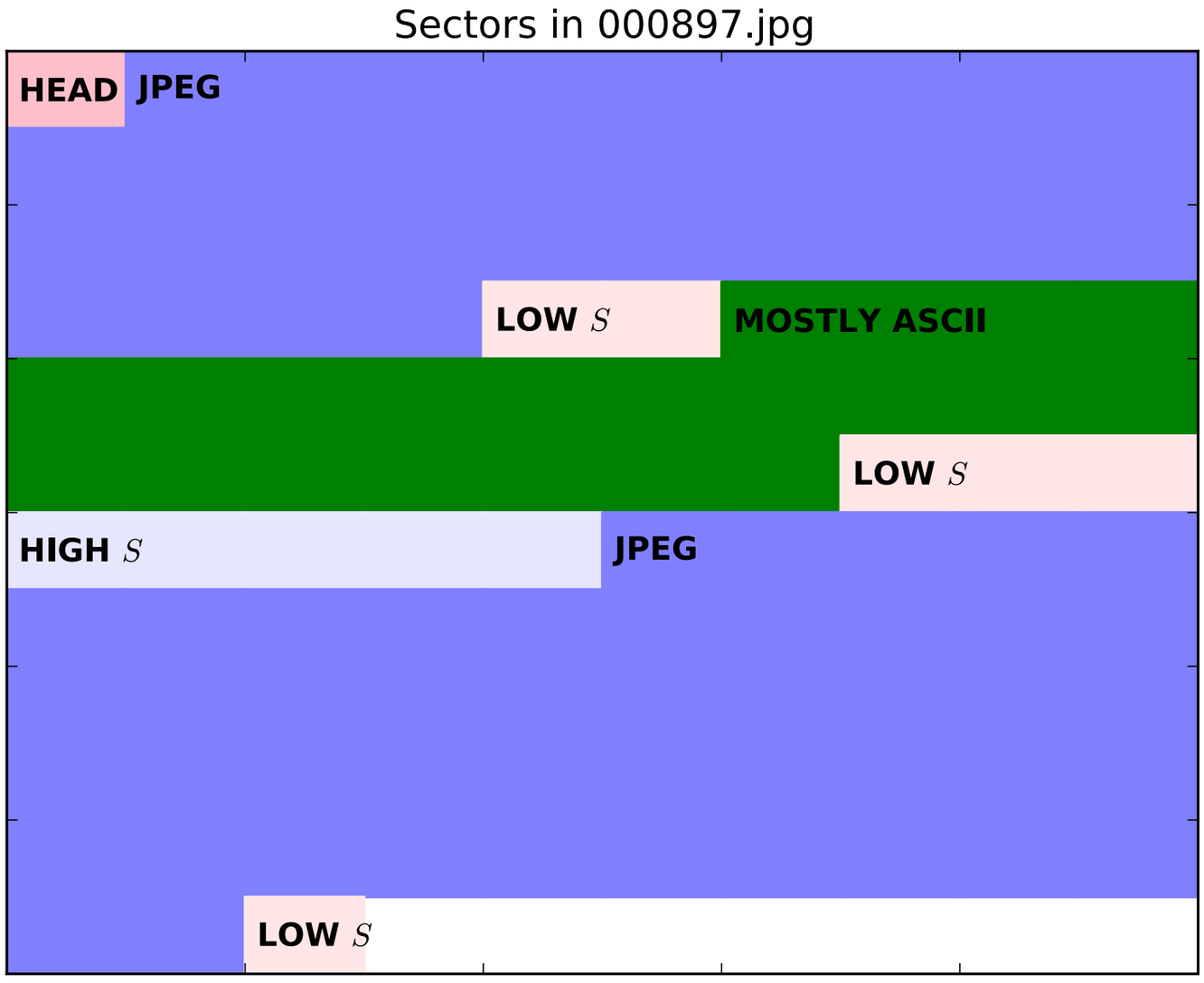
We built detectors to recognize the different parts of a JPEG file.

000107.jpg
Bytes: 41,572



Sectors: 82

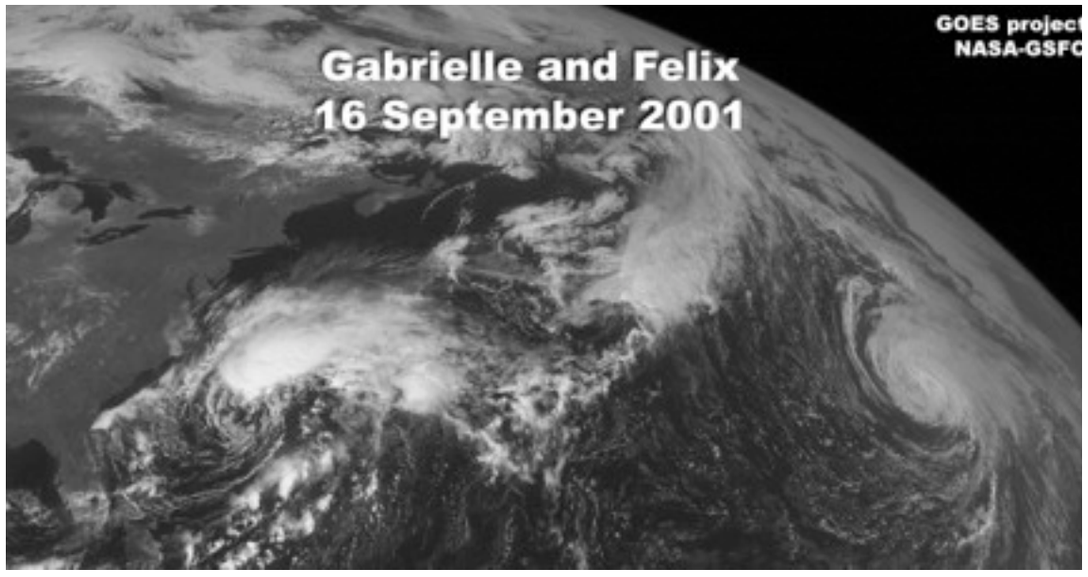
Nearly 50% of this 57K file identifies as “JPEG”

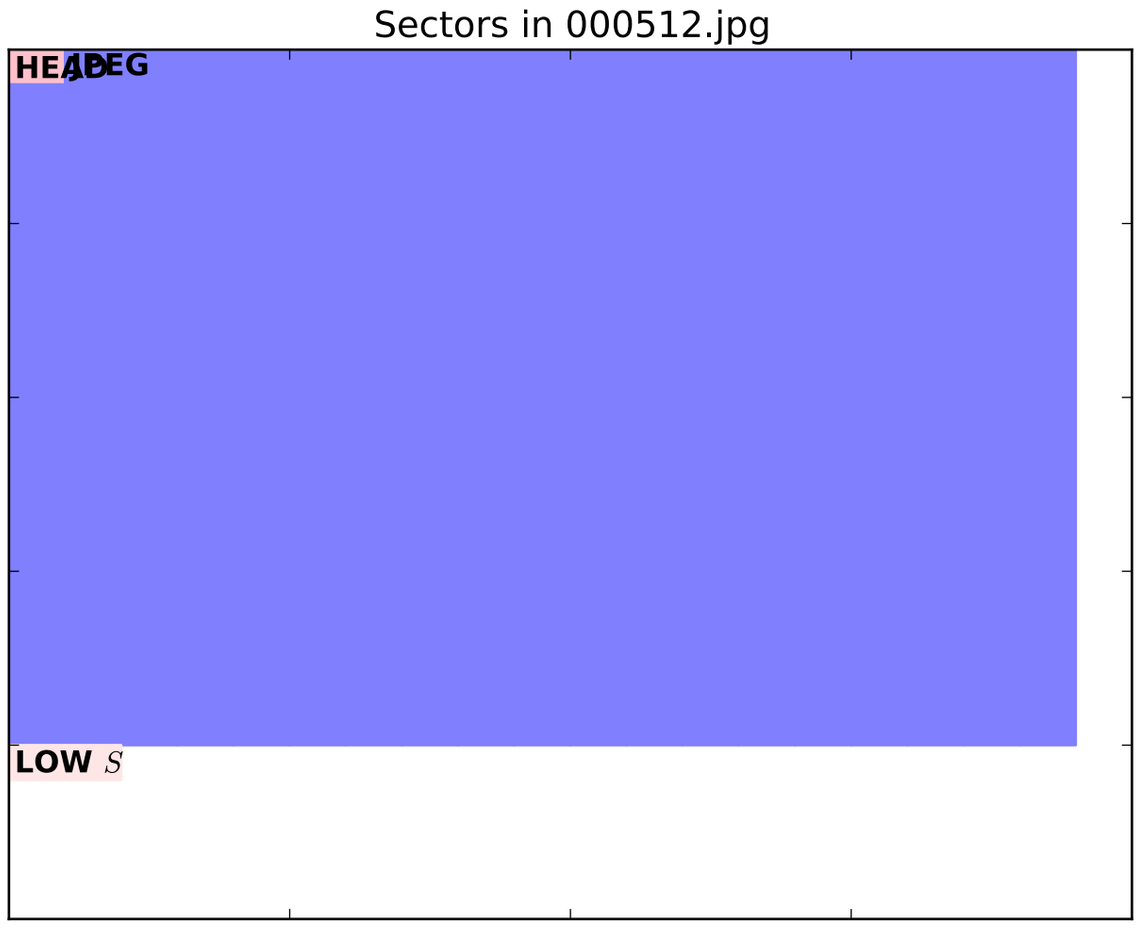


000897.jpg
Bytes: 57596

Sectors: 113

Nearly 100% of this file identifies as “JPEG.”





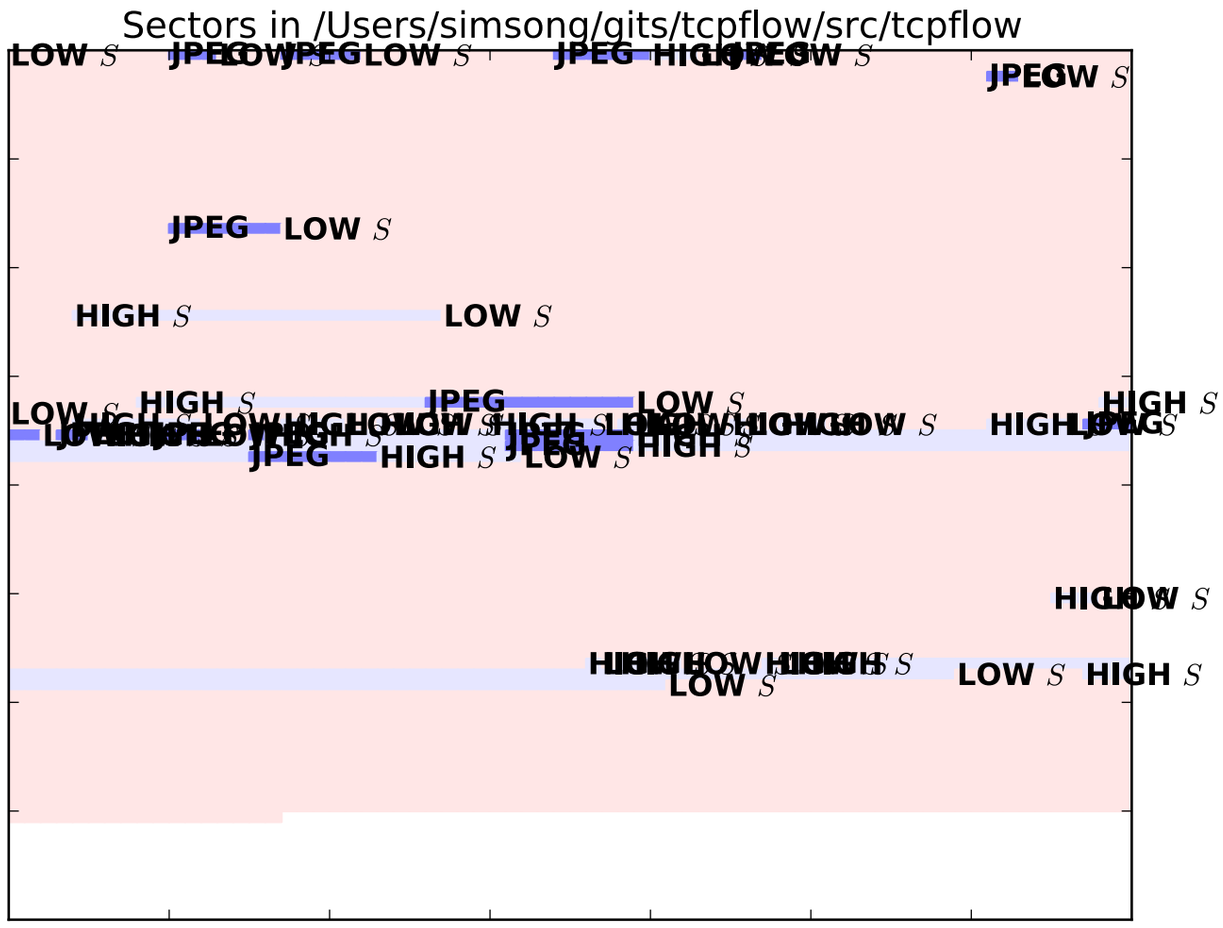
000512.jpg
Bytes: 195,311

Sectors: 382

Non-JPEG files have a low false-positive rate

file: tcpflow (MacOS executable)

- Total sectors: 4917
- Total “JPEG” false positives: 73
- False positive rate: 1.5%

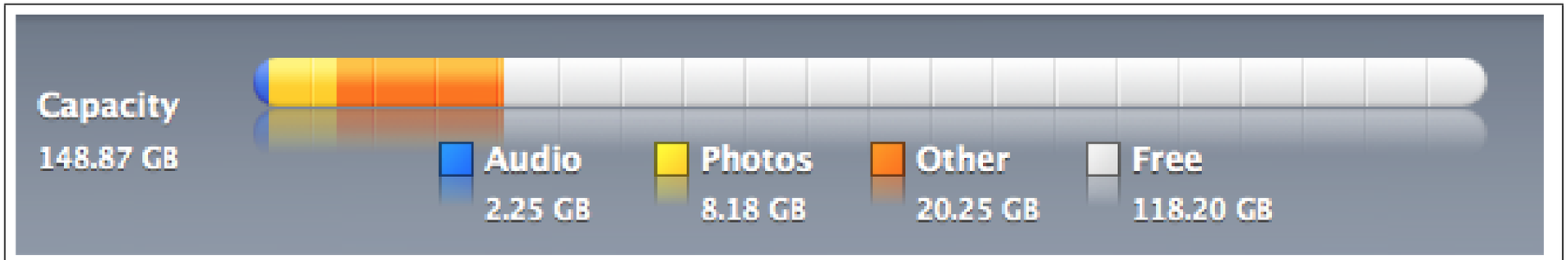


This is called the *file fragment classification problem.*

We can reliably classify JPEG, MPEG, Huffman, and other types.

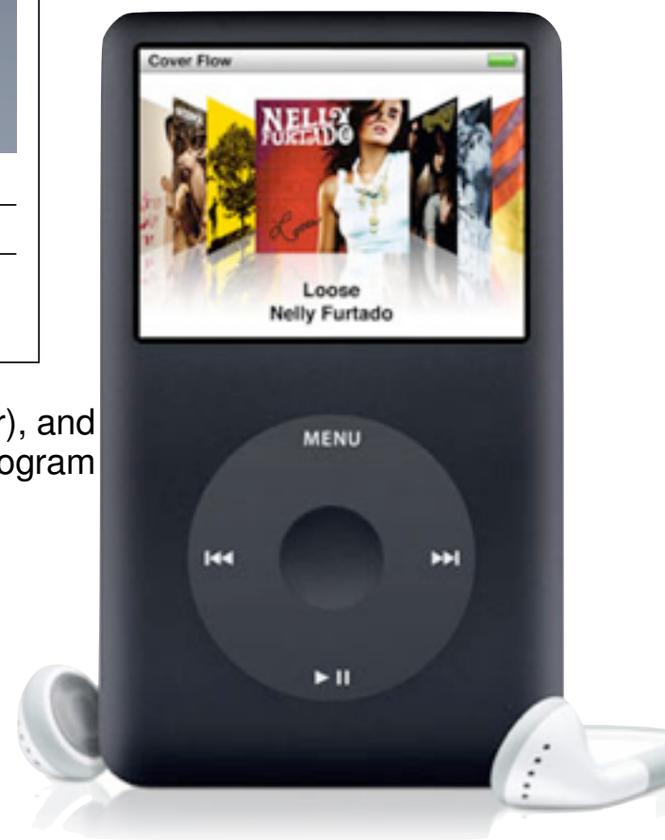
Combine random sampling with sector ID to obtain the forensic contents of a storage device.

Our numbers from sampling are similar to those reported by iTunes.



Audio Data reported by iTunes:	2.25 GiB	2.42 GB	
MP3 files reported by file system:		2.39 GB	
Estimated MP3 usage with random sampling :		2.49 GB	10,000 random samples
		2.71 GB	5,000 random samples

Figure 1: Usage of a 160GB iPod reported by iTunes 8.2.1 (6) (top), as reported by the file system (bottom center), and as computing with random sampling (bottom right). Note that iTunes usage actually in GiB, even though the program displays the "GB" label.



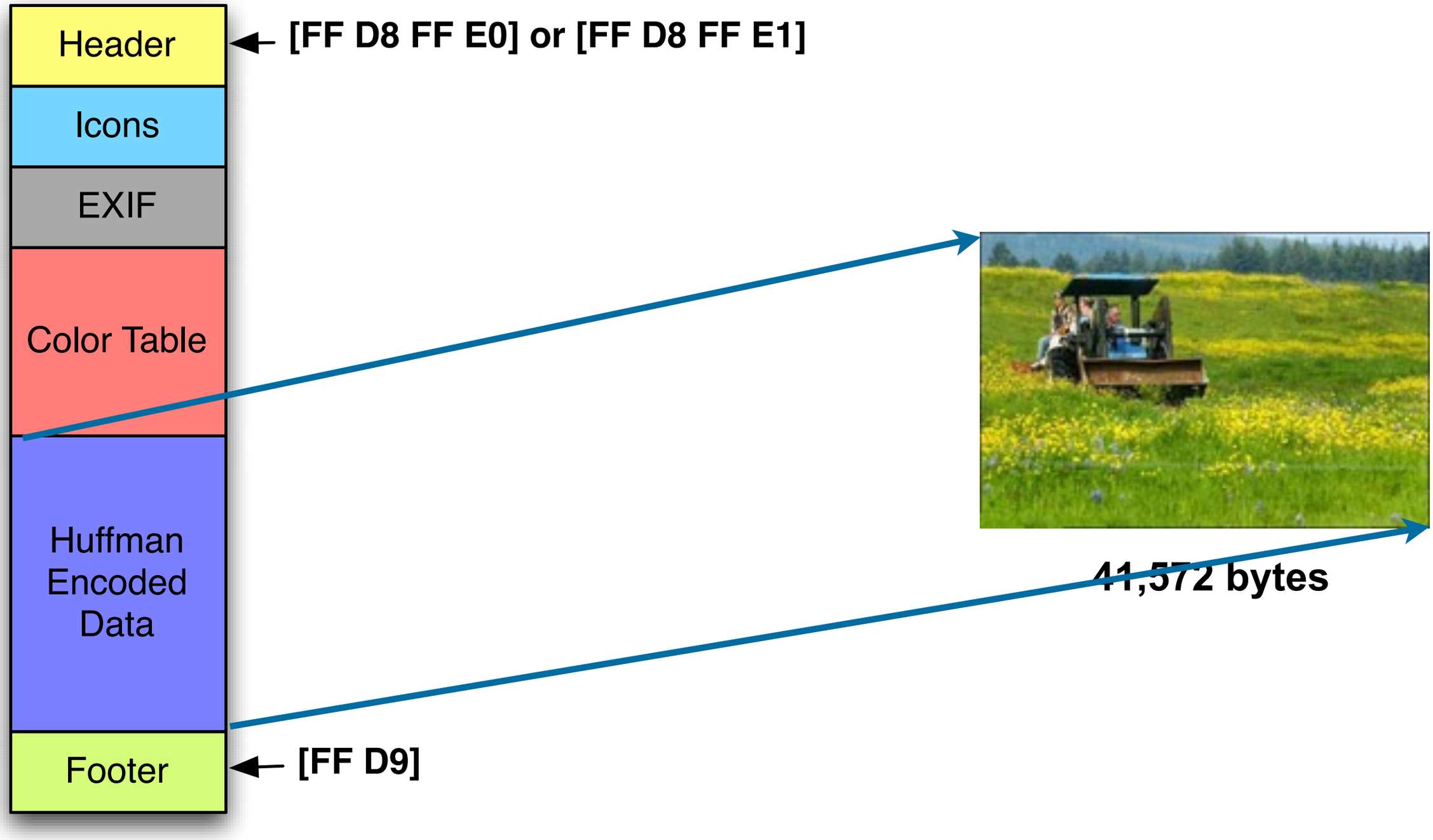
We accurately determined:

- % of free space; % JPEG; % encrypted

—*Simson Garfinkel, Vassil Roussev, Alex Nelson and Douglas White, Using purpose-built functions and block hashes to enable small block and sub-file forensics, DFRWS 2010, Portland, OR*

Finding Known Content with Sector Hashing...

Most forensics processing tries to understand the internal structure of data files...



Files can also be viewed as a set of ordered blocks.

41,572 bytes

Block #	Byte Range	Values...
0	0- 511	ffd8 ffe0 0010 4a46 4946 0001 0201 0048...
1	512-1023	0c0c 0c0c ffc0 0011 0800 6a00 a003 0122...
2	1024-1535	4fa7 7567 ded2 cac5 8c82 2bf4 9e1c 23f9...
3	1536-2047	fafd 1527 e459 e934 c173 59ad 9234 f09f...
4	...	

Compute the cryptographic hash of each block. These are “block hashes.”

Block #	Byte Range	MD5*(block(N))
0	0- 511	**dc0c20abad42d487a74f308c69d18a5a**
1	512-1023	**9e7bc64399ad87ae9c2b545061959778**
2	1024-1535	**6e7f3577b100f9ec7fae18438fd5b047**
3	1536-2047	**4594899684d0565789ae9f364885e303**
4	...	

Question: how often do *these* block hashes occur in other JPEGs?

Should these block hashes be in other files?

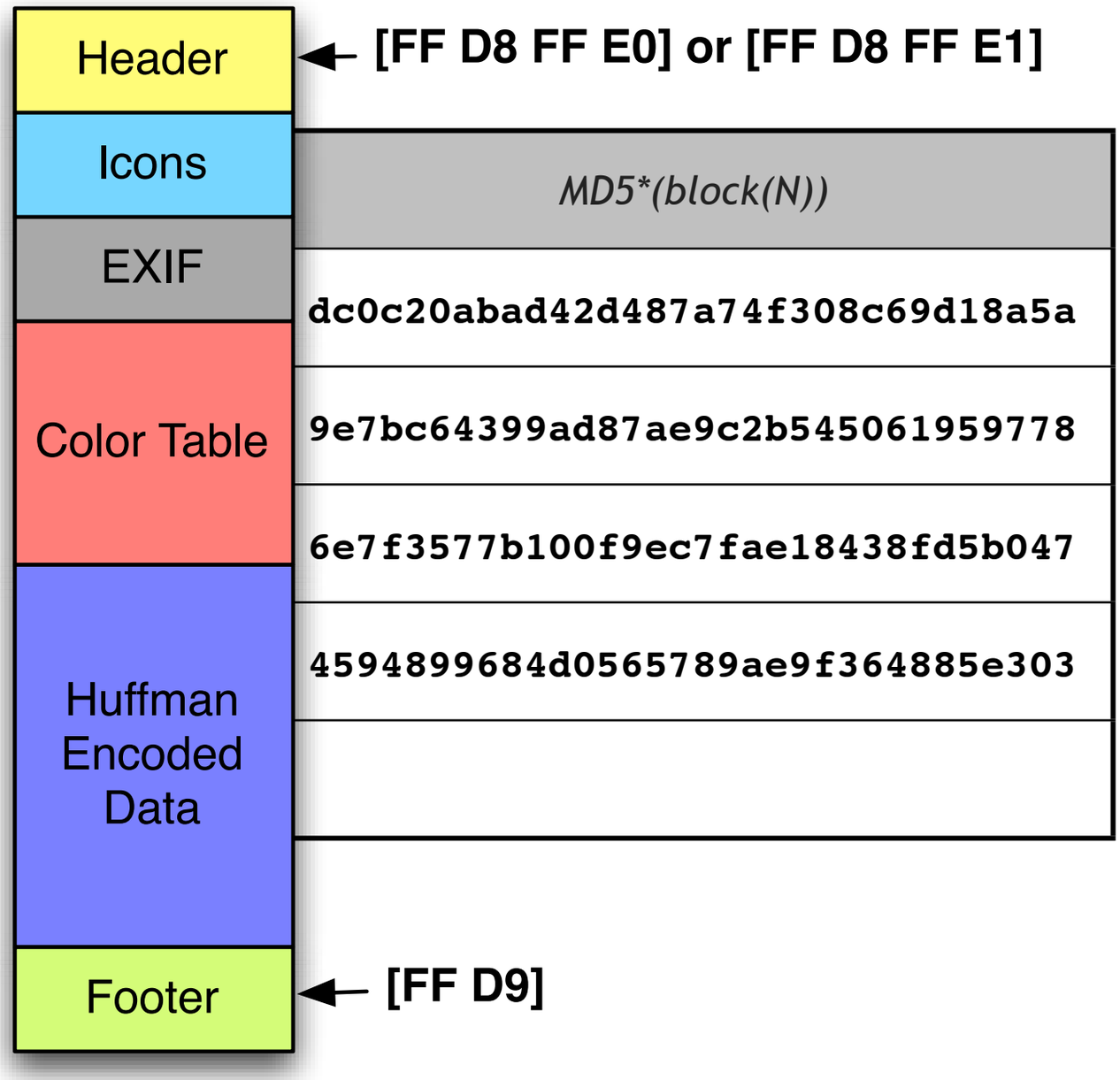
Specific byte sequences in high-entropy data are very rare.

- 512 bytes = 256^{512} =$10^{1,233}$ possible sectors

But metadata might be common:

- Specific headers
- Common color tables
- “all black”

You need to survey the datasphere to find out.



We examined sector hashes from ≃4 million files

≃ 1 million in GOVDOCS1 collection

= 109,282 JPEGs (including 000107)

≃ 3 million samples of Windows malware

Our results:

- Most of the block hashes in 000107.jpg did not appear elsewhere in the corpus.
- Some of the block hashes appeared in other JPEGs.
- None of the block hashes appeared in files that were not JPEGs

The beginning of 000107.jpg contained distinct hashes...

```
dc0c20abad42d487a74f308c69d18a5a   offset 0-511          1
9e7bc64399ad87ae9c2b545061959778   offset 512-1023       1
6e7f3577b100f9ec7fae18438fd5b047   offset 1024-1535      1
4594899684d0565789ae9f364885e303   offset 1536-2047      1
4d21b27ceec5618f94d7b62ad3861e9a   offset 2048-2559      1
03b6a13453624f649bbf3e9cd83c48ae   offset 2560-3071      1
c996fe19c45bc19961d2301f47cabaa6   offset 3072-3583      1
0691baa904933c9946bbda69c019be5f   offset 3584-4095      1
1bd9960a3560b9420d6331c1f4d95fec   offset 4096-4607      1
52ef8fe0a800c9410bb7a303abe35e64   offset 4608-5119      1
b8d5c7c29da4188a4dcaa09e057d25ca   offset 5120-5631      1
3d7679a976b91c6eb8acd1bfa3414f96   offset 5632-6143      1
8649f180275e0b63253e7ee0e8fa4c1d   offset 6144-6655      1
60ebc8acb8467045e9dcbe207f61a6c2   offset 6656-7167      1
440c1c1318186ac0e42b2977779514a1   offset 7168-7679      1
72686172f8c865231e2b30b2829e3dd9   offset 7680-8191      1
fdff55c618d434416717e5ed45cb407e   offset 8192-8703      1
fcd89d71b5f728ba550a7bc017ea8ff1   offset 8704-9215      1
2d733e47c5500d91cc896f99504e0a38   offset 9216-9727      1
2152fdde0e0a62d2e10b4fecc369e4c6   offset 9728-10239     1
692527fa35782db859248634 36d45d7f  offset 10240-10751    1
76dbb9b469273d0e0e467a55728b7883   offset 10752-11263    1
171310e61a8e78364b4965b995f16ff5   offset 11264-11775    1
6865477474f8a6011108c9cbf1fff0f9   offset 11776-12287    1
```

The middle of 000107.JPG had hash collisions...

9df886fdfa6934cc7dcf10c04be3464a	offset 14848-15359	1
95399e7ecc7ba1b38243069bdd5c263a	offset 15360-15871	1
ef1ffcdc11162ecdfedd2dde644ec8f2	offset 15872-16383	1
7eb35c161e91b215e2a1d20c32f4477e	offset 16384-16895	1
38f9b6f045db235a14b49c3fe7b1cec3	offset 16896-17407	1
edceba3444b5551179c791ee3ec627a5	offset 17408-17919	1
6bc8ed0ce3d49dc238774a2bdeb7eca7	offset 17920-18431	1
5070e4021866a547aa37e5609e401268	offset 18432-18943	14
13d33222848d5b25e26aefb87dbdf294	offset 18944-19455	9198
0dfcde85c648d20aed68068cc7b57c25	offset 19456-19967	9076
756f0bbe70642700aafb2557bf2c5649	offset 19968-20479	9118
c2c29016d3005f7a1df247168d34e673	offset 20480-20991	9237
42ff3d72b2b25f880be21fac46608cc9	offset 20992-21503	9708
b943cd0ea25e354d4ac22b886045650d	offset 21504-22015	9615
a003ec2c4145b0bc871118842b74f385	offset 22016-22527	9564
1168c351f57aad14de135736c06665ea	offset 22528-23039	7
51a50e6148d13111669218dc40940ce5	offset 23040-23551	83
365b122f53075cb76b39ca1366418ff9	offset 23552-24063	83
9ad9660e7c812e2568aaf063a1be7d05	offset 24064-24575	84
67bd01c2878172e2853f0aef341563dc	offset 24576-25087	84
fc3e47d734d658559d1624c8b1cbf2c1	offset 25088-25599	84
cb9aef5b7f32e2a983e67af38ce8ff87	offset 25600-26111	1
531aea9e5b2987f923b0f0812bd5846e	offset 26112-26623	1
cef61251eb556fd095b3347dc87d8a24	offset 26624-27135	1

Block 37 had 9198 collisions..
The sector is filled with blank lines 100 characters long...

```
13d33222848d5b25e26aefb87dbdf294   offset 18944-19455      9198
$ dd if=000107.jpg skip=18944 count=512 bs=1|xxd
0000000: 2020 2020 2020 2020 2020 2020 2020 2020
0000010: 2020 2020 2020 2020 2020 2020 0a20 2020                .
0000020: 2020 2020 2020 2020 2020 2020 2020 2020
0000030: 2020 2020 2020 2020 2020 2020 2020 2020
0000040: 2020 2020 2020 2020 2020 2020 2020 2020
0000050: 2020 2020 2020 2020 2020 2020 2020 2020
0000060: 2020 2020 2020 2020 2020 2020 2020 2020
0000070: 2020 2020 2020 2020 2020 2020 2020 2020
0000080: 200a 2020 2020 2020 2020 2020 2020 2020    .
0000090: 2020 2020 2020 2020 2020 2020 2020 2020
00000a0: 2020 2020 2020 2020 2020 2020 2020 2020
00000b0: 2020 2020 2020 2020 2020 2020 2020 2020
00000c0: 2020 2020 2020 2020 2020 2020 2020 2020
00000d0: 2020 2020 2020 2020 2020 2020 2020 2020
00000e0: 2020 2020 2020 0a20 2020 2020 2020 2020          .
00000f0: 2020 2020 2020 2020 2020 2020 2020 2020
0000100: 2020 2020 2020 2020 2020 2020 2020 2020
0000110: 2020 2020 2020 2020 2020 2020 2020 2020
0000120: 2020 2020 2020 2020 2020 2020 2020 2020
0000130: 2020 2020 2020 2020 2020 2020 2020 2020
0000140: 2020 2020 2020 2020 2020 200a 2020 2020                .
0000150: 2020 2020 2020 2020 2020 2020 2020 2020
0000160: 2020 2020 2020 2020 2020 2020 2020 2020
...
```

Block 45 had 83 collisions..
It appears to contain EXIF metadata

```
51a50e6148d13111669218dc40940ce5   offset 23040-23551        83
$ dd if=000107.jpg skip=23040 count=512 bs=1|xxd
0000000: 3936 362d 322e 3100 0000 0000 0000 0000  966-2.1.........
0000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000030: 0000 0000 0000 0000 0058 595a 2000 0000  .........XYZ ...
0000040: 0000 00f3 5100 0100 0000 0116 cc58 595a  ....Q........XYZ
0000050: 2000 0000 0000 0000 0000 0000 0000 0000   ...............
0000060: 0058 595a 2000 0000 0000 006f a200 0038  .XYZ ......o...8
0000070: f500 0003 9058 595a 2000 0000 0000 0062  .....XYZ ......b
0000080: 9900 00b7 8500 0018 da58 595a 2000 0000  .........XYZ ...
0000090: 0000 0024 a000 000f 8400 00b6 cf64 6573  ...$.........des
00000a0: 6300 0000 0000 0000 1649 4543 2068 7474  c........IEC htt
00000b0: 703a 2f2f 7777 772e 6965 632e 6368 0000  p://www.iec.ch..
00000c0: 0000 0000 0000 0000 0016 4945 4320 6874  ..........IEC ht
00000d0: 7470 3a2f 2f77 7777 2e69 6563 2e63 6800  tp://www.iec.ch.
00000e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000100: 0000 0000 0000 0000 0000 0000 0064 6573  .............des
0000110: 6300 0000 0000 0000 2e49 4543 2036 3139  c........IEC 619
0000120: 3636 2d32 2e31 2044 6566 6175 6c74 2052  66-2.1 Default R
0000130: 4742 2063 6f6c 6f75 7220 7370 6163 6520  GB colour space
0000140: 2d20 7352 4742 0000 0000 0000 0000 0000  - sRGB..........
0000150: 002e 4945 4320 3631 3936 362d 322e 3120  ..IEC 61966-2.1
0000160: 4465 6661 756c 7420 5247 4220 636f 6c6f  Default RGB colo
0000170: 7572 2073 7061 6365 202d 2073 5247 4200  ur space - sRGB.
```

Block 48 had 84 collisions..
It appears to contain part of a JPEG color table...

```
67bd01c2878172e2853f0aef341563dc   offset 24576-25087          84
$ dd if=000107.jpg skip=24576 count=512 bs=1 |xxd
0000000: 7a27 ab27 dc28 0d28 3f28 7128 a228 d429  z'.'.(.(?(q(.(.)
0000010: 0629 3829 6b29 9d29 d02a 022a 352a 682a  .)8)k).).*.*5*h*
0000020: 9b2a cf2b 022b 362b 692b 9d2b d12c 052c  .*.+.+6+i+.+.,.,
0000030: 392c 6e2c a22c d72d 0c2d 412d 762d ab2d  9,n,.,.-.-A-v-.-
0000040: e12e 162e 4c2e 822e b72e ee2f 242f 5a2f  ....L....../$/Z/
0000050: 912f c72f fe30 3530 6c30 a430 db31 1231  ././.0501l0.0.1.1
0000060: 4a31 8231 ba31 f232 2a32 6332 9b32 d433  J1.1.1.2*2c2.2.3
0000070: 0d33 4633 7f33 b833 f134 2b34 6534 9e34  .3F3.3.3.4+4e4.4
0000080: d835 1335 4d35 8735 c235 fd36 3736 7236  .5.5M5.5.5.676r6
0000090: ae36 e937 2437 6037 9c37 d738 1438 5038  .6.7$7`7.7.8.8P8
00000a0: 8c38 c839 0539 4239 7f39 bc39 f93a 363a  .8.9.9B9.9.9.:6:
00000b0: 743a b23a ef3b 2d3b 6b3b aa3b e83c 273c  t:.:.;-;k;.;.<'<
00000c0: 653c a43c e33d 223d 613d a13d e03e 203e  e<.<.="=a=.=.> >
00000d0: 603e a03e e03f 213f 613f a23f e240 2340  `>.>.?!?a?.?.@#@
00000e0: 6440 a640 e741 2941 6a41 ac41 ee42 3042  d@.@.A)AjA.A.B0B
00000f0: 7242 b542 f743 3a43 7d43 c044 0344 4744  rB.B.C:C}C.D.DGD
0000100: 8a44 ce45 1245 5545 9a45 de46 2246 6746  .D.E.EUE.E.F"FgF
0000110: ab46 f047 3547 7b47 c048 0548 4b48 9148  .F.G5G{G.H.HKH.H
0000120: d749 1d49 6349 a949 f04a 374a 7d4a c44b  .I.IcI.I.J7J}J.K
0000130: 0c4b 534b 9a4b e24c 2a4c 724c ba4d 024d  .KSK.K.L*LrL.M.M
0000140: 4a4d 934d dc4e 254e 6e4e b74f 004f 494f  JM.M.N%NnN.O.OIO
0000150: 934f dd50 2750 7150 bb51 0651 5051 9b51  .O.P'PqP.Q.QPQ.Q
0000160: e652 3152 7c52 c753 1353 5f53 aa53 f654  .R1R|R.S.S_S.S.T
0000170: 4254 8f54 db55 2855 7555 c256 0f56 5c56  BT.T.U(UuU.V.V\V
```

With blocks of 512 bytes and 4KiB, the vast majority of sectors had distinct hashes.

Table 1. Incidence of singleton, paired, and common sectors in three file corpora.

No. of blocks	Govdocs	OpenMalware 2012	2009 NSRL RDS
Block size: 512 bytes			
Singleton	911.4 M (98.93%)	1,063.1 M (88.69%)	N/A
Pair	7.1 M (.77%)	75.5 M (6.30%)	N/A
Common	2.7 M (.29%)	60.0 M (5.01%)	N/A
Block size: 4 kibibytes			
Singleton	117.2 M (99.46%)	143.8 M (89.51%)	567.0 M (96.00%)
Pair	0.5 M (.44%)	9.3 M (5.79%)	16.4 M (2.79%)
Common	0.1 M (.11%)	7.6 M (4.71%)	7.1 M (1.21%)

Young, Foster, Garfinkel & Fairbanks, IEEE Computer, Dec. 2012

File systems align large files on sector boundaries. We hash file blocks and identify sectors that match.

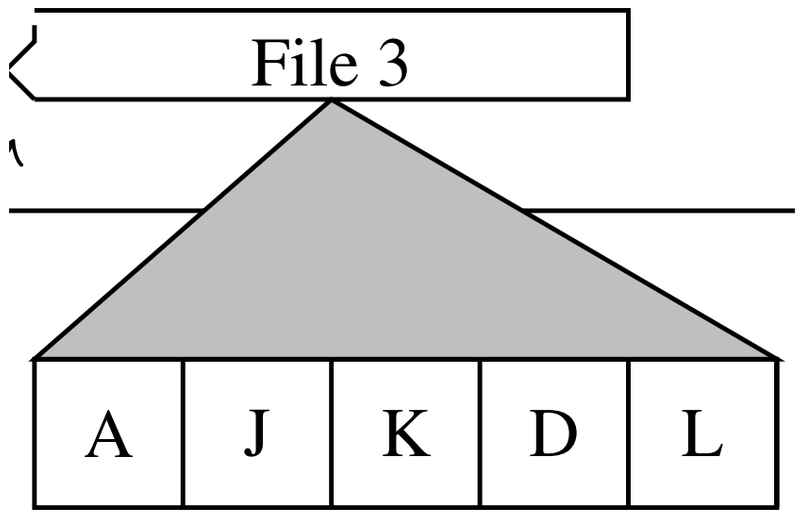
Block #	Byte Range	MD5*(block(N))
0	0– 511	**dc0c20abad42d487a74f308c69d18a5a**
1	512–1023	**9e7bc64399ad87ae9c2b545061959778**
2	1024–1535	**6e7f3577b100f9ec7fae18438fd5b047**
3	1536–2047	**4594899684d0565789ae9f364885e303**
4	...	

Using distinct sectors in media sampling and full media analysis to detect presence of documents from a corpus, Kristina Foster, NPS Master's Thesis, 2012

This means we can use distinct sectors to find known content.

Method #1 — Full media sampling

- Read & hash every disk sector.
- Lookup hash values in a database of block hashes.
- Distinct hash imply presence of files.
- Advantage: Can find a single sector of target content



Method #2 — Random sampling

- Read & hash randomly chosen sectors.
- Lookup hash values in a database of block hashes.
- Distinct hash implies presence of files.
- Advantage: Can find presence of target content very quickly

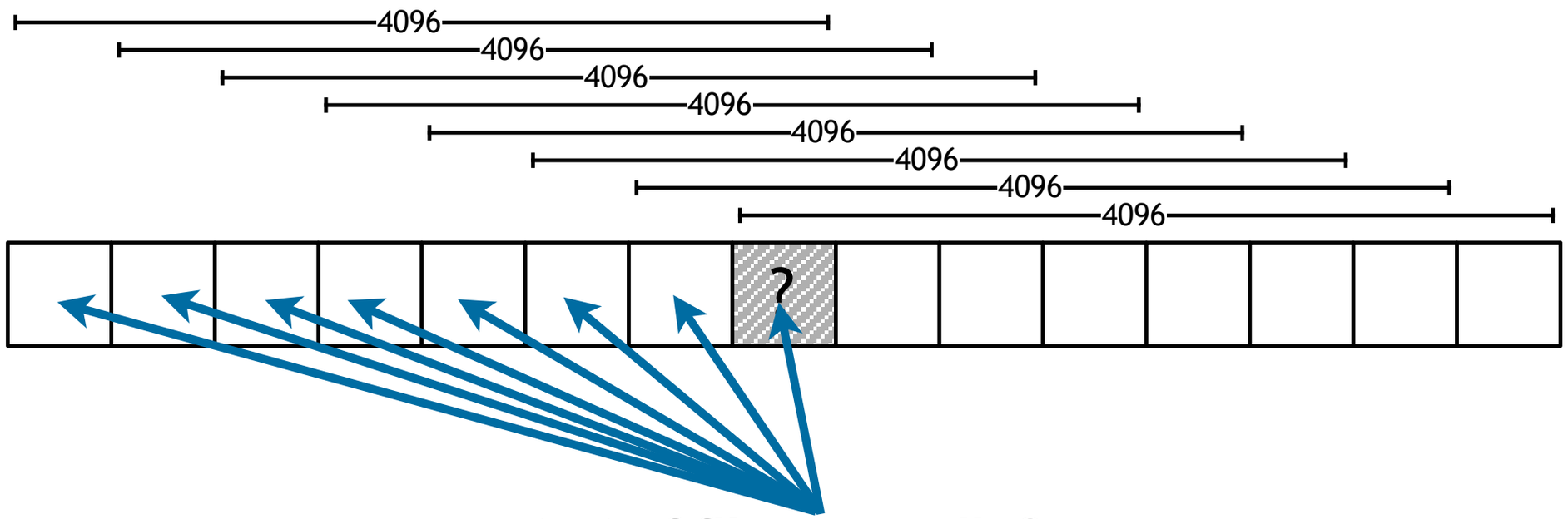
There are significant hash and database requirements.

1TB data in 208 minutes

- ≃ 80 Mbyte/sec
- ≃ 150,000 512-byte sectors/sec
- ≃ 150,000 database lookups/sec

		
Minutes	208	5
Max Data	1 TB	36 GB
Max Seeks		90,000

Alignment uncertainty gives 4096-byte sectors same performance requirements:



By combining a Bloom filter & database, we can perform up to 2.7M TPS on low-cost hardware

Table 2. Total transactions per second (TPS) for best execution.

Bloom filter			Database		TPS at 1 M lookups		TPS at 1,200 seconds	
k	*M*	Size	Strategy	Size	Present	Absent	Present	Absent
100 million records								
3	31	257 MiBytes	B-tree (preload)	2.3 GiBytes	35.3 K	49.5 K	161.3 K	1.8 M
3	31	257 MiBytes	B-tree	2.3 GiBytes	11.6 K	565.8 K	156.8 K	2.3 M
3	31	257 MiBytes	Hash map	5.3 GiBytes	13.9 K	656.9 K	641.9 K	3.0 M
3	31	257 MiBytes	Flat map	2.2 GiBytes	28.2 K	746.9 K	356.4 K	2.6 M
3	31	257 MiBytes	Red/black tree	6.0 GiBytes	12.9 K	694.5 K	187.0 K	2.7 M
1 billion records								
3	34	2.1 GiBytes	B-tree (preload)	23 GiBytes	2.2 K	6.1 K	3.6 K	23.1 K
3	33	1.1 GiBytes	B-tree	23 GiBytes	2.6 K	85.8 K	3.7 K	114.9 K
3	33	1.1 GiBytes	Hash map	57 GiBytes	–	–	0.3 K	3.1 K
3	34	2.1 GiBytes	Flat map	22 GiBytes	–	–	0.4 K	4.0 K
3	33	1.1 GiBytes	Red/black tree	60 GiBytes	–	–	0.1 K	1.4 K

Hardware: 8GiB Laptop; 250GB external SSD.

—*"Distinct sector hashes for target file detection," Young, Garfinkel, Foster & Fairbanks, IEEE Computer, Dec. 2012*

Putting it all together, we have a significant innovation... field deployable on a single laptop.

Use Case #1: Rapidly search for known contraband:

- 1TB subject hard drive.
- 10 min x 60 min/sec x 1000 msec/sec / 3 msec/sample = 200,000 samples
- Searching for a sector from a corpus of 512GB
- 100% recognition of a single sector; 0% false positive rate

Amount of Contraband	p (prob of missing contraband)
5 MB	0.3654
10 MB	0.1335
15 MB	0.0488
20 MB	0.0178
25 MB	0.0065

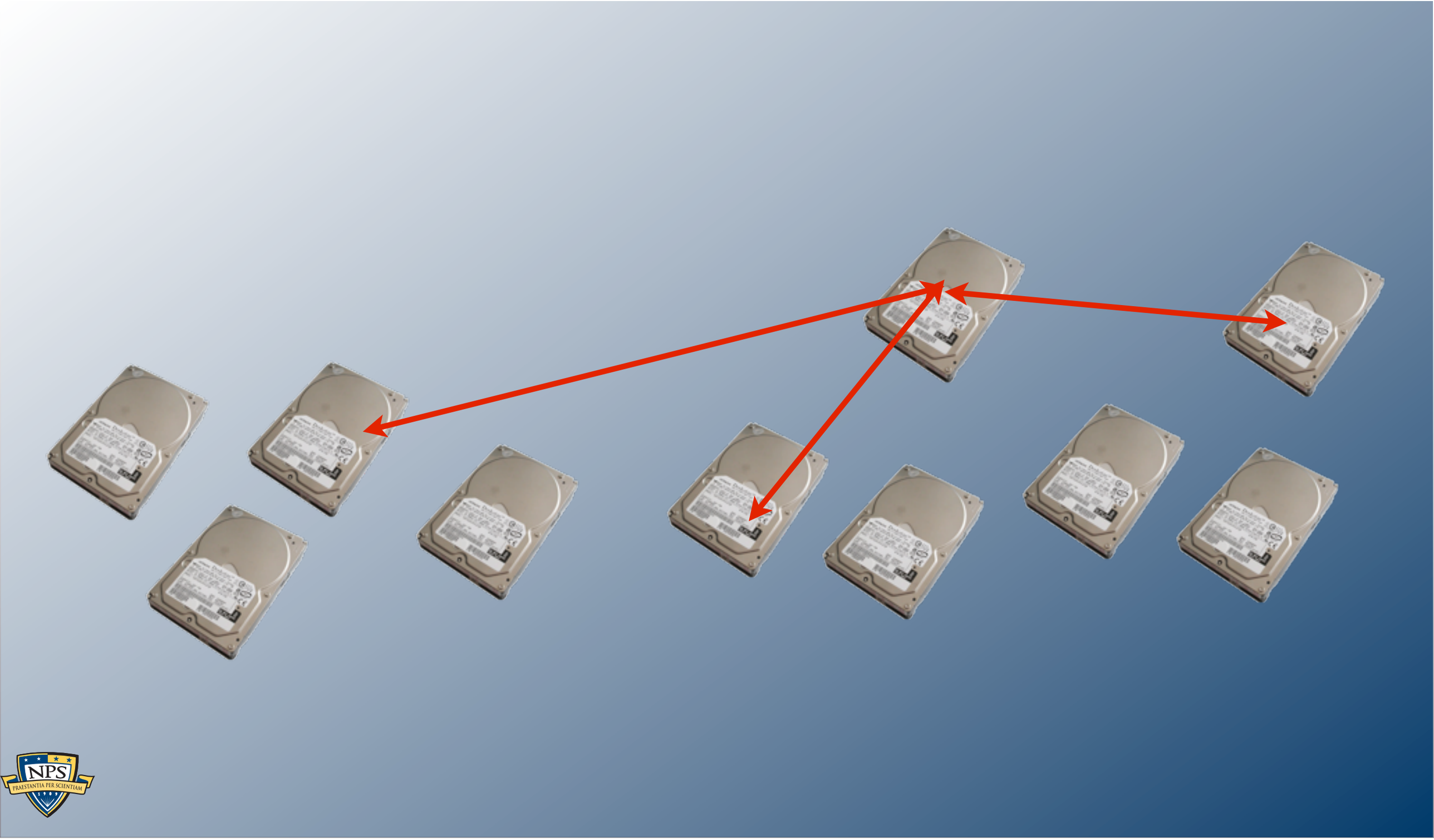
Use Case #2: Find a single sector of known contraband:

- Time to read data & search database: 208 minutes

Technique is file type and file system agnostic

—*JPEG; Video; MSWord; Encrypted PDFs...*

—*provided data is not modified when copied or otherwise re-coded*



Where do we go from here?

Digital Forensics has exciting problems…
… but they are messy

Math and Science:

- Algorithms tolerant of data that are *dirty* and *damaged.*
- New approaches for handling data that are compressed, encoded or encrypted
- Linguistics, Natural Language Processing & Machine Learning
- Visualization

Engineering:

- Reverse engineering & product development
- Approaches for dealing with large data volumes (100TB — 10PB)
- Software that doesn't crash
- Cloud forensics

Many of the techniques here are also applicable to:

- Social network analysis
- Personal information management
- Data mining unstructured information

Dec. 2012

Please try our tools!

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/S0167404812001472 (paper)

  —*Computers & Security, 2013*

  —*http://simson.net/clips/academic/2013.COSE.bulk_extractor.pdf*

DFXML — An XML language for doing computer forensics

- provenance, file extraction, hashes and piecewise-hashes, registry values, etc.
- https://github.com/simsong/dfxml
- http://www.sciencedirect.com/science/article/pii/S1742287611000910

  —*Digital Investigation, 2012*

  —*http://simson.net/clips/academic/2012.DI.dfxml.pdf*

Data!

- http://digitalcorpora.org/

Contact Information:
Simson L. Garfinkel
simsong@acm.org
http://simson.net/