



Automated Digital Forensics

Simson L. Garfinkel

Associate Professor, Naval Postgraduate School

October 18, 2010

<http://simson.net/>

NPS is the Navy's Research University.



Location: Monterey, CA

Campus Size: 627 acres

Students: 1500

- US Military (All 5 services)
- US Civilian (Scholarship for Service & SMART)
- Foreign Military (30 countries)

Schools:

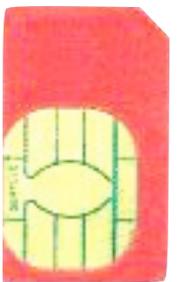
- Business & Public Policy
- Engineering & Applied Sciences
- International Graduate Studies
- Operational & Information Sciences



Law enforcement & military agencies encounter substantial amounts of electronic media.

Typical media includes:

- Desktop & Laptop computers (hard drives)
- Cell phones (SIM chips, flash memory)
- iPods & MP3 music players
- GPS Devices



Typical sources includes:

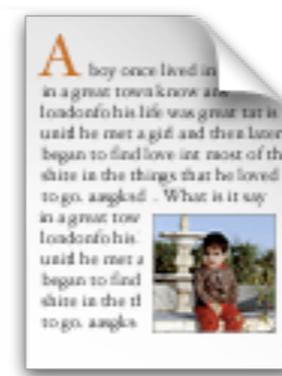
- Domestic searches
- Border searches
- Media collected on the “battlefield”:
 - *on combatants*
 - *inside houses & apartments*
- Cyber security
 - *victim systems*
 - *attacker systems*
 - *intermediaries*



Data quality and quantity make digital forensics hard.

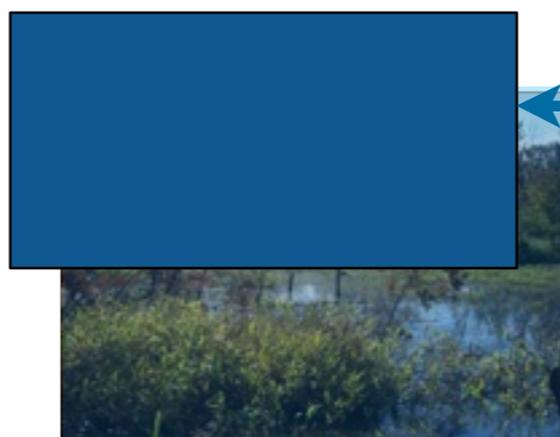
Quality: any piece of data may be critical.

- Heterogeneity is a problem.
 - *Address books*
 - *Email*
 - *Documents*
 - *Photos*



Digital Forensics relies heavily on residual data.

- Slack space within files.
- Deleted Files
- Partially overwritten files
- Virtual memory fragments
- Hibernation files



Newly written data

Earlier JPEG

— *Residual data frequently reveals facts, motives, state-of-mind, or associations that the subject sought to hide from others.*

— *CS lacks principled techniques for resolving incomplete data structures.*

Data quality and quantity make digital forensics hard.

Quantity: analysts have less time than the subject!

- User spent *years* assembling email, documents, etc.
- Analysts have days or hours to process it.

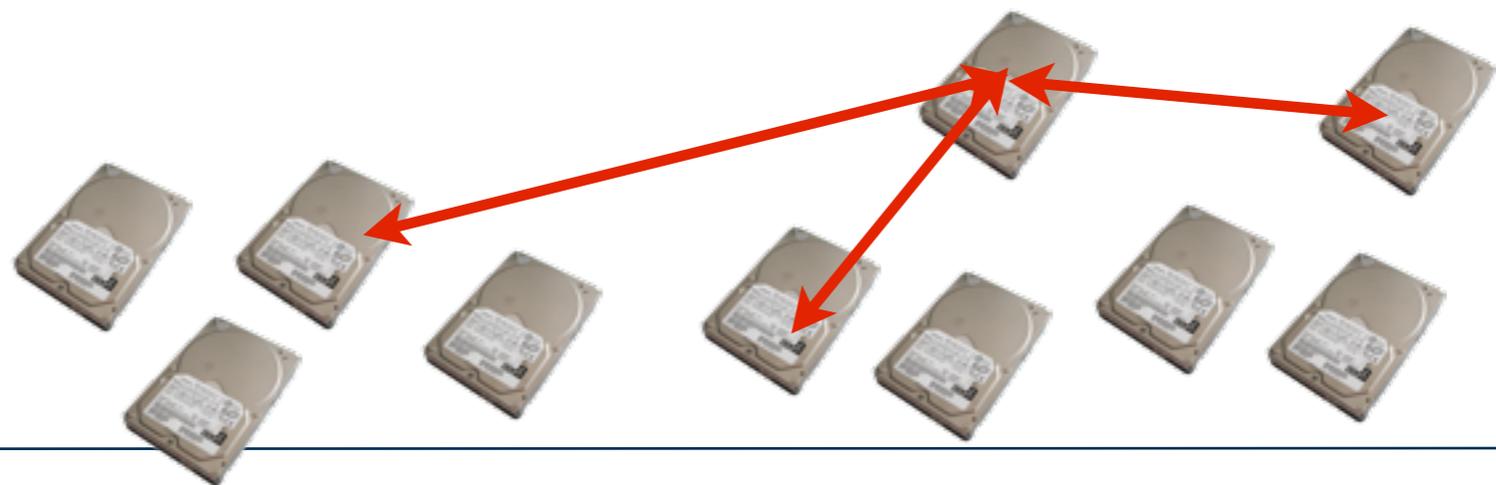


There is no resource advantage.

- Police analyze top-of-the-line systems ... with top-of-the-line systems.
- National Labs have large-scale server farms ... to analyze huge collections.

DF researchers must respond by developing new algorithms that:

- *Provide incisive analysis through cross-drive analysis.*
- *Operate autonomously on incomplete, heterogeneous datasets.*



This talk introduces digital forensics and presents three research projects from my lab.

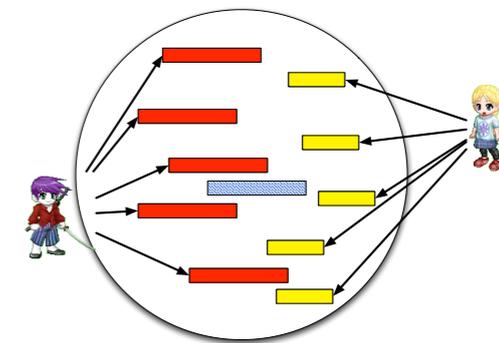
Introducing Digital Forensics



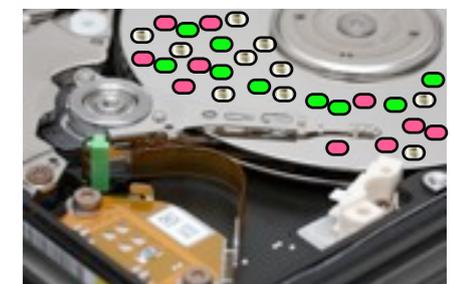
Creating Corpora for Digital Forensics



Multi-User Carved Data Ascription



Random sampling for high speed forensics





Introducing Digital Forensics

Data extraction is the first step of forensic analysis

Imaging tools extract the data without modification.

- "Forensic copy" or "disk image."
- Original media is stored in an evidence locker.



Cell phones present special challenges.

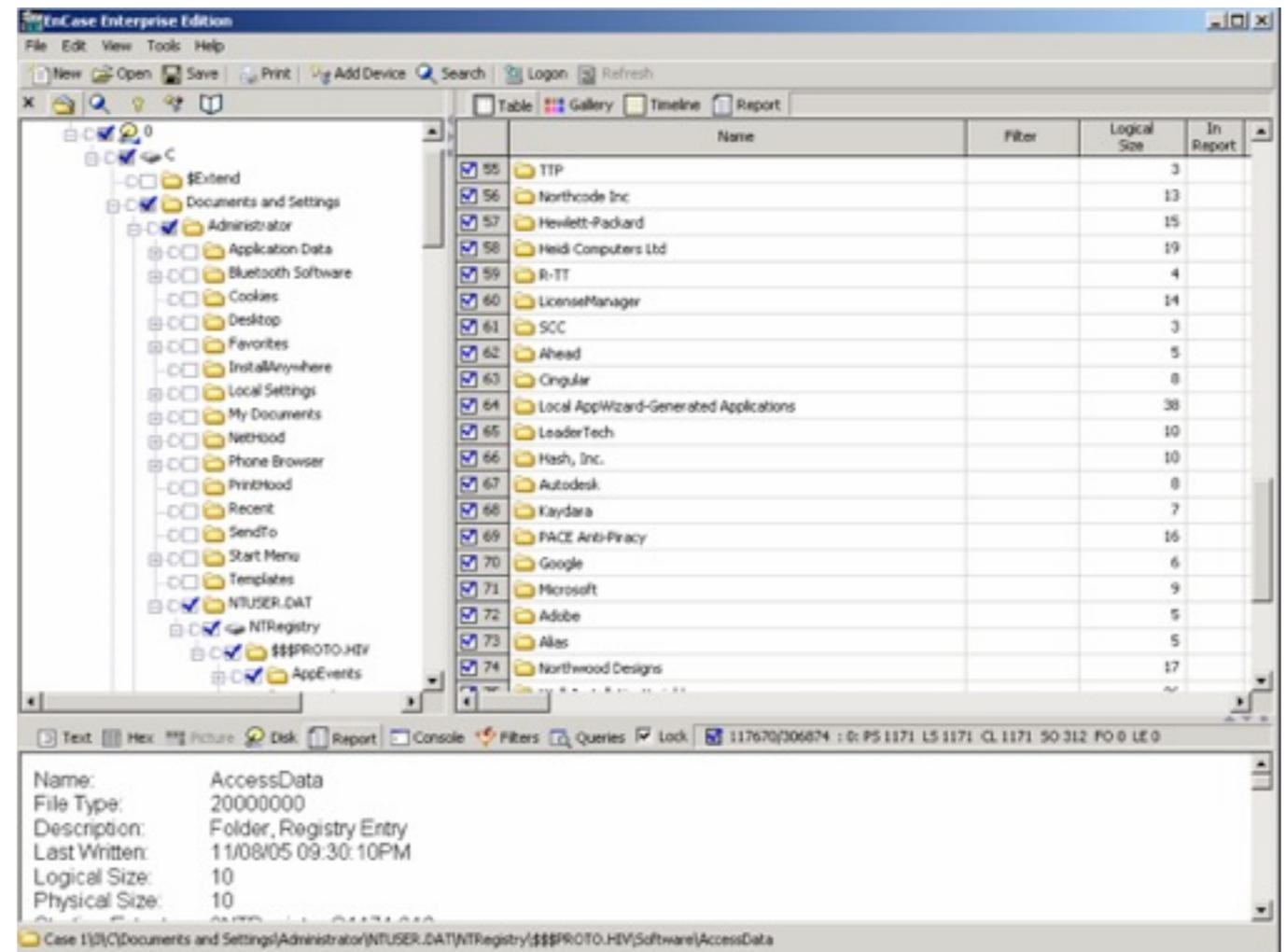
- No standard connectors.
- No standard way to copy data out.
- Difficult to image & store cell phones without changing them.



After extraction, examiners use digital forensic tools to view the evidence.

Today's tools allow the examiner to:

- Display of *allocated & deleted* files.
- String search.
- Data recovery and *file carving*.
- Examining individual disk sectors in hex, ASCII and Unicode



The last decade was a "Golden Age" for digital forensics.

Widespread use of Microsoft Windows, especially Windows XP

Relatively few file formats:

- Microsoft Office (.doc, .xls & .ppt)
- JPEG for images
- AVI and WMV for video



Most examinations confined to a single computer belonging to a single subject



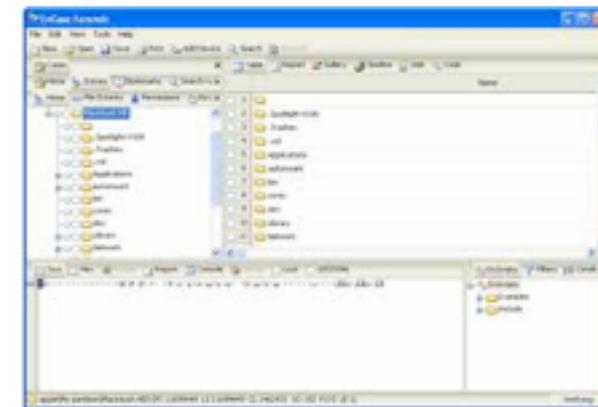
Most storage devices used a standard interface.

- IDE/ATA
- USB



Uniformity gave us good tools and rapid growth.

Commercial tools:



Open Source Tools:



The Sleuth Kit

Content Extraction Toolkits:

Oracle Outside In Technology

Outside In Technology is a suite of software development kits (SDKs) that provides developers with a comprehensive solution to access, transform and control the contents of over 500 unstructured file formats. Each SDK within the suite is optimized to solve a particular problem but they are highly flexible and interoperable. Developers can quickly implement any combination of the Outside In SDKs to provide exactly the right functionality in their application while minimizing integration effort and code footprint. The SDKs offer a wide range of options to give the developer programmatic control of their workflow and output. Thorough documentation and sample applications with source code are included to further accelerate implementation.



[Download](#)



[Documentation](#)

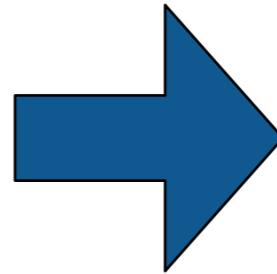


[Sample Code](#)

Today there is a growing digital forensics crisis.

Much of the last decade's progress is quickly becoming irrelevant.

Tools designed to let an analyst find a file and take it into court...



... don't scale to today's problems.

We have identified 5 key problems.



Problem 1 - Increased cost of extraction & analysis.

Data: too much and too complex!

- Increased size of storage systems.
- Cases now require analyzing multiple devices
 - 2 desktops, 6 phones, 4 iPods, 2 digital cameras = 1 case

Shopping results for 2tb drive



[WD Elements Desktop 2 TB External hard](#)
★★★★★ (421)
\$110 new
80 stores



[Seagate Barracuda LP 2 TB Internal](#)
★★★★★ (101)
\$105 new
165 stores



[WD Caviar Green 2 TB Internal hard](#)
★★★★★ (58)
\$99 new
117 stores



[Samsung SpinPoint F3EG Desktop](#)
★★★★★ (8)
\$108 new
44 stores



[WD Caviar Black 2 TB Internal hard](#)
★★★★★ (404)
\$169 new
125 stores

- Non-Removable Flash



- Proliferation of operating systems, file formats and connectors
 - XFAT, XFS, ZFS, YAFFS2, Symbian, Pre, iOS,

Consider FBI Regional Computer Forensic Laboratories growth:

- Service Requests: 5,057 (FY08) → 5,616 (FY09) (+11%)
- Terabytes Processed: 1,756 (FY08) → 2,334 (FY09) (+32%)

Problem 2 — Mobile phones are really hard to examine.

Forensic examiners established bit-copies as the gold standard.

- ... but to image an iPhone, you need to jail-break it.
- Is jail-breaking forensically sound?

How do we validate tools against thousands of phones?

How do we forensically analyze 100,000 apps?

No standardized cables or extraction protocols.



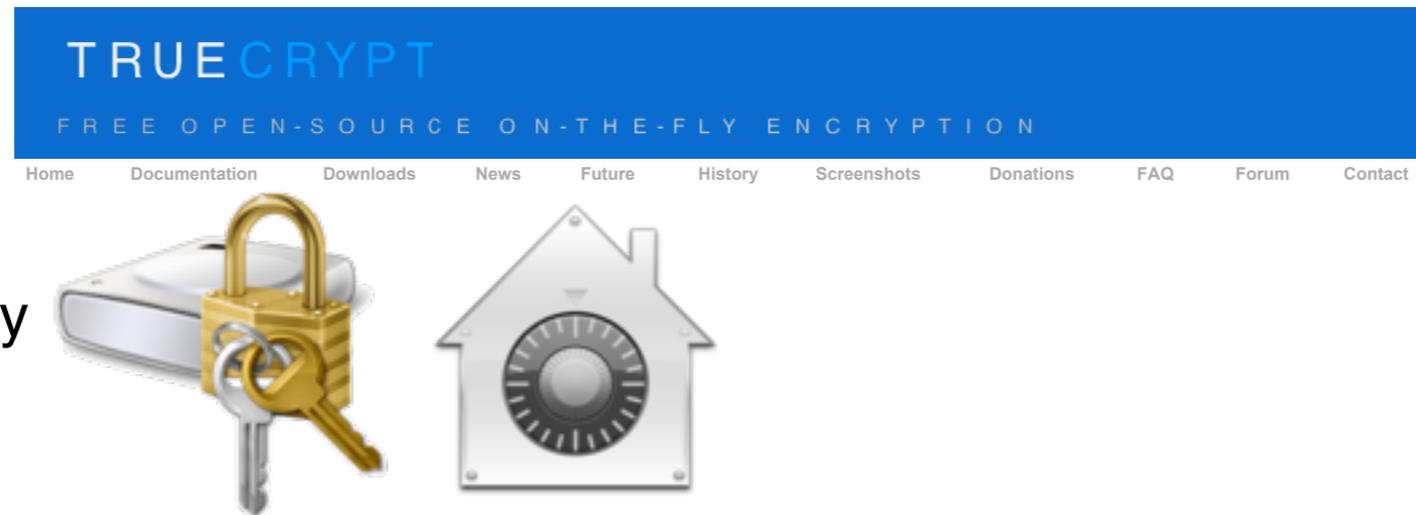
NIST's *Guidelines on Cell Phone Forensics* recommends:

- "searching Internet sites for developer, hacker, and security exploit information."

Problem 3 — Encryption and Cloud Computing make it hard to get to the data

Pervasive Encryption — Encryption is increasingly present.

- TrueCrypt
- BitLocker
- File Vault
- DRM Technology



Cloud Computing — End-user systems won't have the data.

- Google Apps
- Microsoft Office 2010
- Apple Mobile Me



— *But they may have residual data!*

Problem 4 — RAM and hardware forensics is really hard.

RAM Forensics—in its infancy

- RAM structures change frequently (no reason for them to stay constant.)
- RAM is constantly changing.

Malware can hide in many places:

- On disk (in programs, data, or scratch space)
- BIOS & Firmware
- RAID controllers
- GPU
- Ethernet controller
- Motherboard, South Bridge, etc.
- FPGAs



Problem 5 — Time is of the essence.

Most tools were designed to perform a complete analysis.

- Find all the files.
- Index all the terms.
- Report on all the data.
- Take as long as necessary!

Increasingly we are racing the clock:

- Police prioritize based on statute-of-limitations!
- Battlefield, Intelligence & Cyberspace operations require turnaround in days or hours.



My research focuses on three main areas:

Area #1: Data collection and manufacturing

- Large data sets of real data enable science. (+20TB)
- Small data sets of realistic data enable education, training and publishing. (<1TB)

Area #2: Bringing data mining and machine learning to forensics

- Breakthrough algorithms based on correlation and sampling
- Automated social network analysis (cross-drive analysis)
- Automated ascription of carved data

Area #3: Working above and below the files

- Most work to date is with files.
- Digital Forensics XML (DFXML)
 - *Connecting tools.*
 - *Representing applications, behaviors, users.*
- Forensics of bulk data

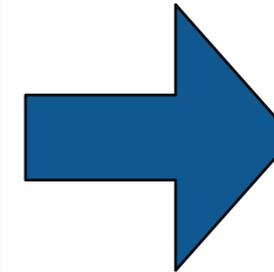
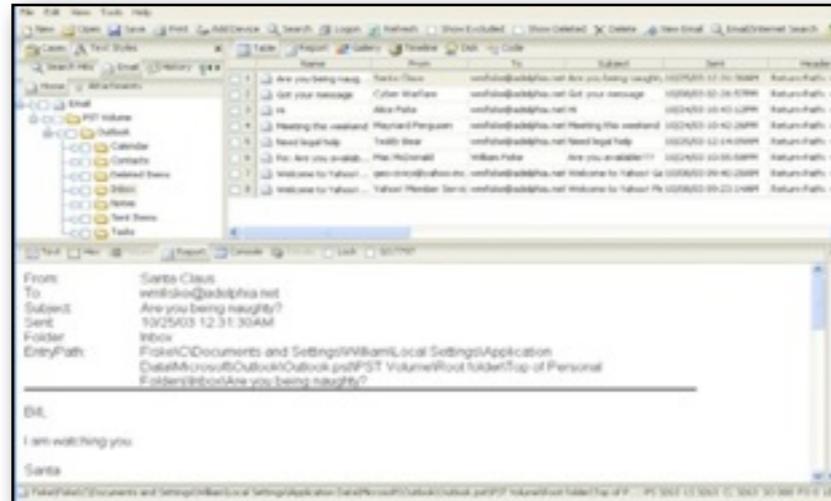
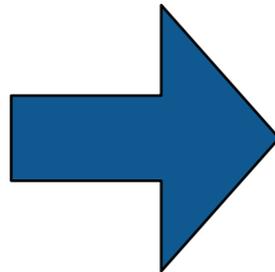


Emphasis on *building tools* and *working with practitioners*.



Creating Forensic Corpora

Digital forensics is at a turning point. Yesterday's work was primarily *reverse engineering*.



Key technical challenges:

- Evidence preservation.
- File recovery (file system support); Undeleting files
- Encryption cracking.
- Keyword search.

Today's work is increasingly *scientific*.

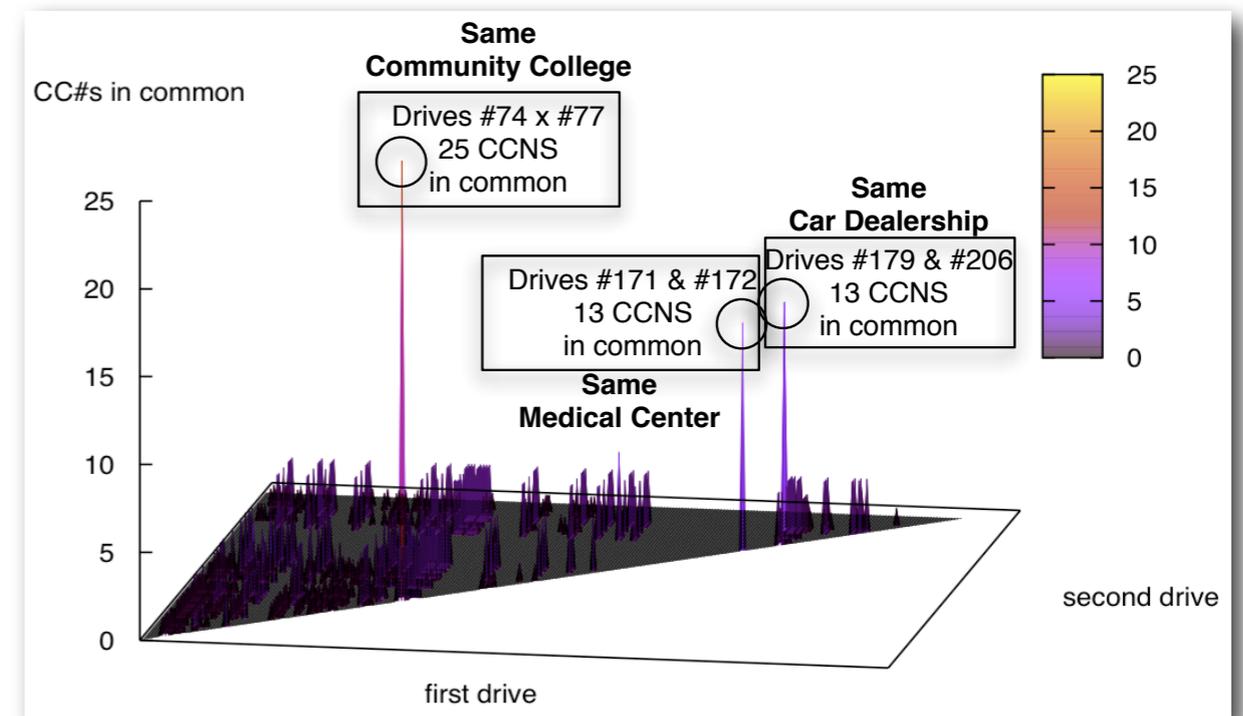
Evidence Reconstruction

- Files (fragment recovery carving)
- Timelines (visualization)

Clustering and data mining

Social network analysis

Sense-making



Science requires the *scientific process*.

Hallmarks of Science:

- Controlled and repeatable experiments.
- No privileged observers.

Why repeat some other scientist's experiment?

- Validate that an algorithm is properly implemented.
- Determine if ***your*** new algorithm is better than ***someone else's*** old one.
- (Scientific confirmation? — perhaps for venture capital firms.)



We can't do this today.

- People work with their own data
 - *Can't sure because of copyright & privacy issues.*
- People work with “evidence”
 - *Can't discuss due to legal sensitivities.*



Digital Forensics education needs corpora too!

To teach forensics, we need real data!

- Disk images
- Memory images
- Network packets



Some teachers get used hard drives from eBay.

- Problem: you don't know what's on the disk.
 - *Ground Truth.*
 - *Potential for illegal Material — distributing porn to minors is illegal.*



Some teachers have students examine other student machines:

- Self-examination: students know what they will find
- Examining each other's machines: potential for inappropriate disclosure

We manufacture data that can be freely redistributed.

Files from US Government Web Servers (500GB)

- \approx 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>

We manufacture data that can be freely redistributed.

Test and Realistic Disk Images (1TB)

- Mostly Windows operating system.
- Some with complex scenarios to facilitate forensics education.

—*NSF DUE-0919593*

University harassment scenario

- Network forensics — browser fingerprinting, reverse NAT, target identification.
- 50MB of packets

Company data theft & child pornography scenario.

- Multi-drive correction.
- Hypothesis formation.
- Timeline reconstruction.

—*Disk images, Memory Dumps, Network Packets*

We also acquire “real data” for use in research.

The Real Data Corpus (20TB)

- 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.)

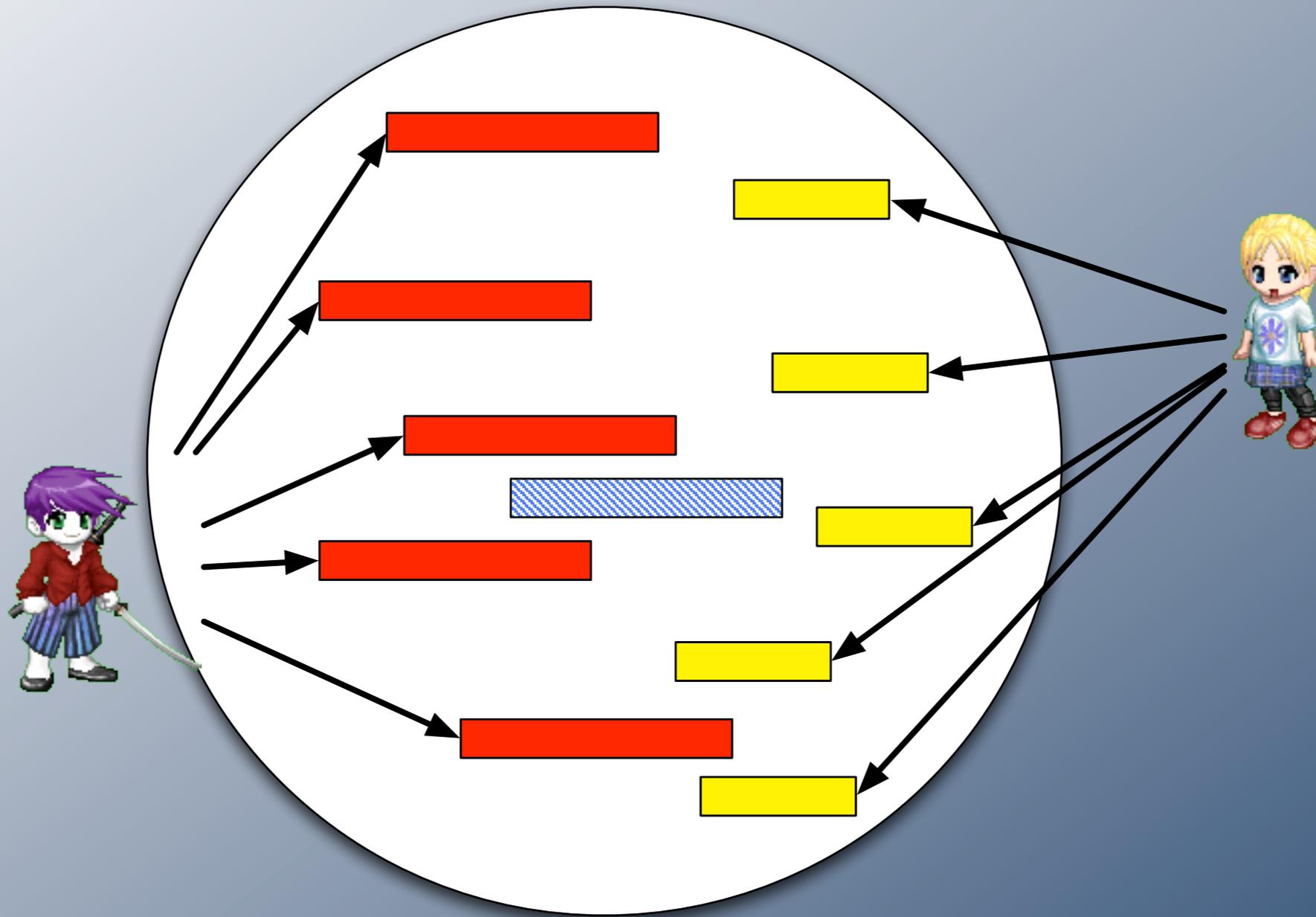


Mobile Phone Application Corpus

- Android Applications; Mobile Malware; etc.

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

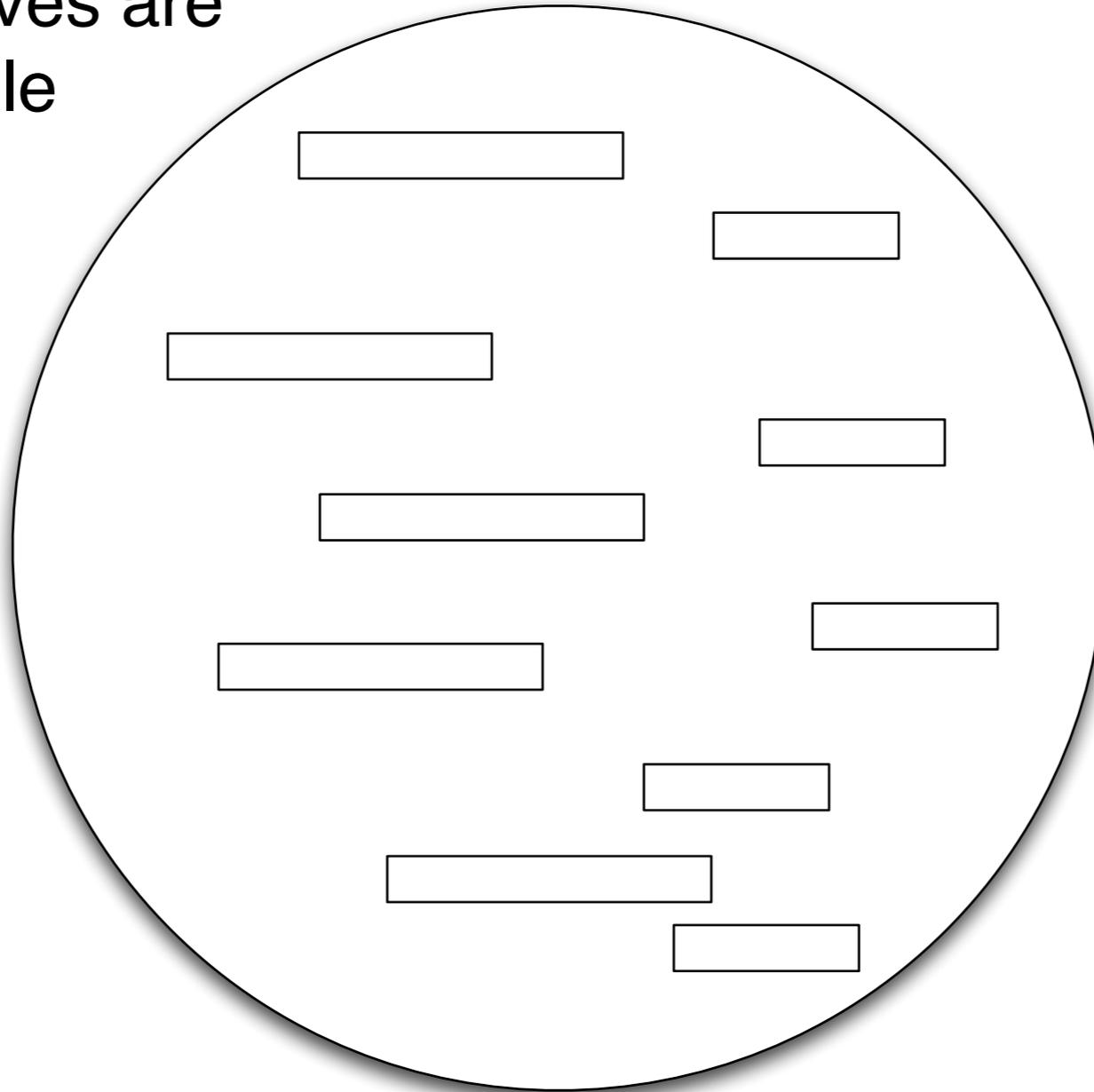
- *Garfinkel, Farrell, Roussev and Dinolt, Bringing Science to Digital Forensics with Standardized Forensic Corpora, DFRWS 2009*
<http://digitalcorpora.org/>



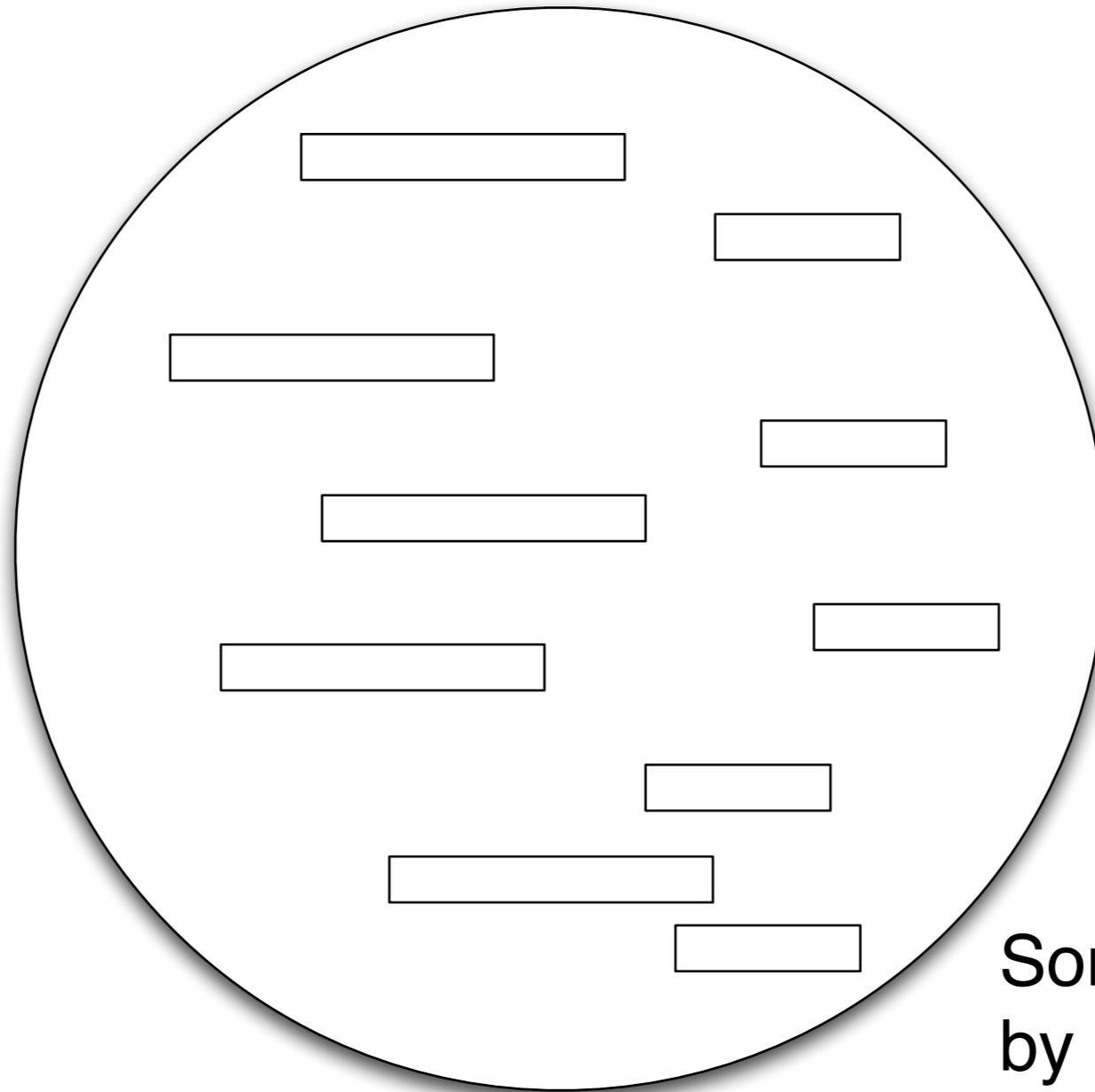
Automated Ascription of Multi-User Data

Disks may have any number of recoverable files.
0 to 1,000,000 is common.

Some hard drives are
used by a single
person.



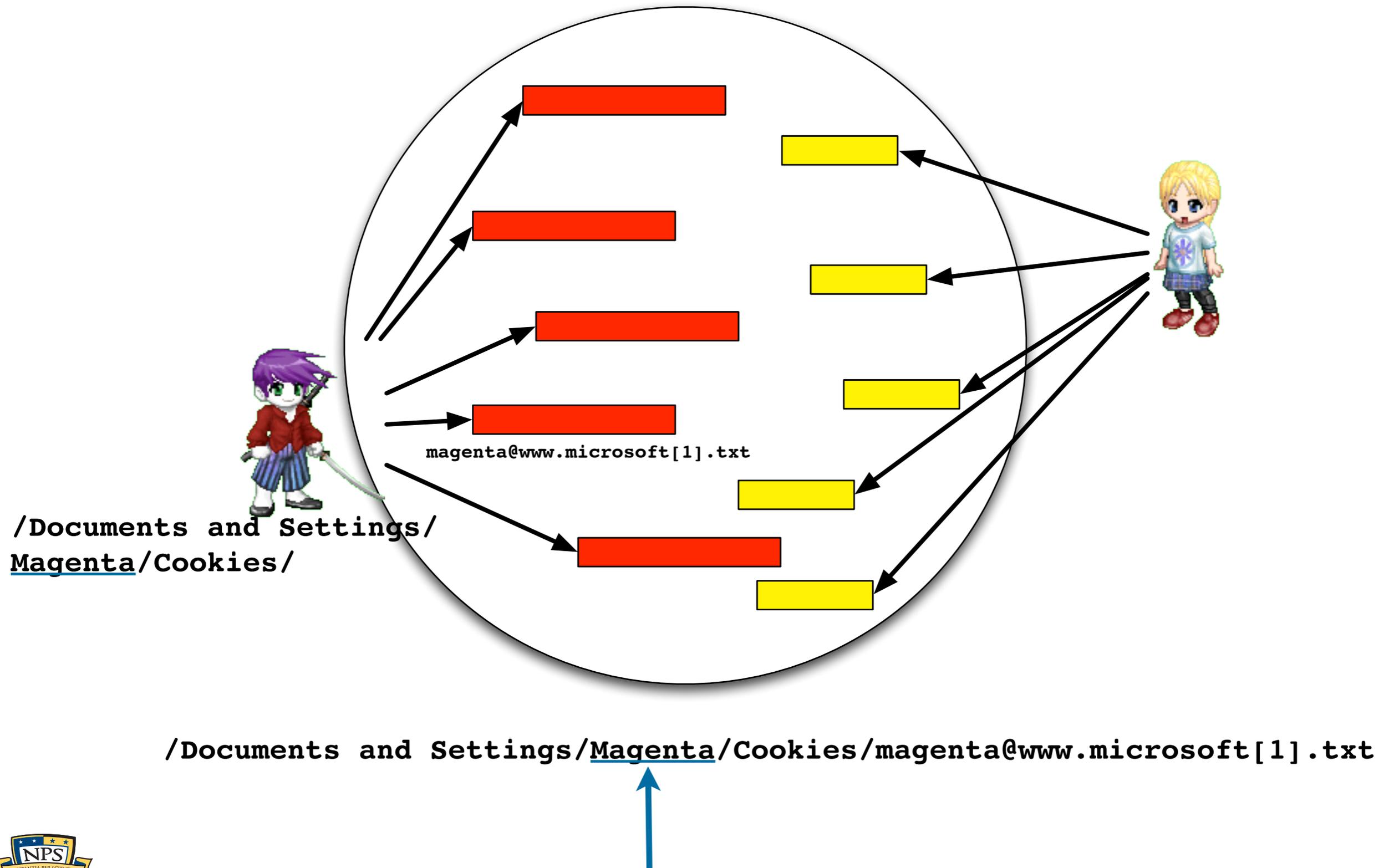
Disks may have any number of recoverable files.
0 to 1,000,000 is common.



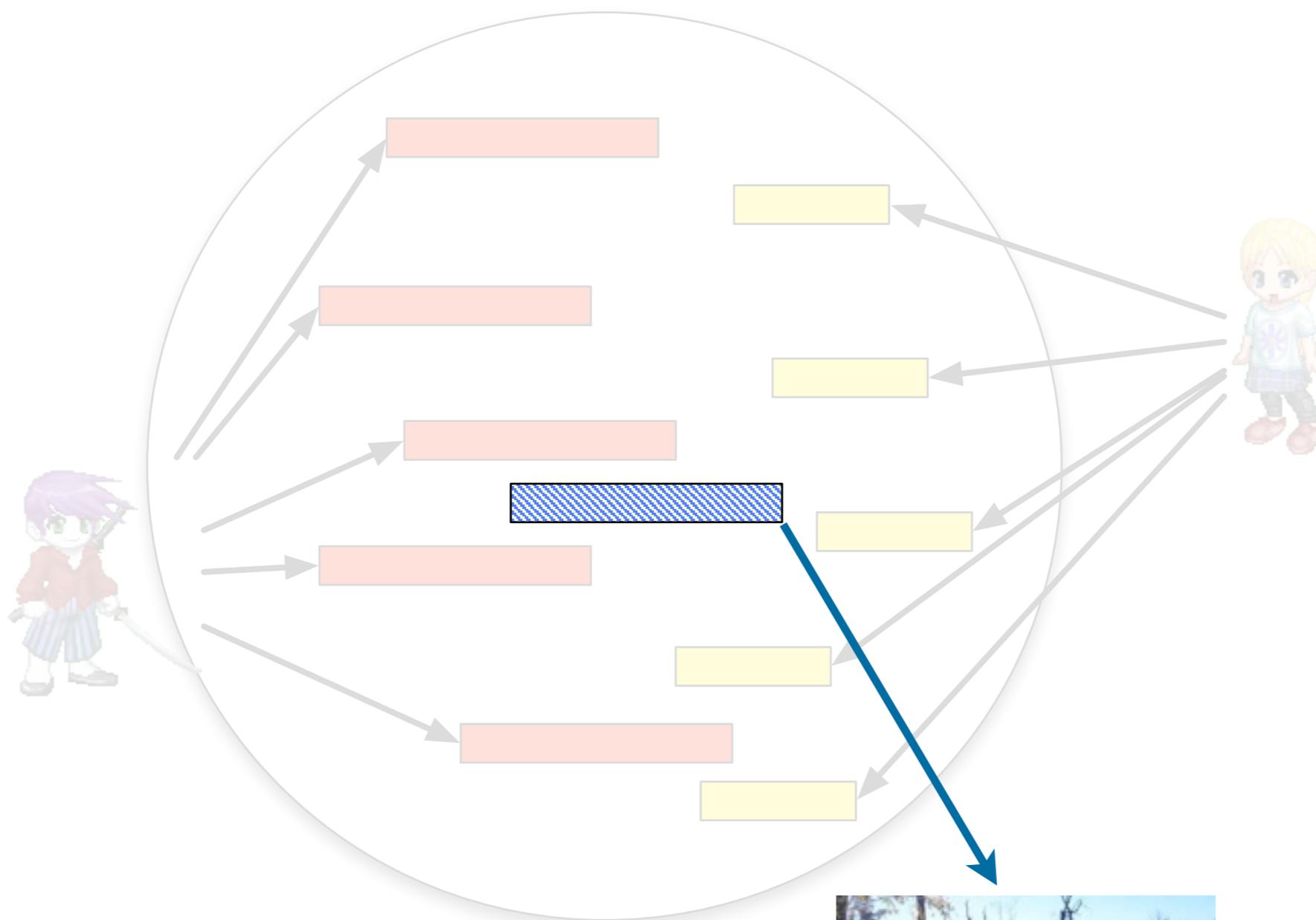
Some drives are used
by multiple people.



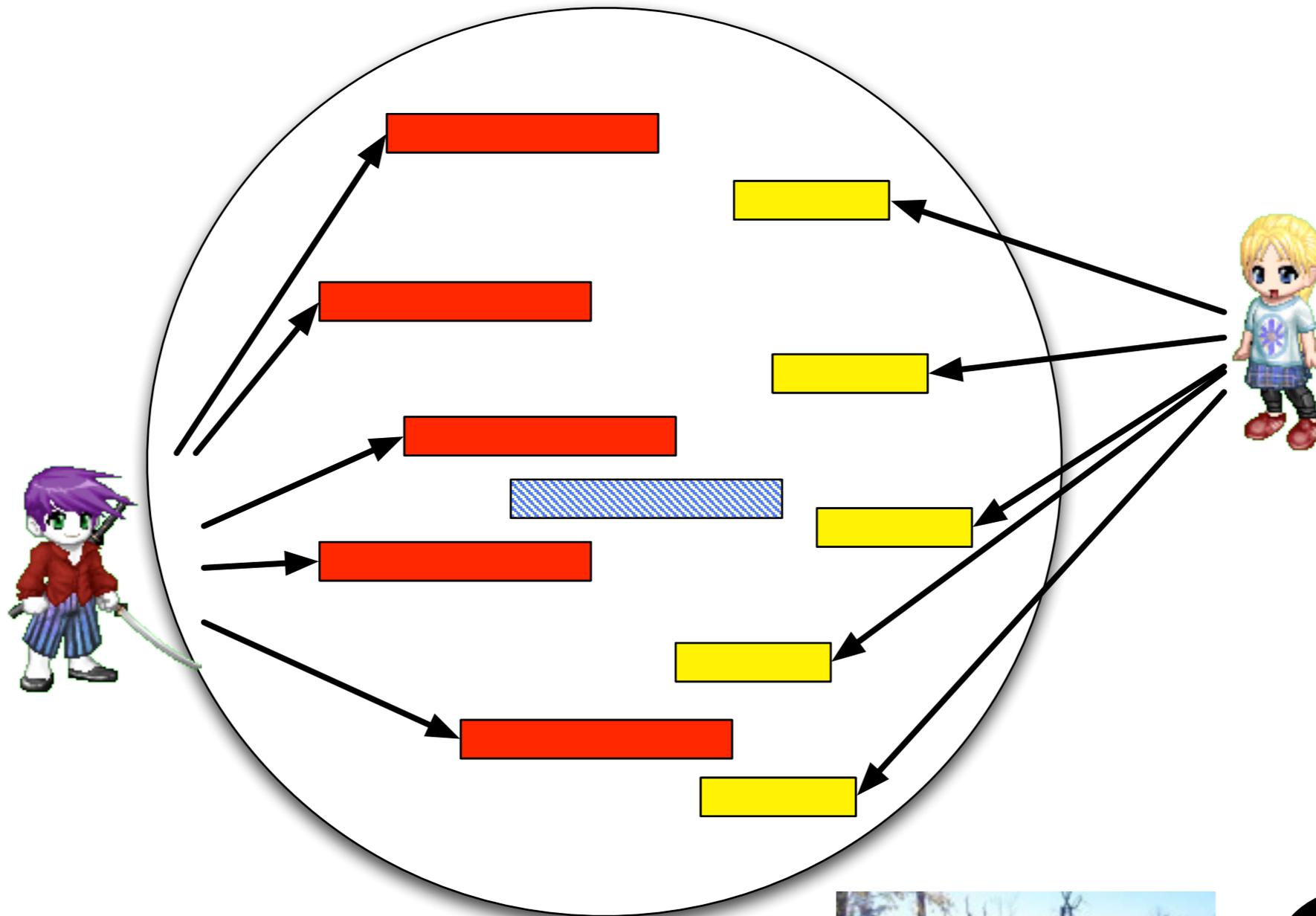
With multiple users, ownership is identified with metadata such as “path” or “file owner.”



“Carved” data is *residual data* on a disk.



Files recovered with “carving” can’t be readily ascribed.



Who is responsible for the file?



Prior work has used *content analysis* to determine authorship

Trait		
“Reading Level”	8 th Grade	College
Characteristic Errors	JUmp higher. FLy high.	Skilz Killz Spilz

This project uses metadata to infer *ownership or agency* — who is *responsible* for the data.

File system metadata (“extrinsic metadata”):

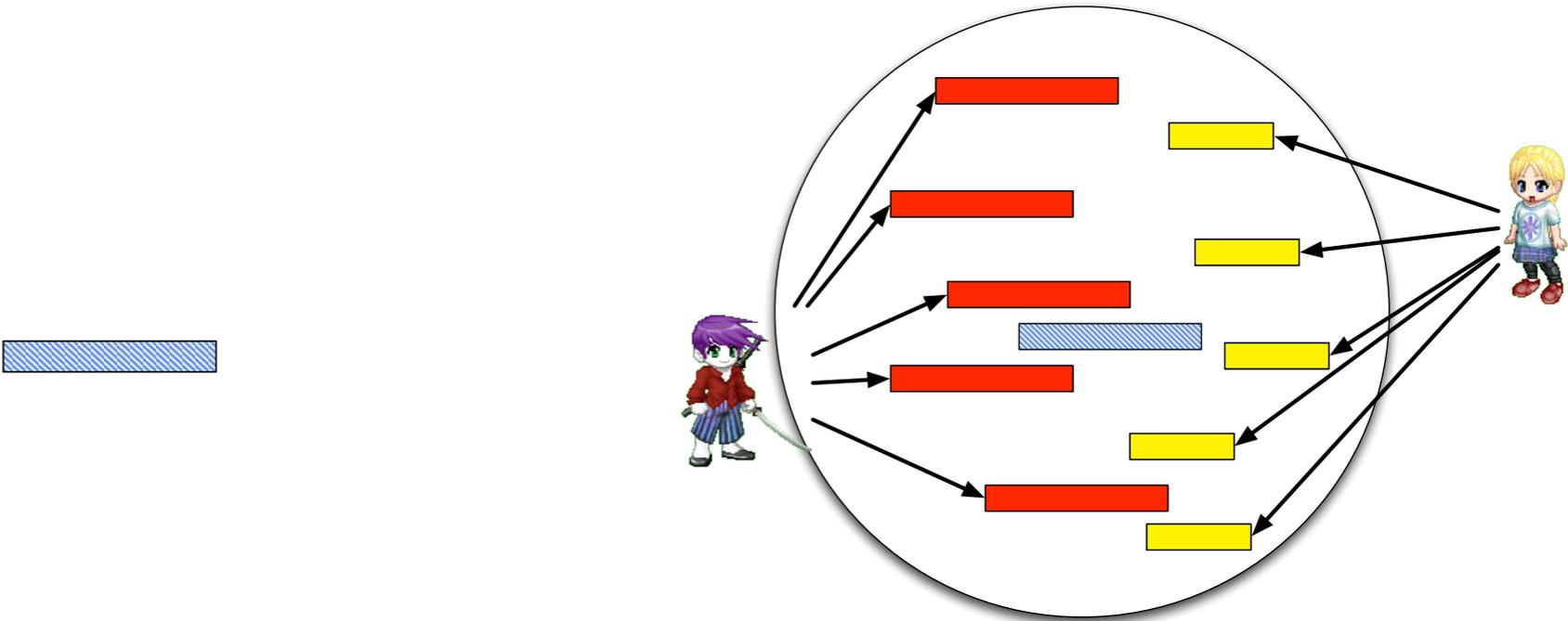
- Fragmentation patterns (disk usage)
- Where the file is on the hard drive (sector numbers)
- Timestamps for “orphan” files.

File metadata (“intrinsic metadata”):

- Embedded timestamps
 - *Creation Time*
 - *Print Time*
- Make & model of digital cameras
- Usage patterns.

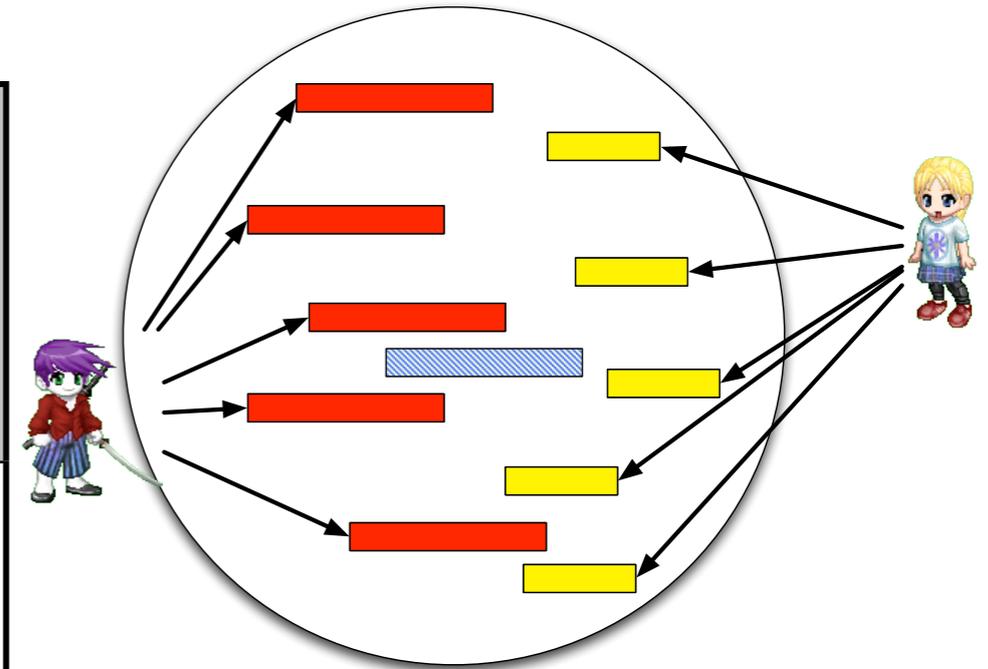


Today some examiners do this manually by surveying the disk for exemplars and looking for patterns.



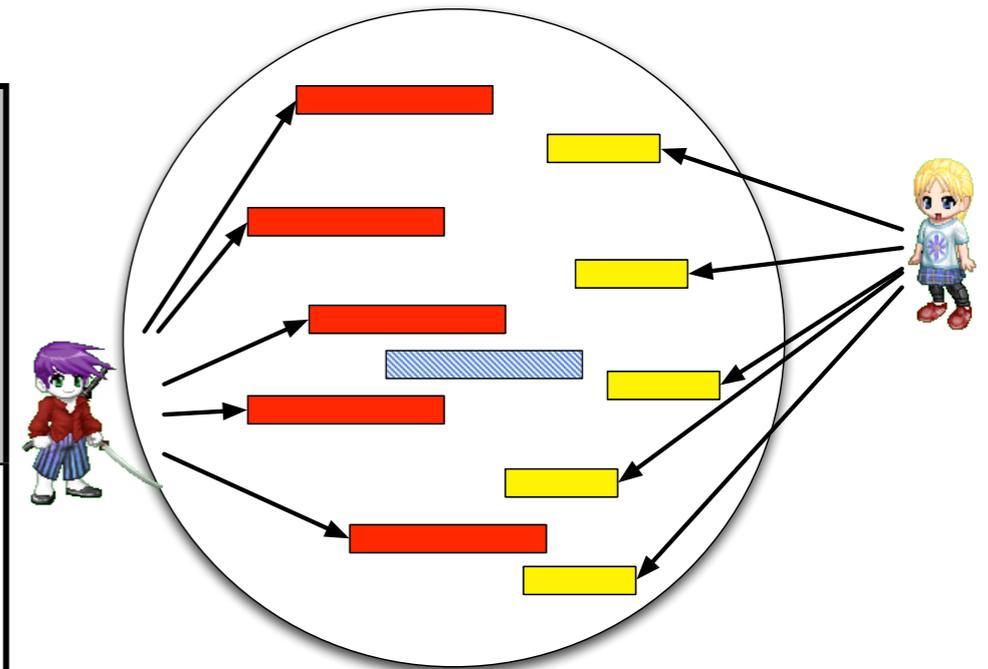
Today some examiners do this manually by surveying the disk for exemplars and looking for patterns.

Magenta	Yellow	Carved 	Likely User



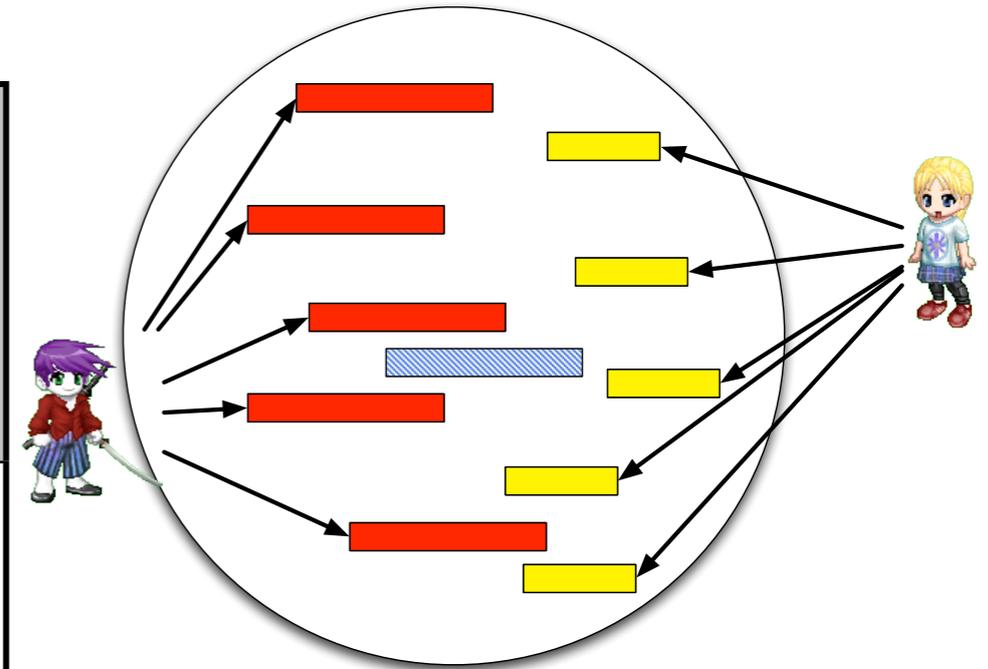
Today some examiners do this manually by surveying the disk for exemplars and looking for patterns.

Magenta	Yellow	Carved 	Likely User
100 JPEGs 5 DOCs	75 XLS 400 HTML		



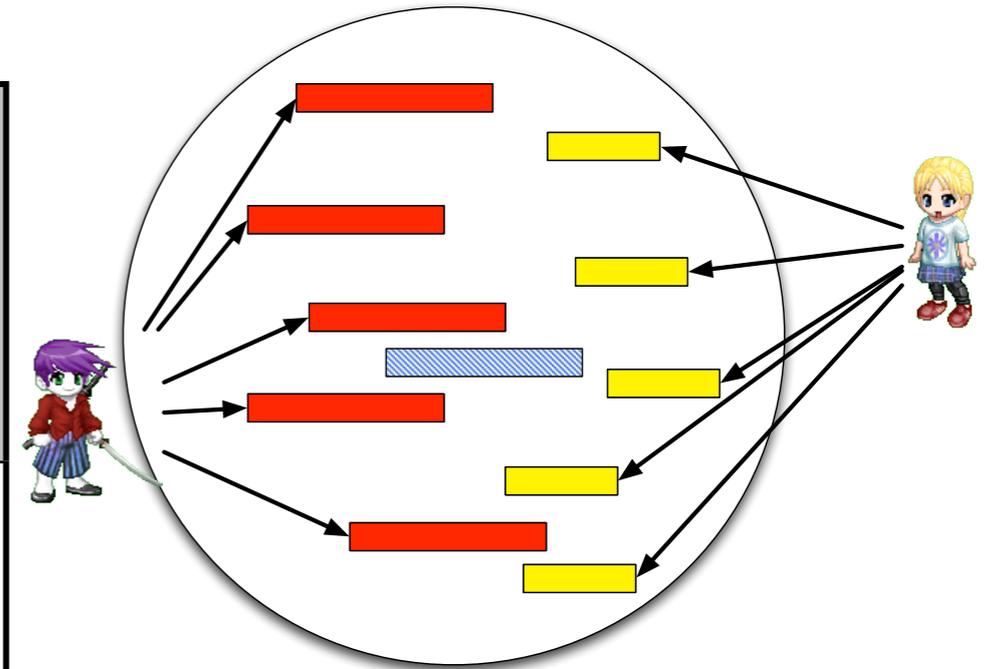
Today some examiners do this manually by surveying the disk for exemplars and looking for patterns.

Magenta	Yellow	Carved 	Likely User
100 JPEGs 5 DOCs	75 XLS 400 HTML	JPEG	



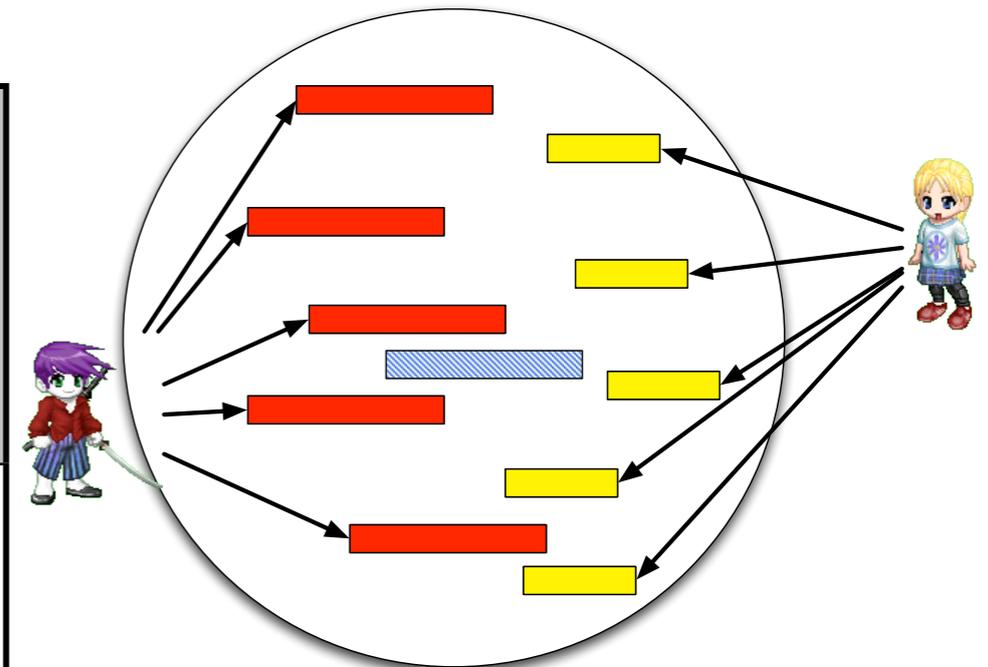
Today some examiners do this manually by surveying the disk for exemplars and looking for patterns.

Magenta	Yellow	Carved 	Likely User
100 JPEGs 5 DOCs	75 XLS 400 HTML	JPEG	



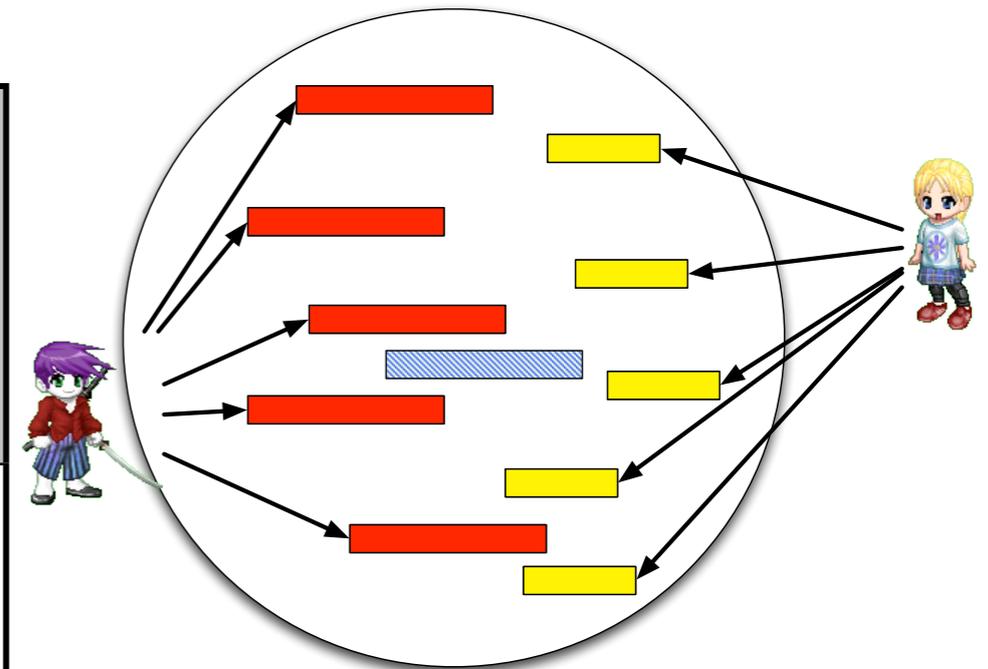
Today some examiners do this manually by surveying the disk for exemplars and looking for patterns.

Magenta	Yellow	Carved 	Likely User
100 JPEGs 5 DOCs	75 XLS 400 HTML	JPEG	
Printed 9am, 10am, 11am	Printed 8pm, 9pm		



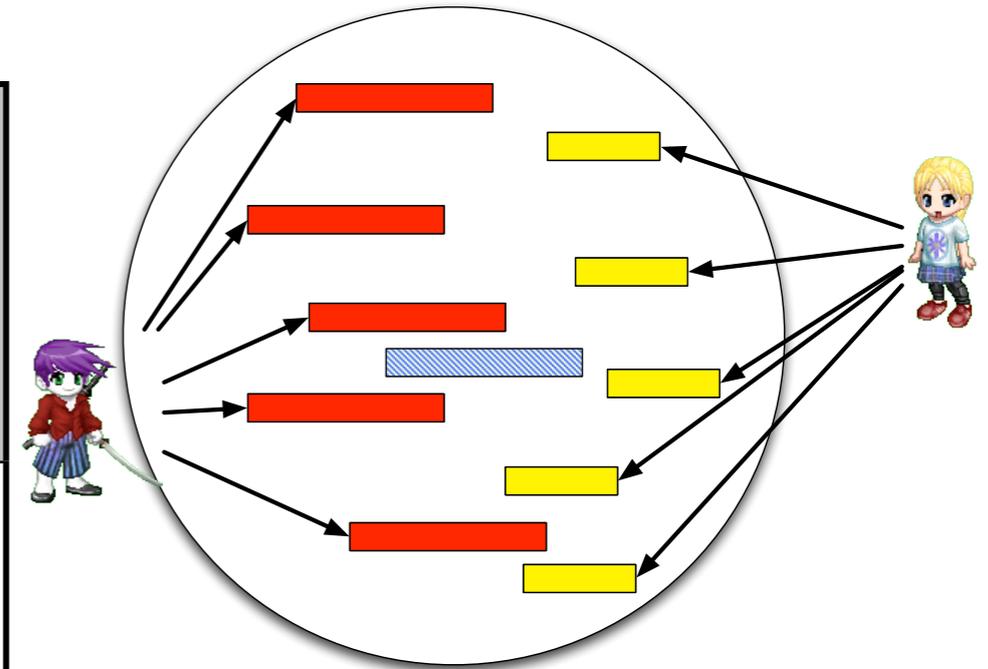
Today some examiners do this manually by surveying the disk for exemplars and looking for patterns.

Magenta	Yellow	Carved 	Likely User
100 JPEGs 5 DOCs	75 XLS 400 HTML	JPEG	
Printed 9am, 10am, 11am	Printed 8pm, 9pm	Printed 8:30pm	



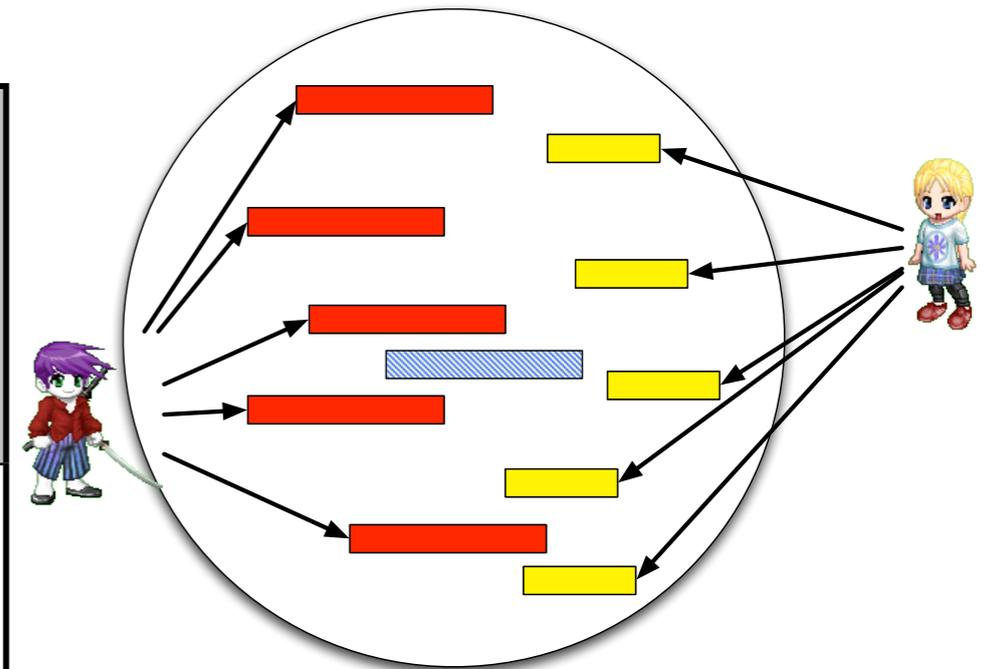
Today some examiners do this manually by surveying the disk for exemplars and looking for patterns.

Magenta	Yellow	Carved 	Likely User
100 JPEGs 5 DOCs	75 XLS 400 HTML	JPEG	
Printed 9am, 10am, 11am	Printed 8pm, 9pm	Printed 8:30pm	



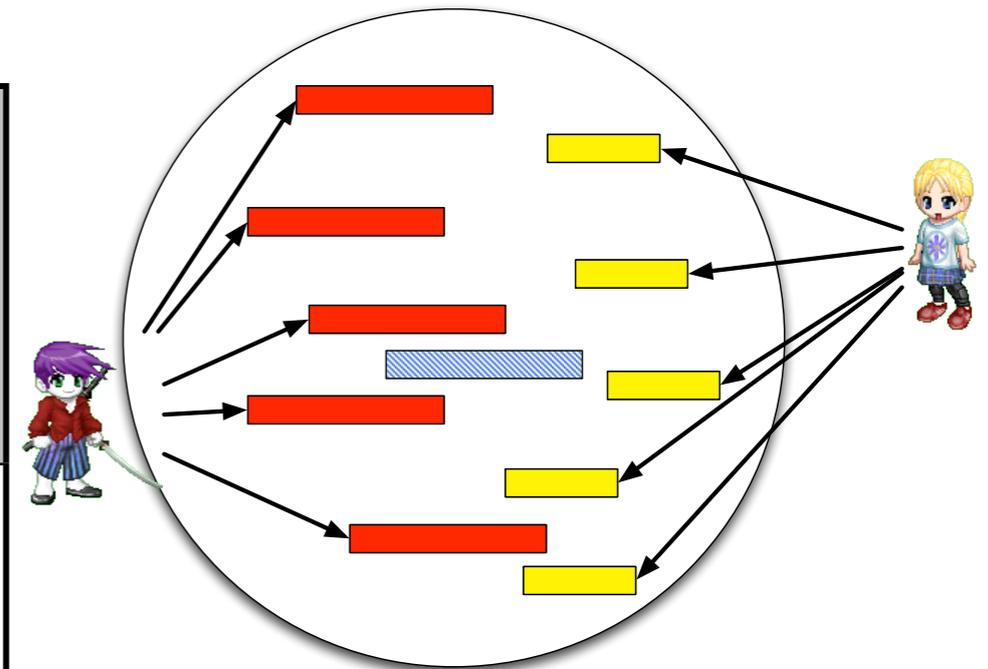
Today some examiners do this manually by surveying the disk for exemplars and looking for patterns.

Magenta	Yellow	Carved 	Likely User
100 JPEGs 5 DOCs	75 XLS 400 HTML	JPEG	
Printed 9am, 10am, 11am	Printed 8pm, 9pm	Printed 8:30pm	
At sectors 10, 20, 30	At sectors 500, 600, 700		



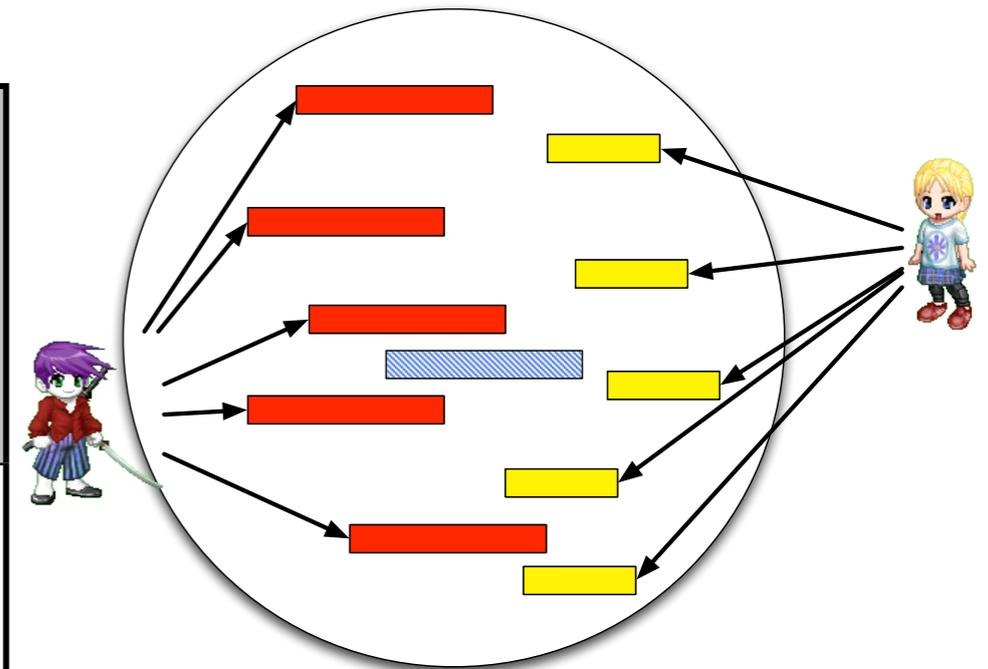
Today some examiners do this manually by surveying the disk for exemplars and looking for patterns.

Magenta	Yellow	Carved 	Likely User
100 JPEGs 5 DOCs	75 XLS 400 HTML	JPEG	
Printed 9am, 10am, 11am	Printed 8pm, 9pm	Printed 8:30pm	
At sectors 10, 20, 30	At sectors 500, 600, 700	Sector 550	

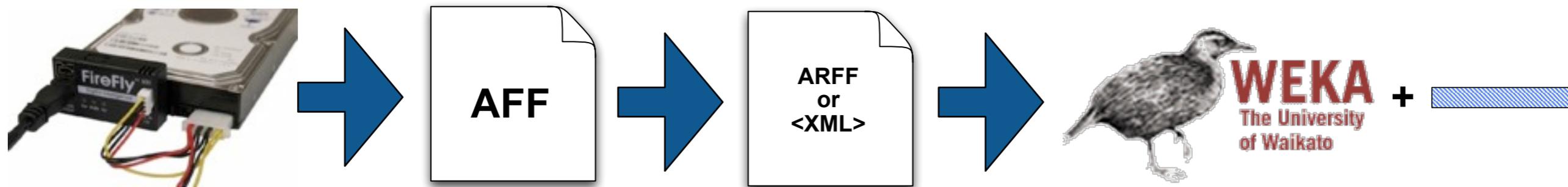


Today some examiners do this manually by surveying the disk for exemplars and looking for patterns.

Magenta	Yellow	Carved 	Likely User
100 JPEGs 5 DOCs	75 XLS 400 HTML	JPEG	
Printed 9am, 10am, 11am	Printed 8pm, 9pm	Printed 8:30pm	
At sectors 10, 20, 30	At sectors 500, 600, 700	Sector 550	



We developed an approach and tool set for *automated ascription*.



Step 1: Extract all files and file *metadata*

- File Owner (from filename or metadata)
- All files: Location on disk
- JPEGs: Camera Serial Number
- Word Documents: Author, Last Edit Time, Print Time, etc.



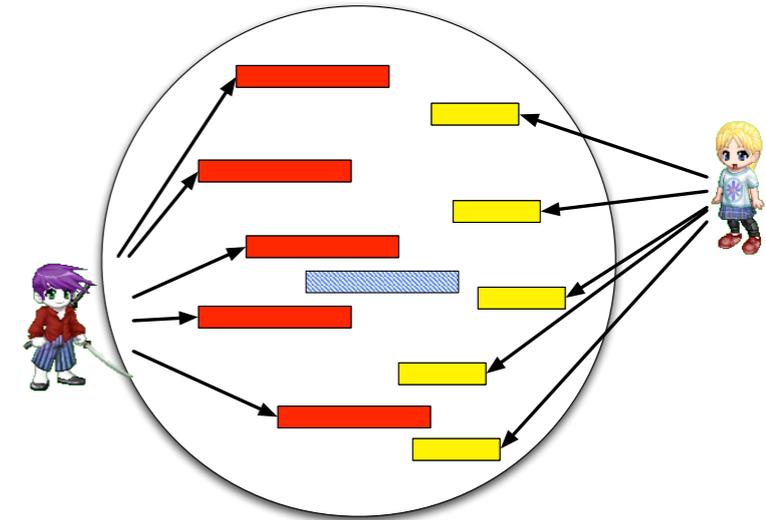
Step 2: Build a classifier using ascribable files as exemplars

Step 3: Use classifier to ascribe carved data.

Several factors complicate this data mining problem.

High dimensionality, heterogeneous data

- **All files:** *inode, mode, timestamps, sector #,*
- **JPEG:** *Serial Number, f-stop, exposure date*
- **Word:** *Author, Print Time, Create Time, etc.*



Sparse data; many missing values

- Every data element is missing values in one or more dimensions!

Multiple regions for each class

- User files interleave in time, space, etc.

Many different time dimensions

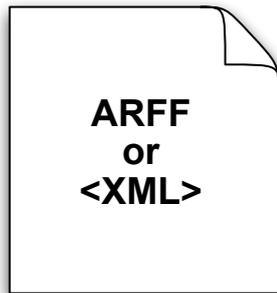
- File Print Time; File Modify Time; File Access Time; etc.
- Projecting times onto a "User Activity Timeline" dramatically improves accuracy.



fiwalk is our tool for converting disk images to XML or ARFF files

Per-Image tags

```
<fiwalk> – outer tag  
<fiwalk_version>0.4</fiwalk_version>  
<Start_time>Mon Oct 13 19:12:09 2008</Start_time>  
<Imagefile>dosfs.dmg</Imagefile>  
<volume startsector="512">
```



Per <volume> tags:

```
<Partition_Offset>512</Partition_Offset>  
<block_size>512</block_size>  
<ftype>4</ftype>  
<ftype_str>fat16</ftype_str>  
<block_count>81982</block_count>
```

Per <fileobject> tags:

```
<filesize>4096</filesize>  
<partition>1</partition>  
<filename>linedash.gif</filename>  
<libmagic>GIF image data, version 89a, 410 x 143</libmagic>
```

fiwalk has a pluggable metadata extraction system

Metadata extractors are specified in the *configuration file*

```
*.jpg    dgi    ../plugins/jpeg_extract
*.pdf    dgi    java -classpath plugins.jar Libextract_plugin
```

- *Currently the extractor is chosen by the file extension*
- *fiwalk runs the plugins in a different process*
- *We have designed a native Java interface that uses IPC and 1 process, but nobody wants to use it.*

Metadata extractors produce name:value pairs on STDOUT

```
Manufacturer: SONY
Model: CYBERSHOT
Orientation: top - left
```

fiwalk incorporates metadata into XML and ARFF:

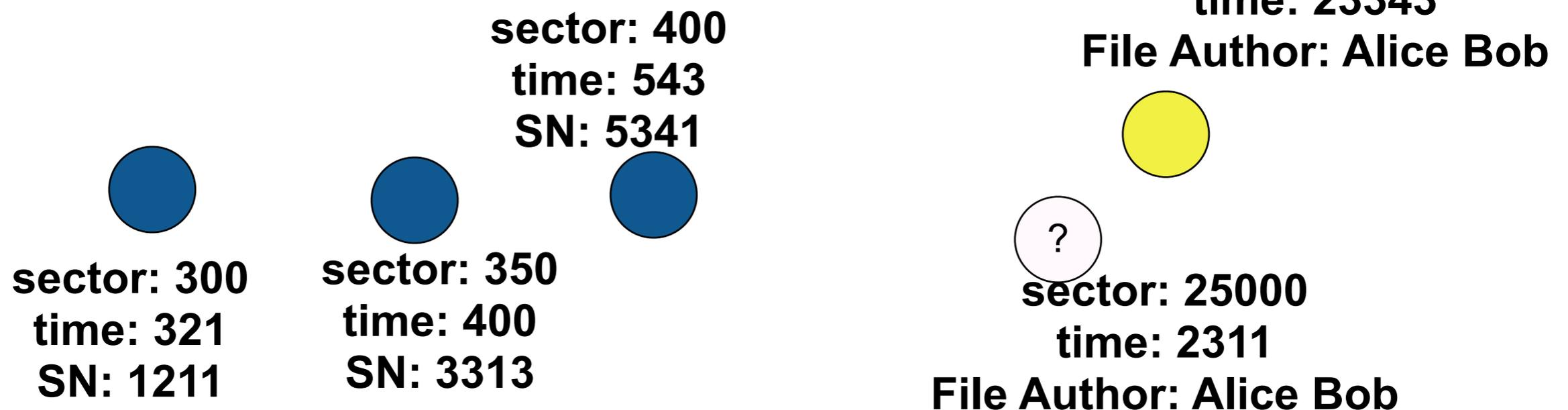
```
<fileobject>
...
<Manufacturer>SONY</Manufacturer>
<Model>CYBERSHOT</Model>
<Orientation>top - left</Orientation>
...
</fileobject>
```



Approach #1: K-Nearest-Neighbor

Special Features:

- N=1 works best (N=3 works pretty well)
- We had to create a special distance metric
 - *Nominal Data is distance 0 or 1.0*
 - *Time needs to be specially handled*
- Hypothesis:
 - *If there is a close exemplar, then that's the match.*



This approach is easy to explain to a jury!

Approach #2: Decision Tree

Algorithm: C4.5

- Very fast: typically less than 60 seconds.



```
|
| inode > 28455
| | inode <= 36552
| | | mode <= 365
| | | | inode <= 28892: magenta (132.0)
| | | | inode > 28892
| | | | | timeline <= 1225239807000: All Users (116.0)
| | | | | timeline > 1225239807000
| | | | | | frag1startsector <= 2585095
| | | | | | | libmagic = ASCII text, with CRLF line terminators
| | | | | | | | timeline <= 1225330086000: magenta (8.0)
| | | | | | | | timeline > 1225330086000: yellow (8.0)
| | | | | | | libmagic = data: magenta (16.0)
```

This approach generally provides higher accuracy than KNN.

- But the decision trees don't make as much sense (to a jury).

What do we mean by "accuracy?"

"Accuracy" means the probability that a classification is correct.

- We build a different classifier for every drive!
- So different drives will have different accuracy!

We determine the accuracy using take-one-out cross-validation.

- The only difference between allocated data and carved data is:
 - *Carved data is no longer attached to a directory.*
 - *Carved data is likely to be overwritten if the system is heavily used.*
- Take-one-out simply moves a file from the "allocated" set to the "carved" set.
- Each HD has its own accuracy.
- Each carved file has its own classifier
 - *Only use the relevant dimensions.*
 - *(e.g. don't use JPEG Camera SN if classifying a Word document.)*

A complex drive took 100,000+ runs on our cluster.

- A police department could run on Amazon EC2
- Other possibilities: gauge accuracy with sampling or 10-fold validation.

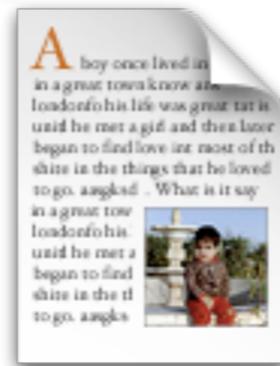


Key insight: User Activity Timeline

There are *many* sources of timestamps on a modern computer:

- File “MAC” times (Modify, Access, Create)

- Document Print Time (Microsoft Word)



- JPEG EXIF time (Time photo taken or edited.)



- Time email message authored, sent & received.



Timestamps that correspond to user activity on a single timeline.

- Significantly increases ascription accuracy
 - *Only for carved data that has embedded timestamps!*
- Not every timestamp can be used. (Digital Forensics is *messy*.)

Results with "realistic" drive created in the lab.

User	Classified As							total
	a	b	c	d	e	f	g	
a "Administrator"	5118	62	0	26	4	7	4	5221
b "All Users"	57	1422	17	32	12	4	0	1544
c "Default User"	1	39	392	0	0	0	4	436
d "domex1"	21	62	0	3051	96	0	0	3230
e "domex2"	24	16	0	94	2335	0	0	2469
f "LocalService"	12	0	0	0	0	64	0	76
g "NetworkService"	2	2	0	0	0	4	48	56
% correct classifications	97.77	88.71	95.84	95.25	95.42	81.01	85.71	

Table 7: domexusers (C4.5) Confusion matrix;

Real data performs better (multi-user drive purchased in Mexico.)

User	Classified As														total
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	
a "Administrator"	320	8	0	1	11	0	3	1	0	0	0	0	0	0	344
b "All Users"	13	971	0	3	11	0	15	7	0	18	3	1	24	2	1068
c (Blinded)	0	0	443	1	1	4	20	0	0	15	4	0	4	0	492
d (Blinded)	0	8	1	288	0	0	82	0	0	0	0	0	1	0	380
e "Default User"	8	36	4	0	343	1	4	0	4	0	4	0	0	0	404
f (Blinded)	0	0	12	0	1	440	4	0	3	0	0	0	0	0	460
g (Blinded)	2	8	22	46	6	11	66540	2	8	23	13	0	7	2	66690
h "LocalService"	5	2	0	0	0	0	4	75	0	0	0	0	2	0	88
i (Blinded)	0	0	1	0	1	0	0	0	707	0	2	0	1	0	712
j (Blinded)	0	12	4	2	0	1	22	1	0	594	0	0	0	0	636
k (Blinded)	0	3	4	0	1	0	8	0	0	0	1204	0	0	0	1220
l "NetworkService"	0	0	0	0	0	0	0	2	0	0	0	54	0	0	56
m (Blinded)	0	16	0	0	0	0	8	6	2	0	0	0	70952	224	71208
n (Blinded)	0	8	0	0	0	1	0	0	2	0	0	0	81	436	528
% correct classifications	91.95	90.58	90.22	84.46	91.47	96.07	99.75	79.79	97.38	91.38	97.89	98.18	99.83	65.66	

Table 9: 0844 (C4.5) Confusion matrix; non-system names are blinded.

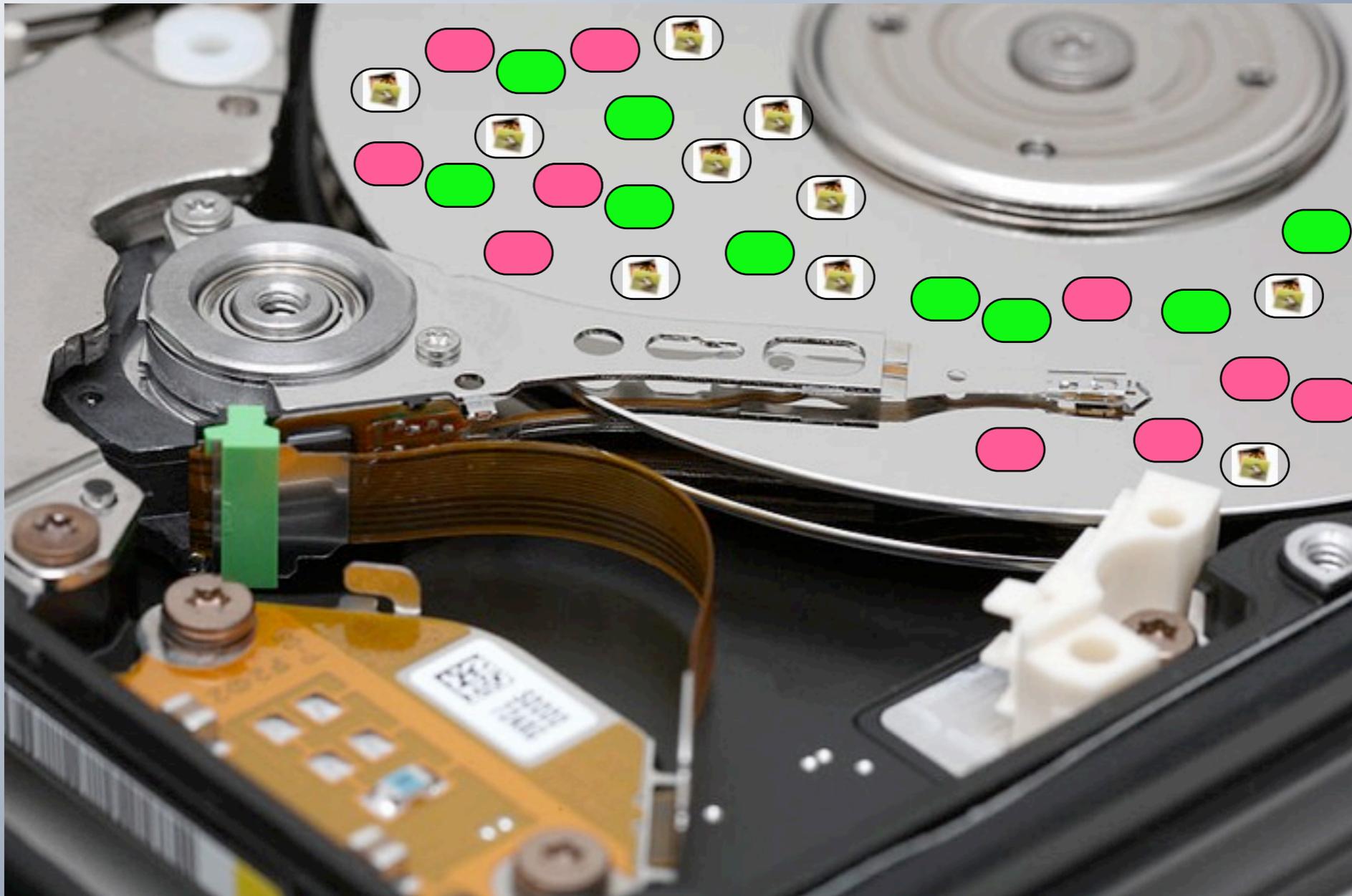
Publications

Student Theses:

- Cpt. Daniel Huynh, "Exploring and Validating Data Mining Algorithms for use in Data Ascription," June 2008
- Maj. James Migletz, "Automated Metadata Extraction," June 2008

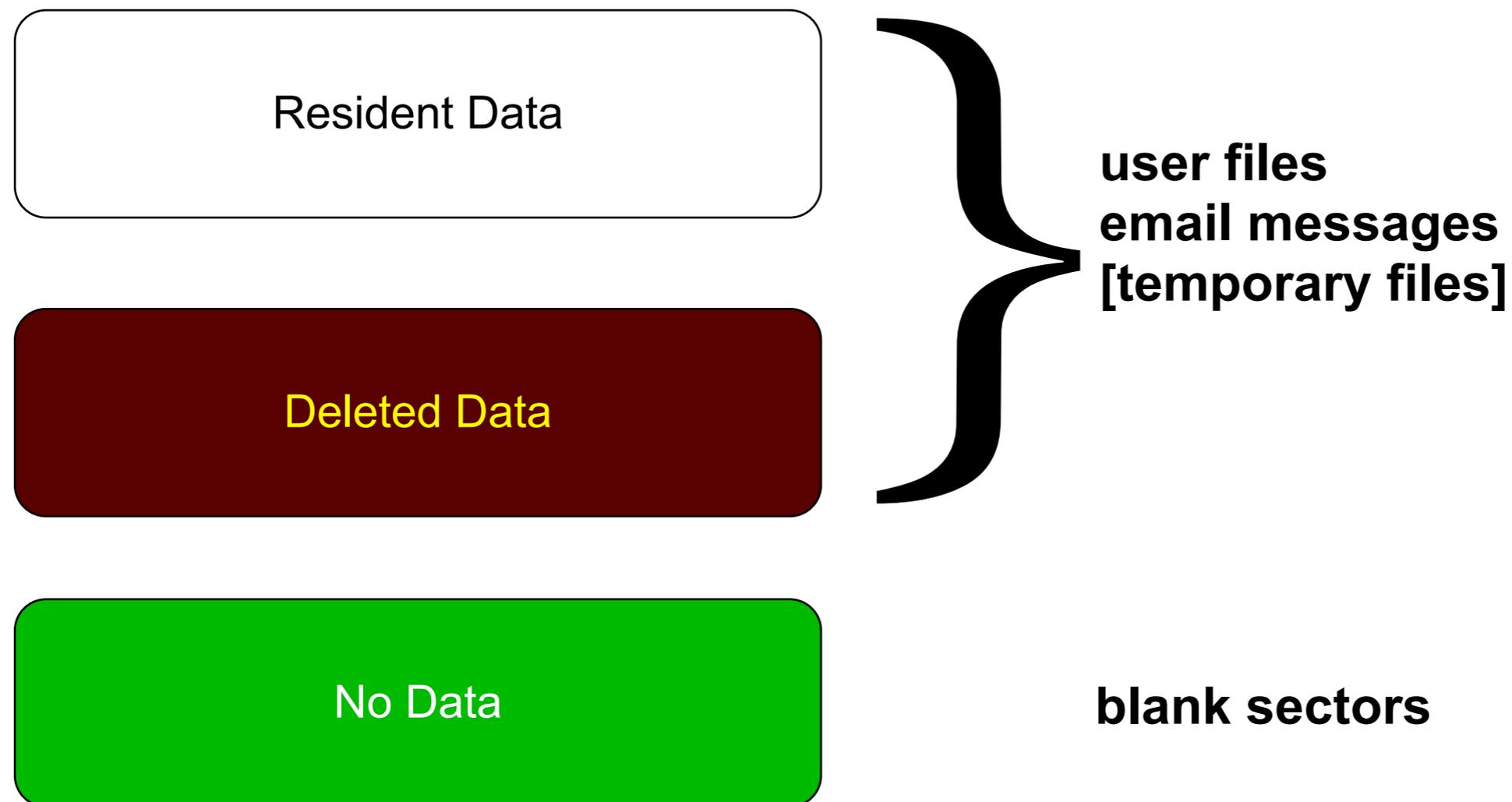
Articles:

- "An Automated Solution to the Multi-User Carved Data Ascription Problem,"
Simson L. Garfinkel, Aleatha Parker-Wood, Daniel Huynh and James Migletz,
IEEE Transactions on Information Forensics & Security, Dec. 2010

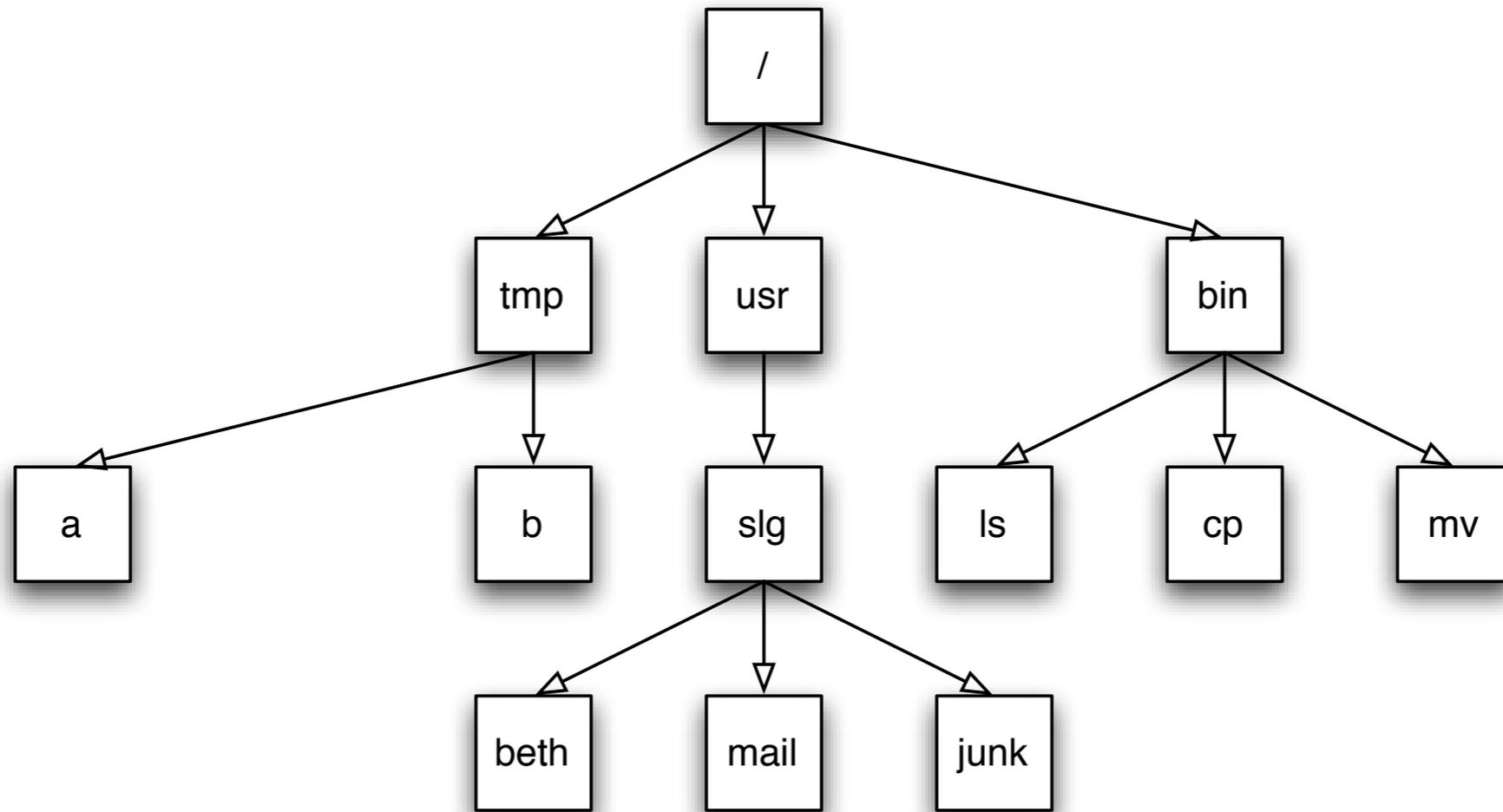


High Speed Forensics

Data on hard drives can be divided into three categories:

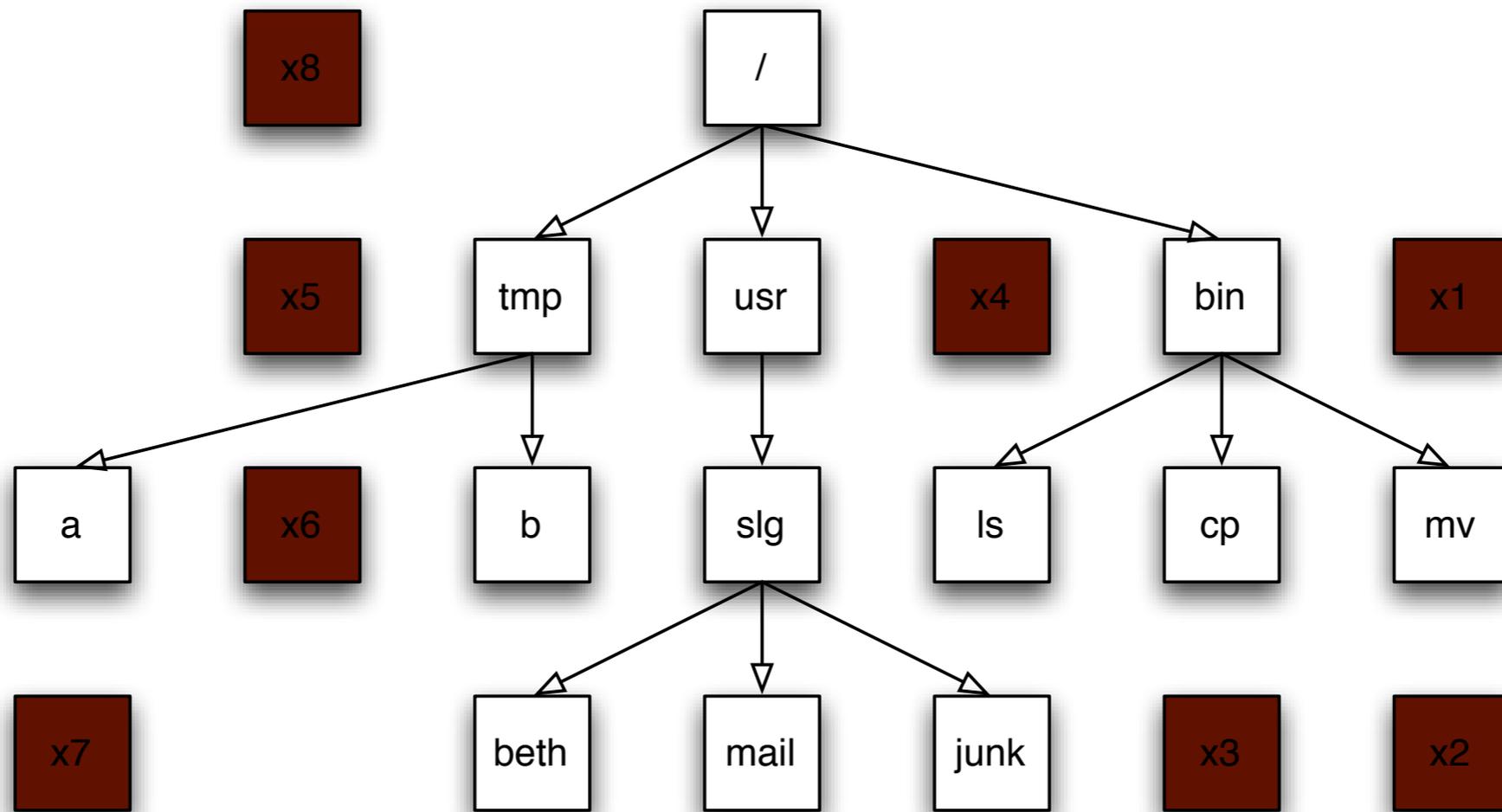


Resident data is the data you see from the root directory.



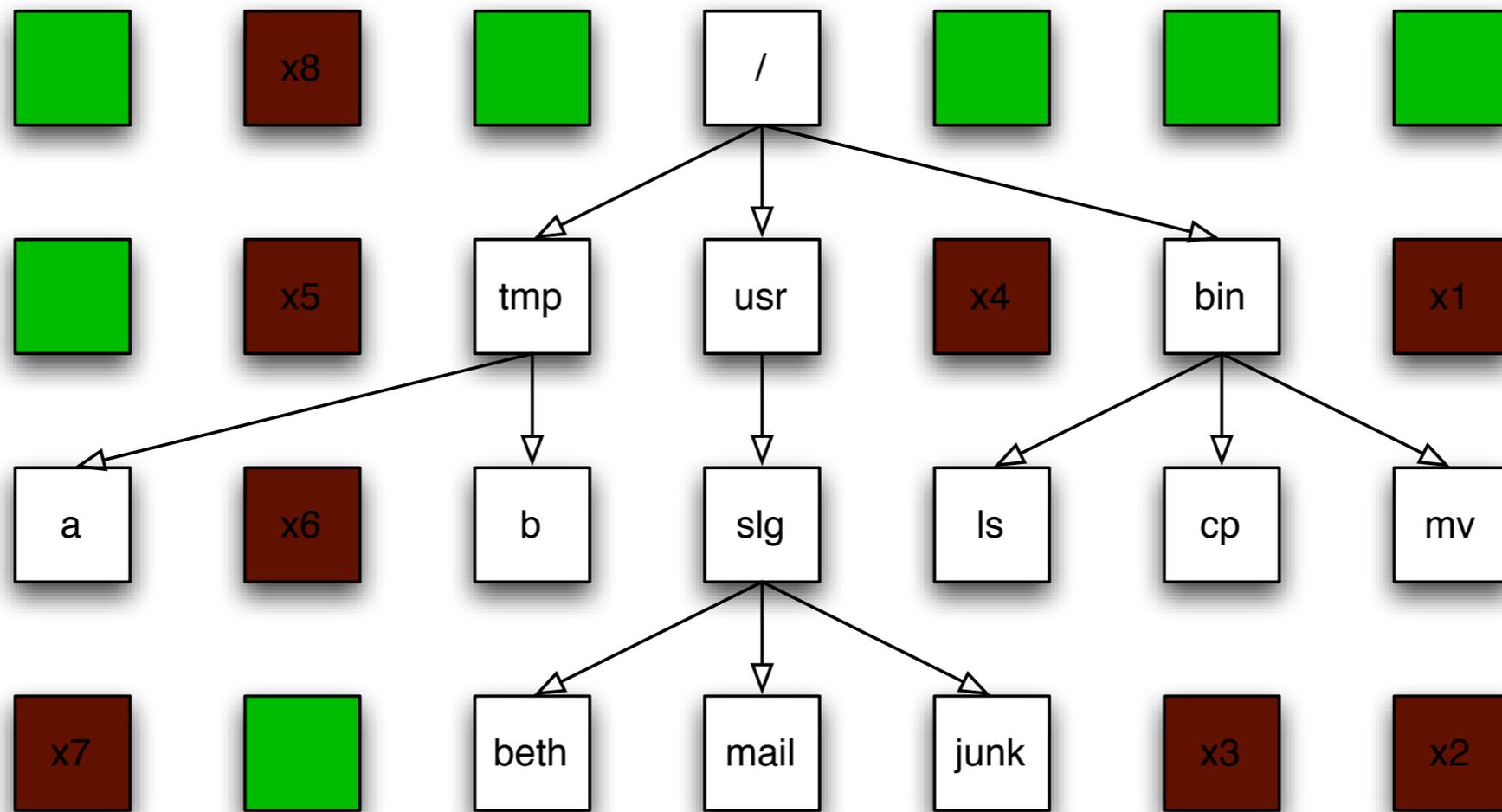
Resident Data

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



Deleted Data

Sectors with "No Data" are blank.



No Data

Today most forensic tools follow the same steps to analyze a disk drive.

Walk the file system to map out all the files (allocated & deleted).

For each file:

- Seek to the file.
- Read the file.
- Hash the file (MD5)
- Index file's text.

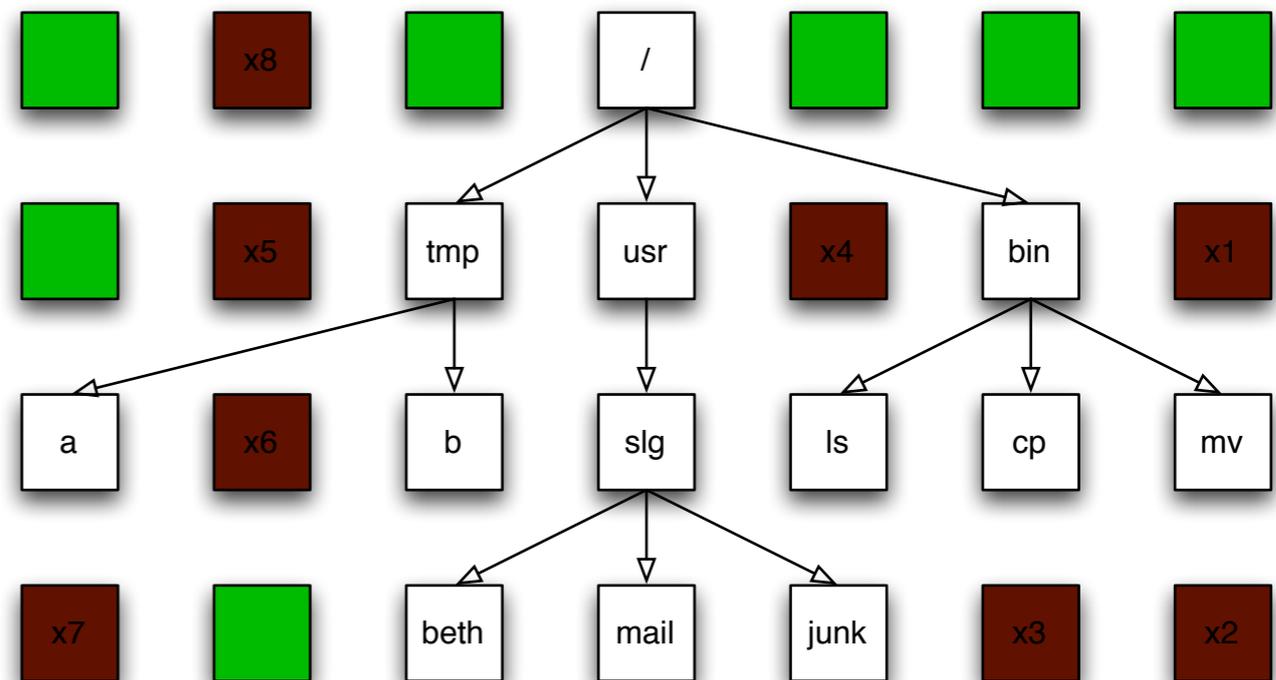
"Carve" space between files for other documents, text, etc.

Problem #1: Time

- 1TB drive takes 3.5 hours to read
— *10-80 hours to process!*

Problem #2: Completeness

- Lots of residual data is ignored.
— *Many investigations don't carve!*



Stream-Based Disk Forensics:

Scan the disk from beginning to end; do your best.

1. Read all of the blocks in order.
2. Look for information that might be useful.
3. Identify & extract what's possible in a single pass.

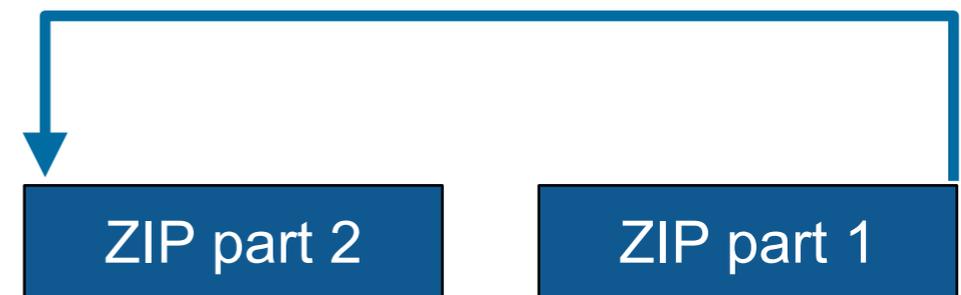
Advantages:

- No disk seeking.
- Read the disk at maximum transfer rate.
- Reads *all the data* — allocated files, deleted files, file fragments.



Disadvantages:

- Fragmented files won't be recovered:
 - *Compressed files with part2-part1 ordering*
 - *Files with internal fragmentation (.doc)*
- A second pass may be needed to map contents to file names.

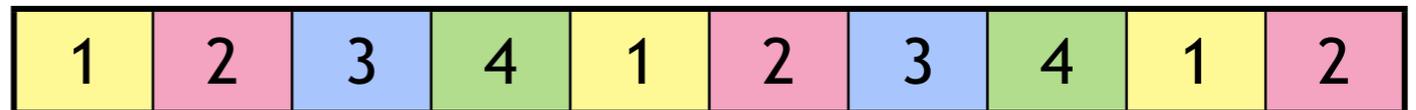


bulk_extractor: a high-speed disk scanner.

Written in C, C++ and Flex.

Key Features:

- Uses regular expressions and rules to scan for:
 - *email addresses; credit card numbers; JPEG EXIFs; URLs; Email fragments.*
- Recursively re-analyzes ZIP components.
- Produces a histogram of the results.
- Multi-threaded.
 - *Disk is "striped" and then the results are combined with merge sort.*



Challenges:

- Must work with evidence files of *any size* and on *limited hardware*.
- Users can't provide their data when the program crashes.
- Users are *analysts* and *examiners*, not engineers.

bulk_extractor output: text files of "features" and context.

email addresses from domexusers:

48198832	domexuser2@gmail.com	to: <<name>domexuser2@gmail.com/Home</name>>
48200361	domexuser2@live.com	to: <<name>domexuser2@live.com</name>> <pass
48413829	siege@preoccupied.net	siege) O'Brien <siege@preoccupied.net>_hp://meanwhi
48481542	daniilo@gnome.org	Daniilo __egan <daniilo@gnome.org>_Language-Team:
48481589	gnom@prevod.org	: Serbian (sr) <gnom@prevod.org>_MIME-Version:
49421069	domexuser1@gmail.com	server2.name", "domexuser1@gmail.com");__user_pref("
49421279	domexuser1@gmail.com	er2.userName", "domexuser1@gmail.com");__user_pref("
49421608	domexuser1@gmail.com	tp1.username", "domexuser1@gmail.com");__user_pref("

Histogram:

n=579	domexuser1@gmail.com
n=432	domexuser2@gmail.com
n=340	domexuser3@gmail.com
n=268	ips@mail.ips.es
n=252	premium-server@thawte.com
n=244	CPS-requests@verisign.com
n=242	someone@example.com

bulk_extractor success:

City of San Luis Obispo Police Department, Spring 2010

District Attorney filed charges against two individuals:

- Credit Card Fraud
- Possession of materials to commit credit card fraud.



Defendants:

- arrested with a computer.
- Expected to argue that defends were unsophisticated and lacked knowledge.



Examiner given 250GiB drive *the day before preliminary hearing.*

- In 2.5 hours Bulk Extractor found:
 - *Over 10,000 credit card numbers on the HD (1000 unique)*
 - *Most common email address belonged to the primary defendant (possession)*
 - *The most commonly occurring Internet search engine queries concerned credit card fraud and bank identification numbers (intent)*
 - *Most commonly visited websites were in a foreign country whose primary language is spoken fluently by the primary defendant.*
- Armed with this data, the DA was able to have the defendants held.

Eliminating false positives: Many of the email addresses come with Windows!

Sources of these addresses:

- Windows binaries
- SSL certificates
- Sample documents

n=579	domexuser1@gmail.com
n=432	domexuser2@gmail.com
n=340	domexuser3@gmail.com
n=268	ips@mail.ips.es
n=252	premium-server@thawte.com
n=244	CPS-requests@verisign.com
n=242	someone@example.com

It's important to suppress email addresses not relevant to the case.

Approach #1 — Suppress emails seen on many other drives.

Approach #2 — Stop list from bulk_extractor run on clean installs.

Both of these methods *white list* commonly seen emails.

- A problem — Operating Systems have a LOT of emails. (FC12 has 20,584!)
- Should we give the Linux developers a free pass?

Approach #3: Context-sensitive stop list.

Instead of extracting just the email address, extract the context:

- Offset: **351373329**
- Email: **zeeshan.ali@nokia.com**
- Context: **ut_Zeeshan Ali <zeeshan.ali@nokia.com>, Stefan Kost <**

- Offset: **351373366**
- Email: **stefan.kost@nokia.com**
- Context: **>, Stefan Kost <stefan.kost@nokia.com>_____sin**

Here "context" is 8 characters on either side of feature.

We created a context-sensitive stop list for Microsoft Windows XP, 2000, 2003, Vista, and several Linux.

Total stop list: 70MB (628,792 features)

Applying it to domexusers HD image:

- # of emails found: 9143 → 4459

without stop list

n=579 domexuser1@gmail.com
n=432 domexuser2@gmail.com
n=340 domexuser3@gmail.com
n=268 ips@mail.ips.es
n=252 premium-server@thawte.com
n=244 CPS-requests@verisign.com
n=242 someone@example.com
n=237 inet@microsoft.com
n=192 domexuser2@live.com
n=153 domexuser2@hotmail.com
n=146 domexuser1@hotmail.com
n=134 domexuser1@live.com
n=115 example@passport.com
n=115 myname@msn.com
n=110 ca@digsigtrust.com

with stop list

n=579 domexuser1@gmail.com
n=432 domexuser2@gmail.com
n=340 domexuser3@gmail.com
n=192 domexuser2@live.com
n=153 domexuser2@hotmail.com
n=146 domexuser1@hotmail.com
n=134 domexuser1@live.com
n=91 premium-server@thawte.com
n=70 talkback@mozilla.org
n=69 hewitt@netscape.com
n=54 DOMEXUSER2@GMAIL.COM
n=48 domexuser1%40gmail.com@imap.gmail.com
n=42 domex2@rad.li
n=39 lord@netscape.com
n=37 49091023.6070302@gmail.com

Can we analyze a hard drive in five minutes?

US agents encounter a hard drives at a border crossings...



Searches turn up rooms filled with servers....



If it takes 3.5 hours to read a 1TB hard drive, what can you learn in 5 minutes?

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

36 GB is a lot of data!

- $\approx 2.4\%$ of the disk
- But it can be a *statistically significant sample*.

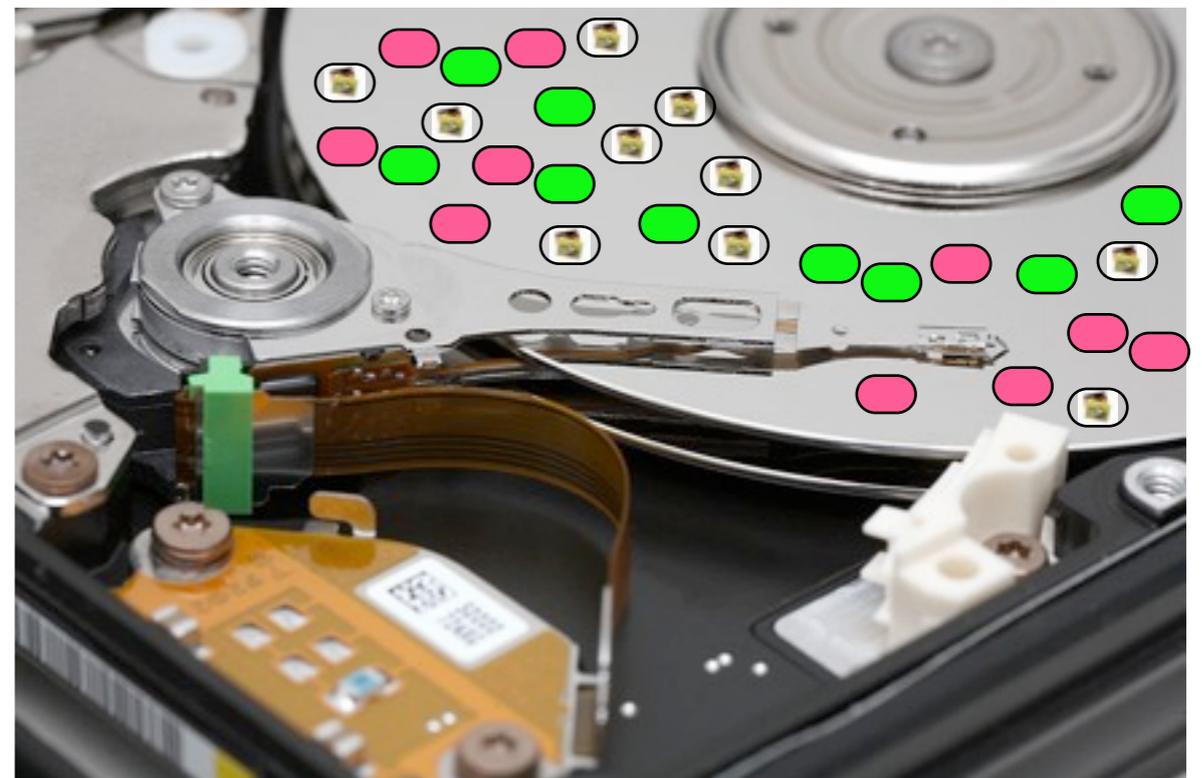
We can predict the statistics of a *population* by sampling a *randomly chosen sample*.

US elections can be predicted by sampling a few thousand households:



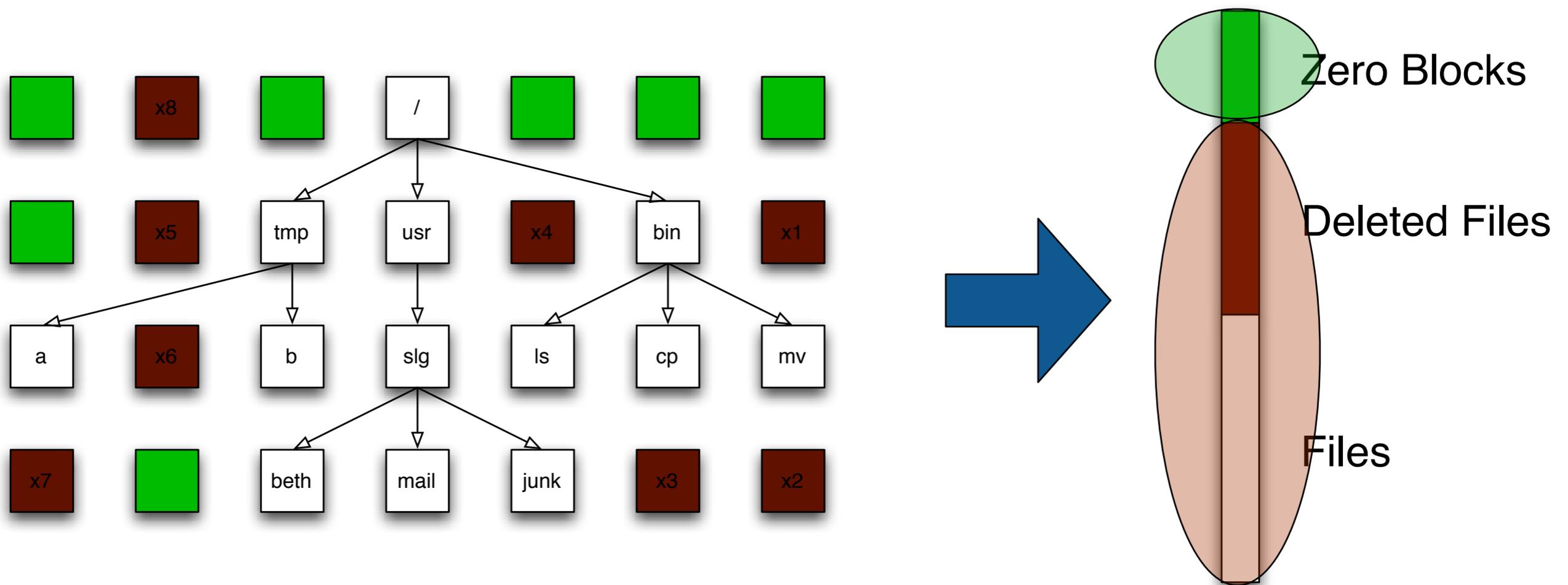
The challenge is identifying *likely voters*.

Hard drive contents can be predicted by sampling a few thousand sectors:



The challenge is *identifying the sectors* that are sampled.

Sampling can distinguish between "zero" and data. It can't distinguish between resident and deleted.



Simplify the problem.

Can we use statistical sampling to verify wiping?

Many organizations discard used computers.

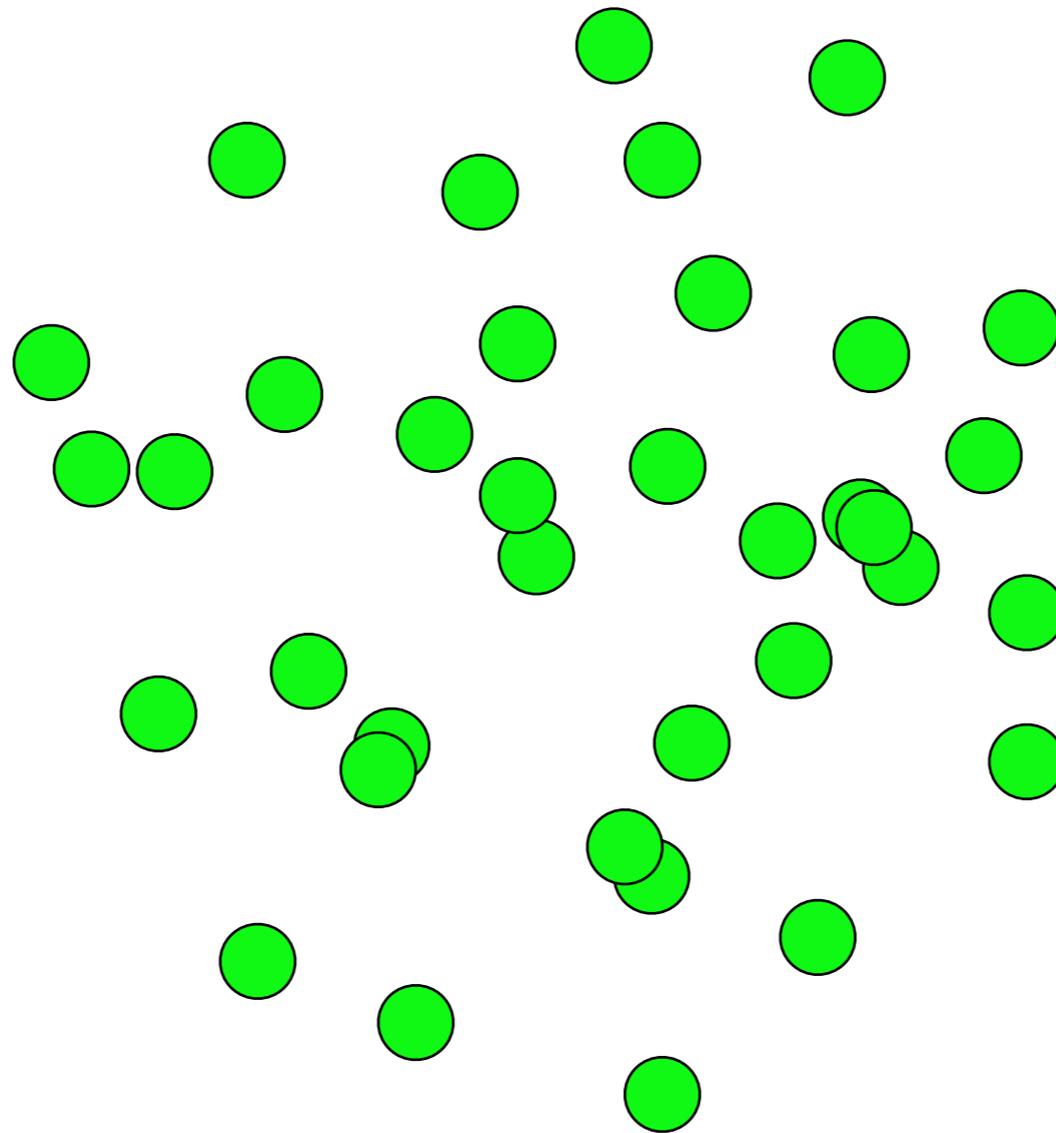
Can we verify if a disk is properly wiped in 5 minutes?



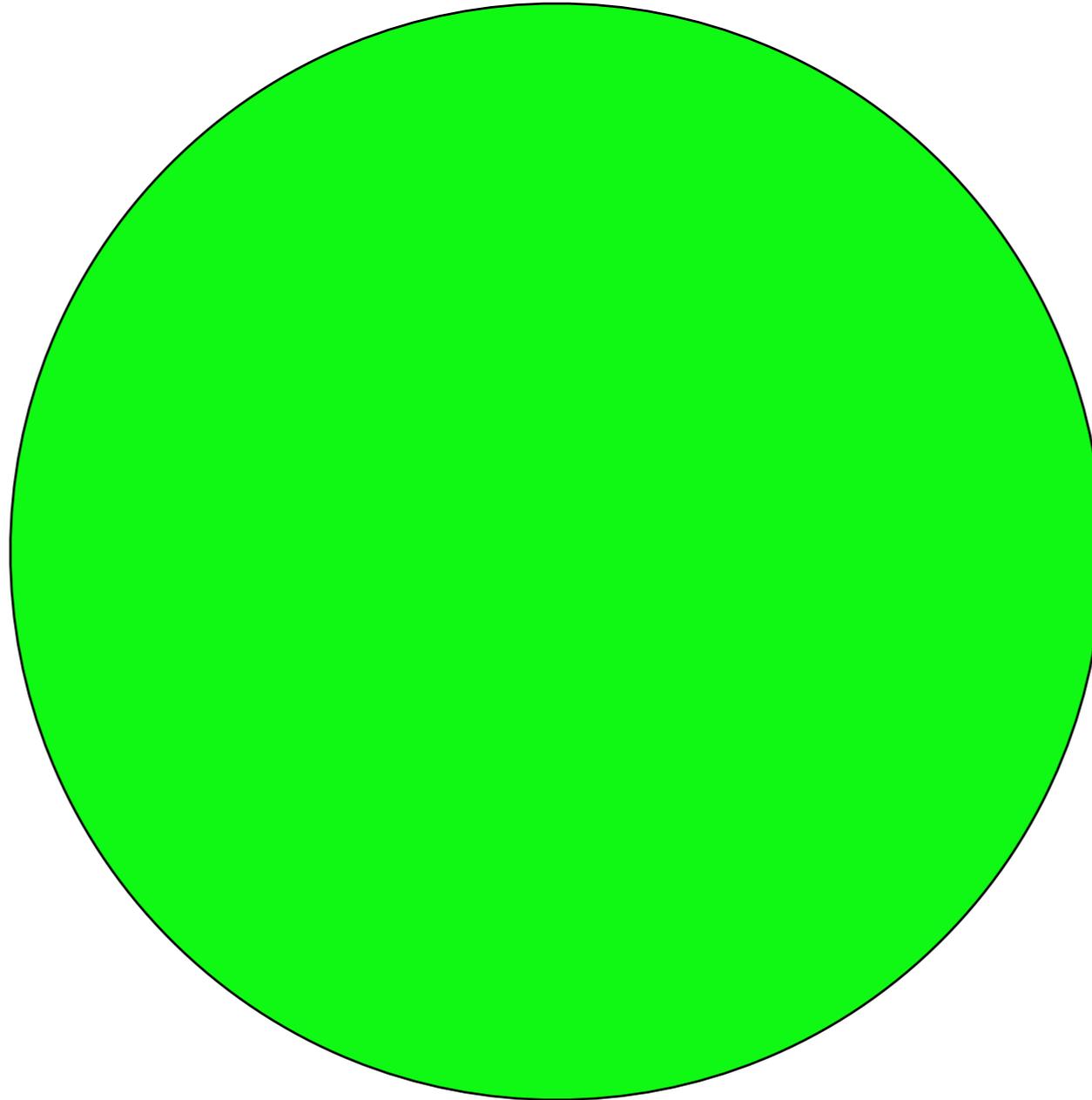
We read 10,000 randomly-chosen sectors ...
and they are all blank



We read 10,000 randomly-chosen sectors ...
and they are all blank



We read 10,000 randomly-chosen sectors ...
and they are all blank

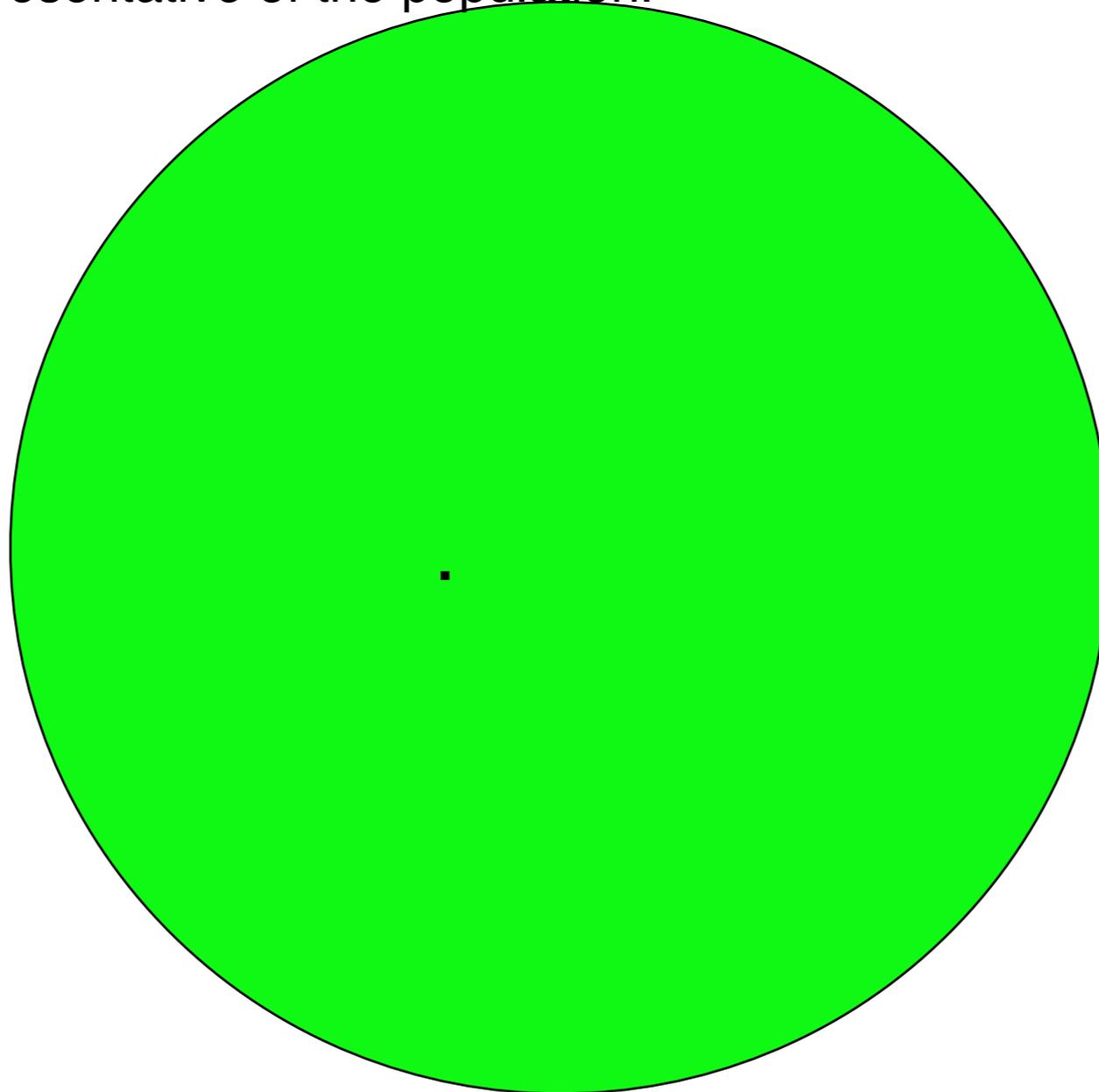


Chances are good that they are all blank.

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.

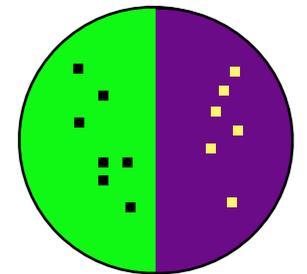


We will only find that 1 sector with exhaustive search.

What about other distributions?

If the disk has 1,000,000,000 blank sectors (1,000,000,000 with data)

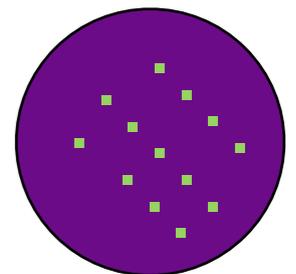
- The sampled frequency should match the distribution.
- *This is why we use random sampling.*



If the disk has 10,000 blank sectors (1,999,990,000 with data)

— and all these are the sectors that we read???

- We are incredibly unlucky.
- ***Somebody has hacked our random number generator!***



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

Assume the disk has 10MB of data --- 20,000 non-zero sectors.

Read just 1 sector; the odds of finding a non-blank sector are:

$$\frac{200,000,000 - 20,000}{200,000,000} = 0.9999.$$

Read 2 sectors. The odds are:

$$\left(\frac{200,000,000 - 20,000}{200,000,000} \right) \left(\frac{199,999,999 - 20,000}{199,999,999} \right) = 0.99980001$$

first pick **second pick** **Odds we may have missed something**

The more sectors picked, the less likely we are to miss all of the sectors that have non-NULL data.

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

Sampled sectors	Probability of not finding data	Non-null data		Probability of not finding data with 10,000 sampled sectors
		Sectors	Bytes	
1	0.99999	20,000	10 MB	0.90484
100	0.99900	100,000	50 MB	0.60652
1000	0.99005	200,000	100 MB	0.36786
10,000	0.90484	300,000	150 MB	0.22310
100,000	0.36787	400,000	200 MB	0.13531
200,000	0.13532	500,000	250 MB	0.08206
300,000	0.04978	600,000	300 MB	0.04976
400,000	0.01831	700,000	350 MB	0.03018
500,000	0.00673	1,000,000	500 MB	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.

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

— So pick 500,000 random sectors. If they are all NULL, then the disk has $p=(1-.00673)$ chance of having 10MB of non-NULL data.

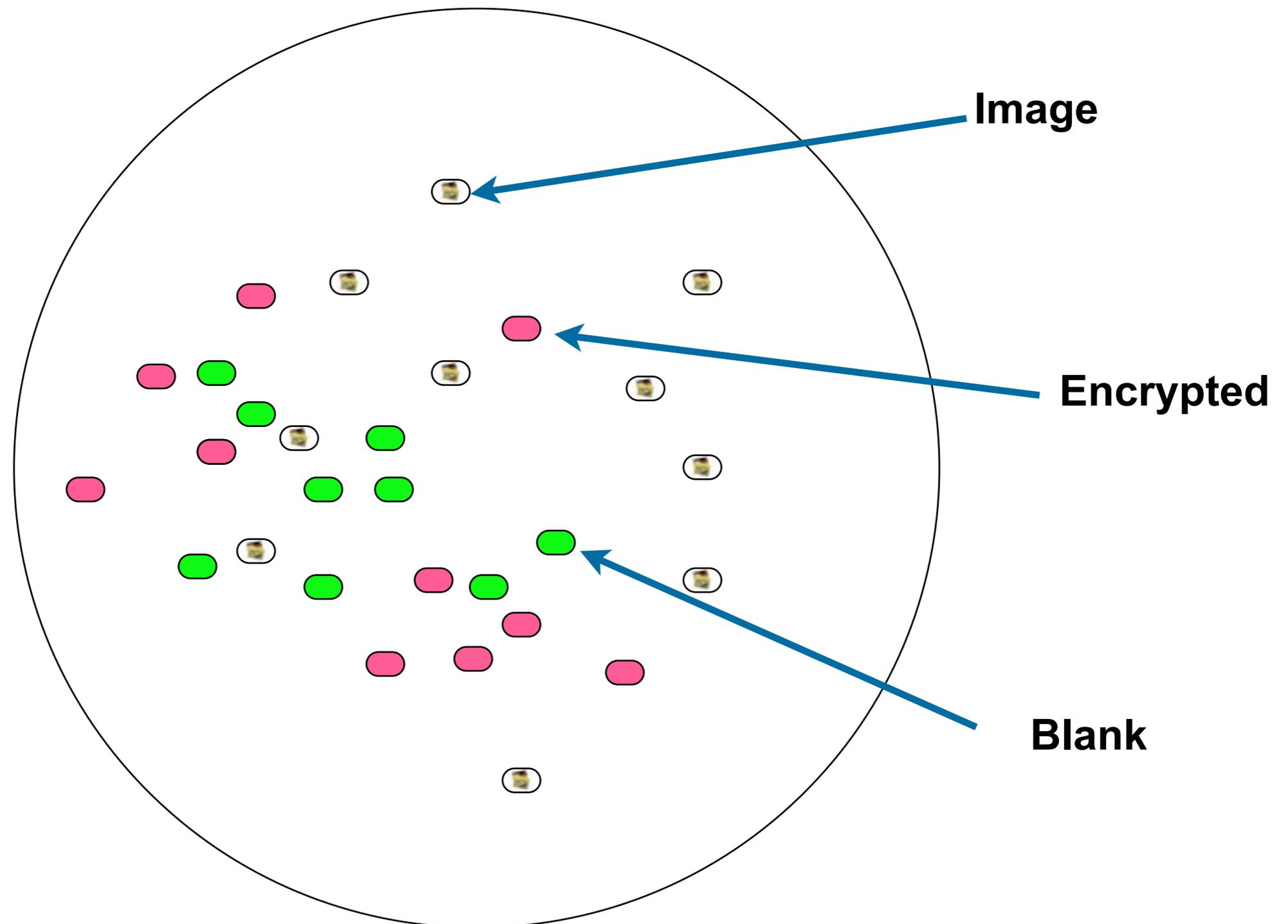
We can use this same technique to calculate the size of the TrueCrypt volume on this iPod.

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

- Apple bought very slow hard drives.



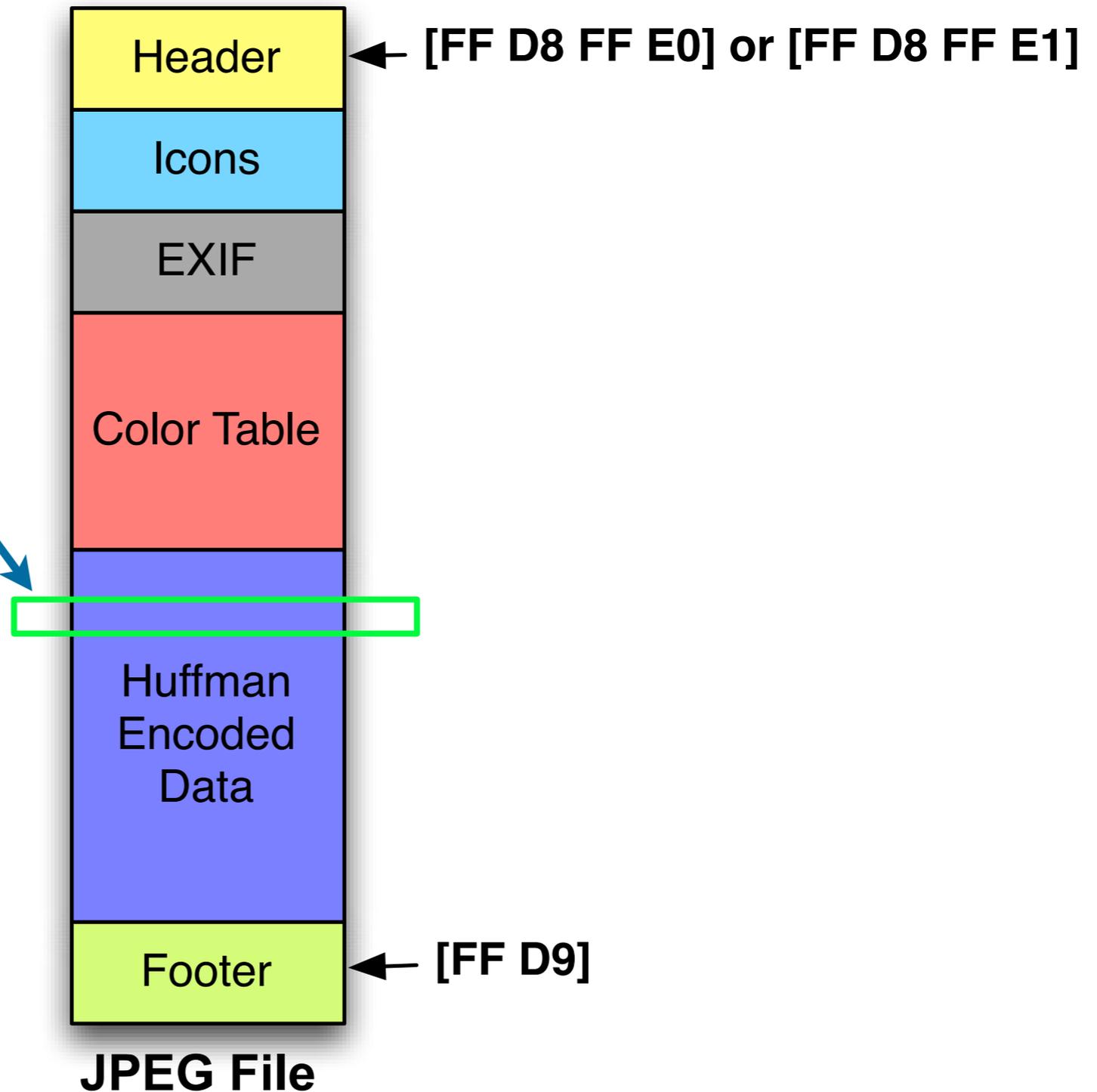
We can get a statistically significant sample in two minutes.



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?



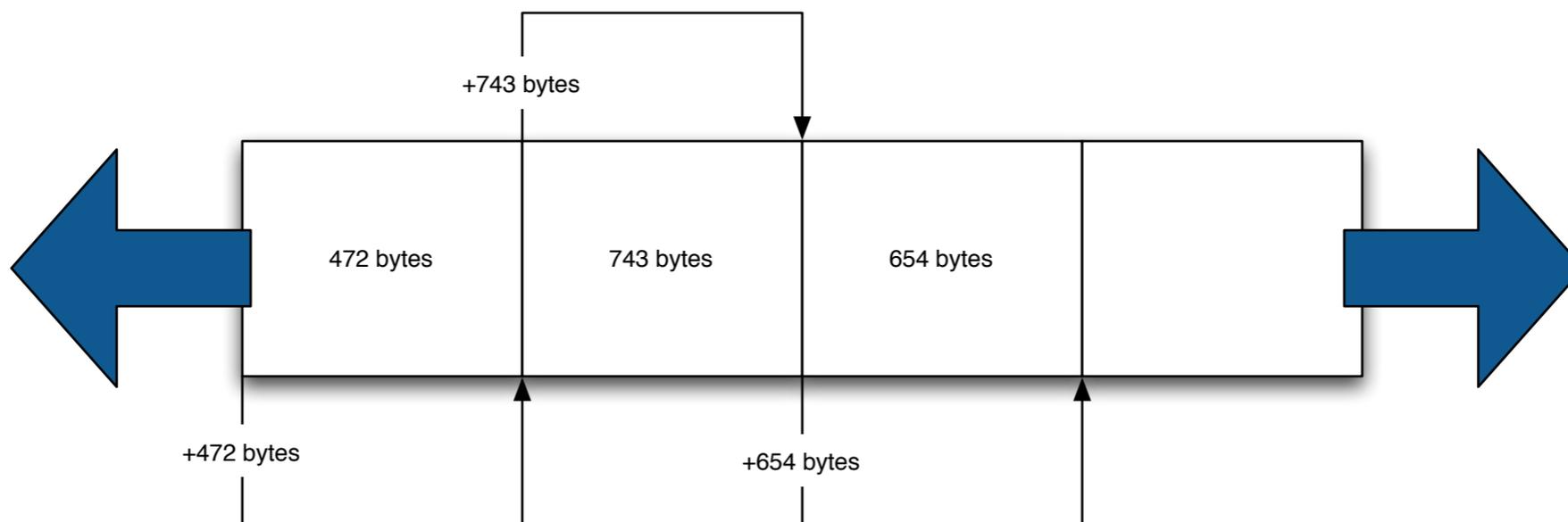
This is called the file fragment classification problem.

HTML files can be reliably detected with HTML tags

```
<body onload="document.getElementById('quicksearch').terms.focus()">  
  <div id="topBar">  
    <div class="widthContainer">  
      <div id="skiplinks">  
        <ul>  
          <li>Skip to:</li>
```

MPEG files can be readily identified through framing

- Each frame has a header and a length.
- Find a header, read the length, look for the next header.



10 years of research on fragment identification...

... mostly using n-gram analysis (bigrams)

Standard approach:

- Get samples of different file types
- Train a classifier (typically k-nearest-neighbor or Support Vector Machines)
- Test classifier on "unknown data"

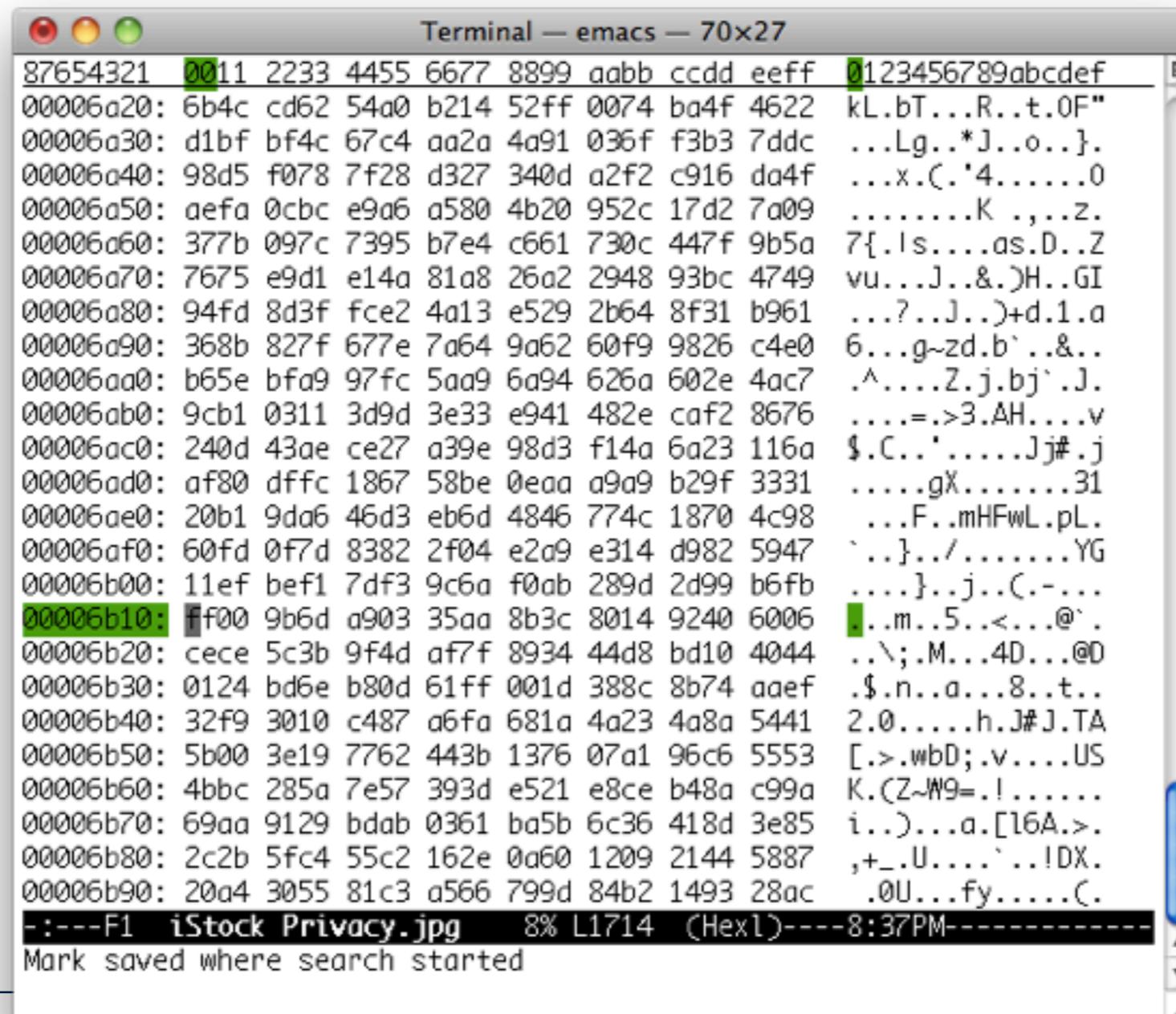
Examples:

- 2001 — McDaniel — "Automatic File Type Detection Algorithm"
 - *header, footer & byte frequency (unigram) analysis (headers work best)*
- 2005 — LiWei-Jen et. al — "Fileprints"
 - *unigram analysis*
- 2006 — Haggerty & Taylor — "FORSIGS"
 - *n-gram analysis*
- 2007 — Calhoun — "Predicting the Type of File Fragments"
 - *statistics of unigrams*

— http://www.forensicswiki.org/wiki/File_Format_Identification

Our approach: hand-tuned discriminators based on a close reading of the specification.

For example, the JPEG format "stuffs" FF with a 00.



```
Terminal — emacs — 70x27
87654321 0011 2233 4455 6677 8899 aabb ccdd eeff 0123456789abcdef
00006a20: 6b4c cd62 54a0 b214 52ff 0074 ba4f 4622 kL.bT...R..t.0F"
00006a30: d1bf bf4c 67c4 aa2a 4a91 036f f3b3 7ddc ...Lg..*J..o..}.
00006a40: 98d5 f078 7f28 d327 340d a2f2 c916 da4f ...x.(.'4.....0
00006a50: aefa 0cbc e9a6 a580 4b20 952c 17d2 7a09 .....K ,...z.
00006a60: 377b 097c 7395 b7e4 c661 730c 447f 9b5a 7{.ls....as.D..Z
00006a70: 7675 e9d1 e14a 81a8 26a2 2948 93bc 4749 vu...J..&.)H..GI
00006a80: 94fd 8d3f fce2 4a13 e529 2b64 8f31 b961 ...?..J..)+d.1.a
00006a90: 368b 827f 677e 7a64 9a62 60f9 9826 c4e0 6...g~zd.b`..&..
00006aa0: b65e bfa9 97fc 5aa9 6a94 626a 602e 4ac7 .^....Z.j.bj`.J.
00006ab0: 9cb1 0311 3d9d 3e33 e941 482e caf2 8676 ....=>3.AH....v
00006ac0: 240d 43ae ce27 a39e 98d3 f14a 6a23 116a $.C..'.....Jj#.j
00006ad0: af80 dffc 1867 58be 0eaa a9a9 b29f 3331 .....gX.....31
00006ae0: 20b1 9da6 46d3 eb6d 4846 774c 1870 4c98 ...F..mHFwL.pL.
00006af0: 60fd 0f7d 8382 2f04 e2a9 e314 d982 5947 `..}..../.....YG
00006b00: 11ef bef1 7df3 9c6a f0ab 289d 2d99 b6fb ....}..j..(-....
00006b10: ff00 9b6d a903 35aa 8b3c 8014 9240 6006 ..m..5..<...@`.
00006b20: cece 5c3b 9f4d af7f 8934 44d8 bd10 4044 ..\;.M...4D...@D
00006b30: 0124 bd6e b80d 61ff 001d 388c 8b74 aaef .$.n..a...8..t..
00006b40: 32f9 3010 c487 a6fa 681a 4a23 4a8a 5441 2.0.....h.J#J.TA
00006b50: 5b00 3e19 7762 443b 1376 07a1 96c6 5553 [.>.wbD;.v....US
00006b60: 4bbc 285a 7e57 393d e521 e8ce b48a c99a K.(Z~W9=!......
00006b70: 69aa 9129 bdab 0361 ba5b 6c36 418d 3e85 i..)...a.[16A.>.
00006b80: 2c2b 5fc4 55c2 162e 0a60 1209 2144 5887 ,+_.U....`...!DX.
00006b90: 20a4 3055 81c3 a566 799d 84b2 1493 28ac .0U...fy.....C.
-:---F1 iStock Privacy.jpg 8% L1714 (HexL)---8:37PM-----
Mark saved where search started
```



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

JPEG HEADER @ byte 0



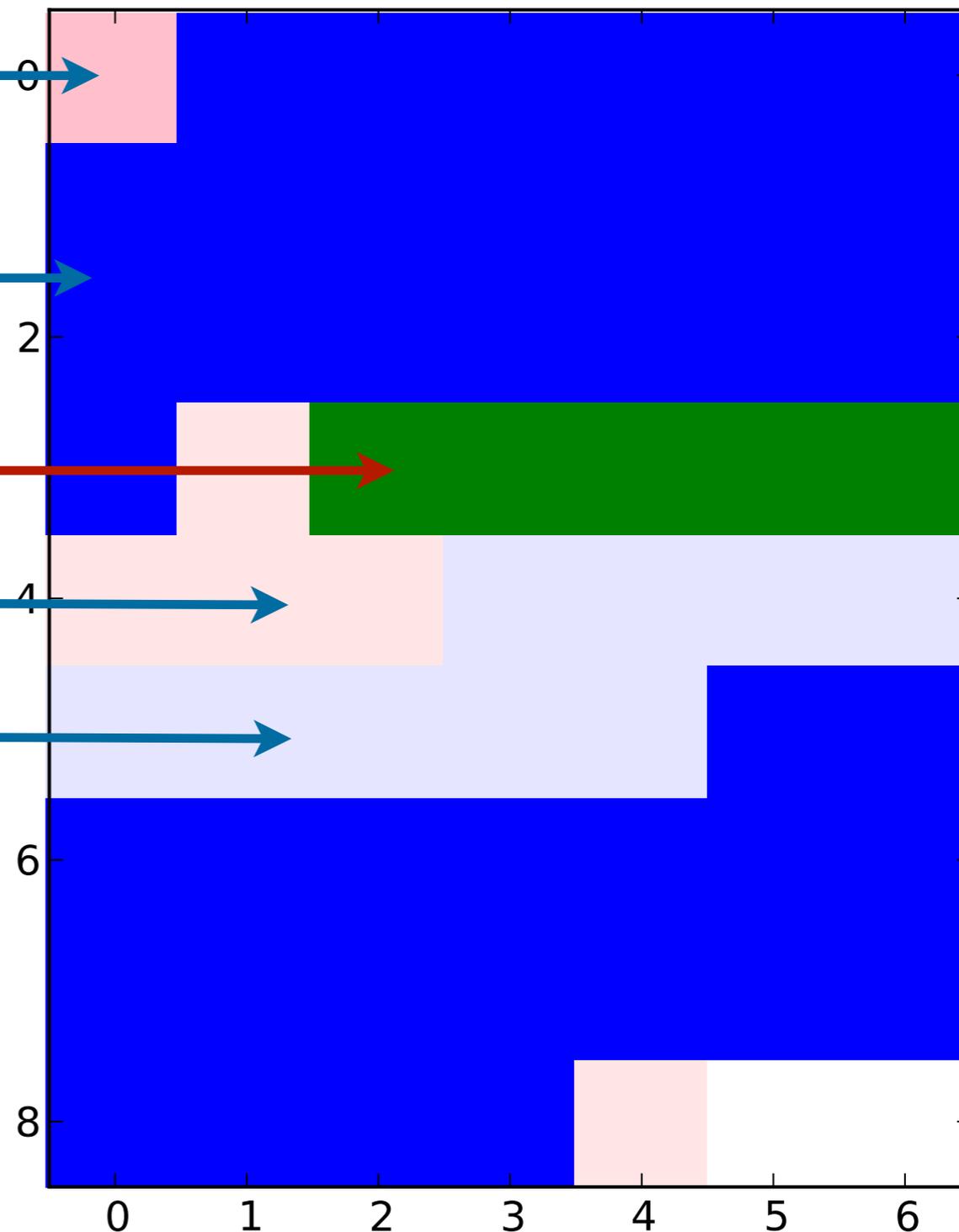
Bytes: 31,046

IN JPEG

Mostly ASCII

low entropy

high entropy



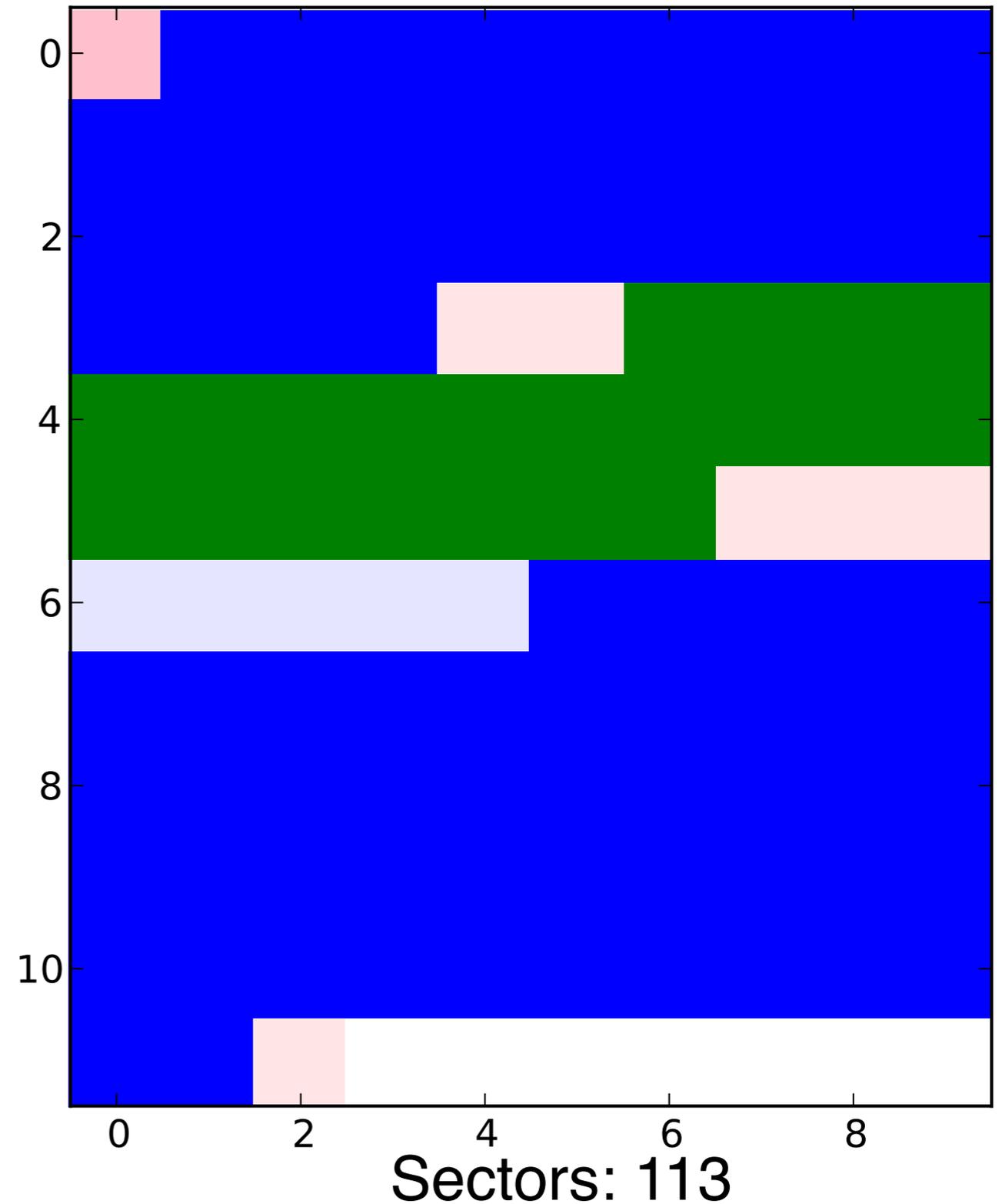
Sectors: 61

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



000897.jpg

Bytes: 57596

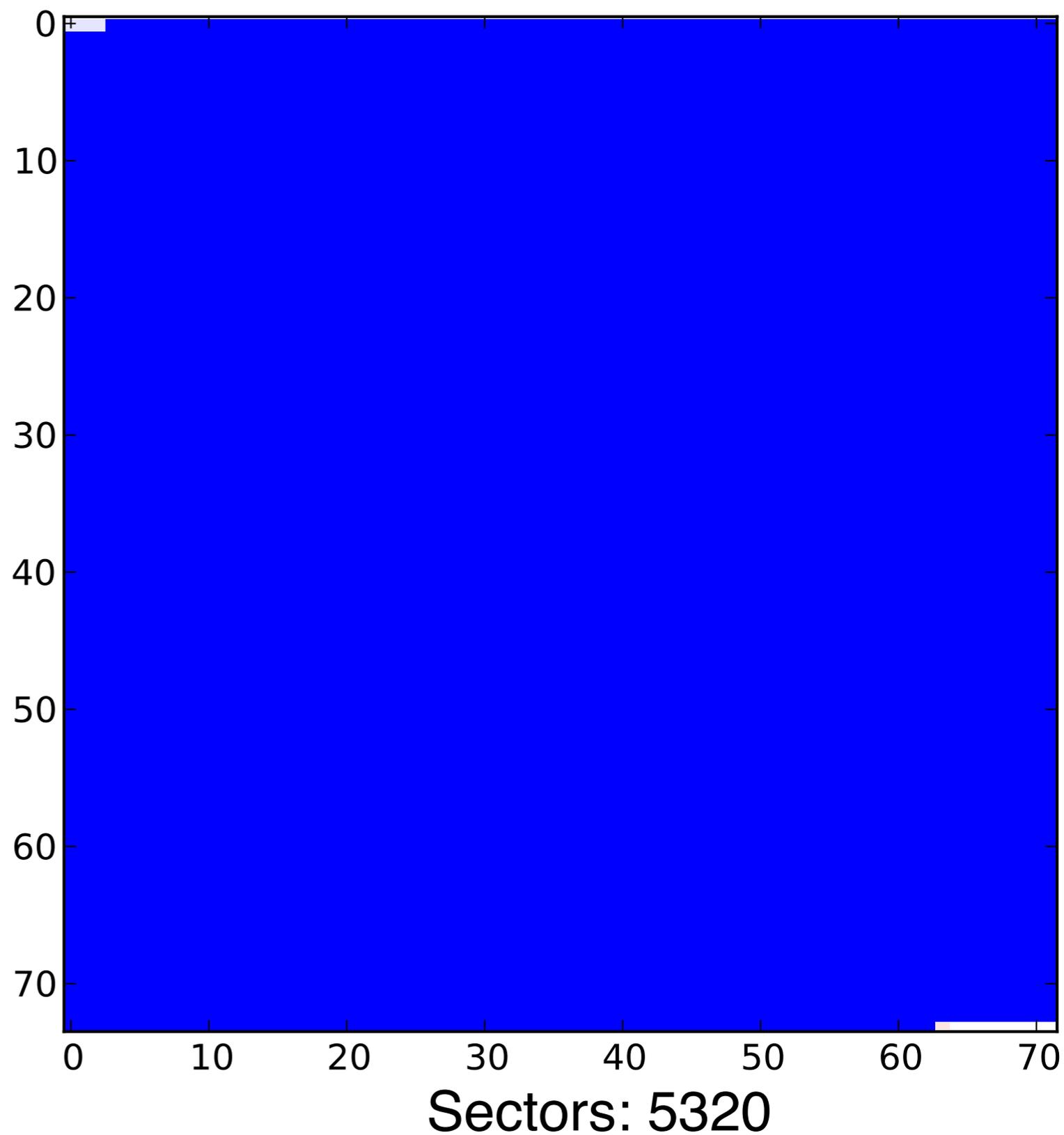


Nearly 100% of this file identifies as “JPEG.”



000888.jpg

Bytes: 2,723,425



We developed five fragment discriminators.

JPEG — High entropy and FF00 pairs.

MPEG — Frames

Huffman-Coded Data — High Entropy & Autocorrelation

"Random" or "Encrypted" data — High Entropy & No autocorrelation

Distinct Data — a block from an image, movie, or encrypted file.



208 distinct 4096-byte
block hashes



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

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

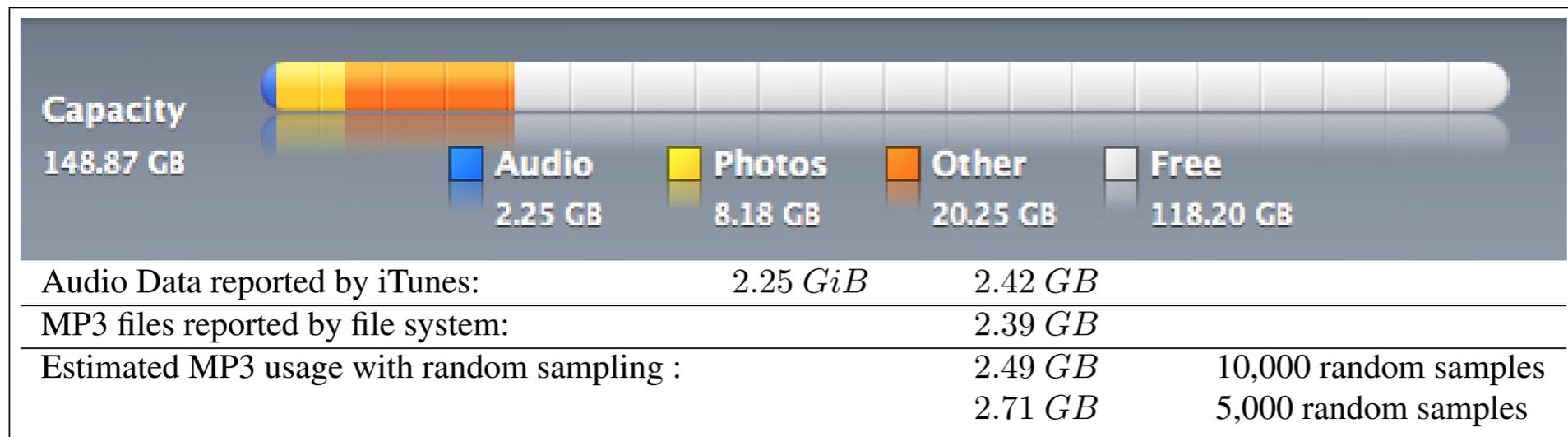


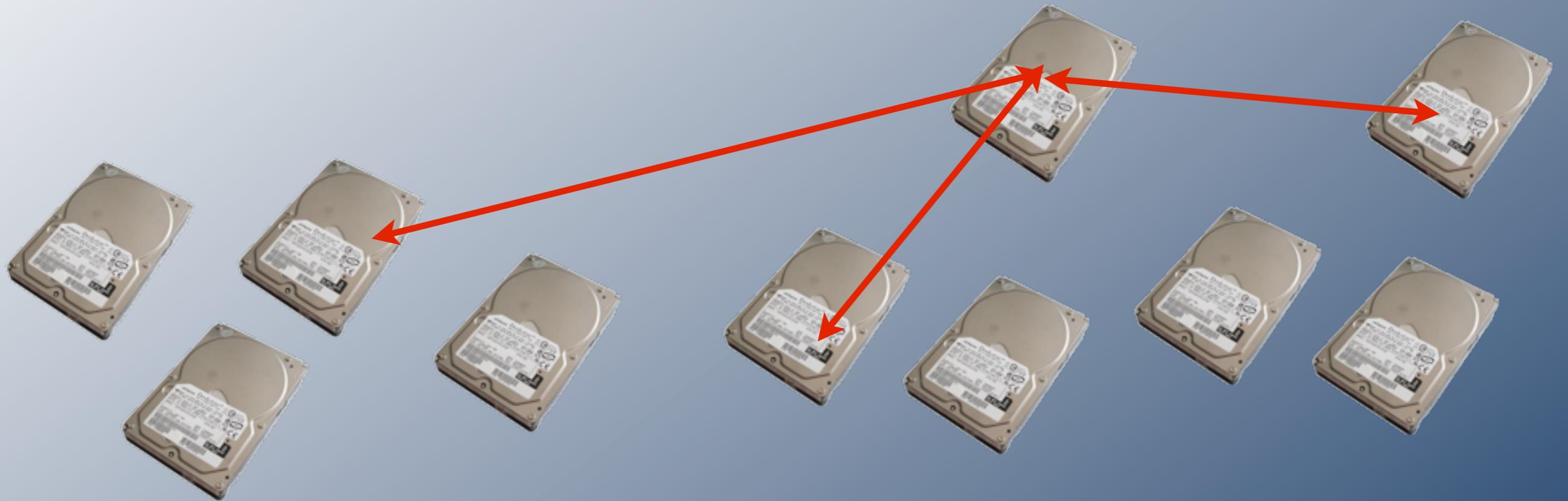
Figure 1: Usage of a 160GB iPod reported by iTunes 8.2.1 (6) (top), as reported by the file system (bottom center), and as computing with random sampling (bottom right). Note that iTunes usage actually in GiB, even though the program displays the “GB” label.



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*



Where do we go from here?

There are a lot of fun projects to pursue.

Algorithms that work...

- With different kinds of data.
- At different resolutions & orders of magnitude.

Software that can...

- Automatically identify outliers and inconsistencies.
- Automatically present complex results in simple, straightforward reports.
- Combine stored data, network data, and Internet-based information.

Many of the techniques here are also applicable to:

- Social Network Analysis.
- Personal Information Management.
- Data mining unstructured information.

My challenges: innovation, scale & community

Most innovative forensic tools fail when they are deployed.

- Production data *much larger* than test data.
 - *One drive might have 10,000 email addresses, another might have 2,000,000.*
- Production data *more heterogeneous* than test data.
- Analysts have less experience & time than tool developers.

How to address?

- Attention to usability & recovery.
- High Performance Computing for testing.
- Programming languages that are *safe* and *high-performance*.

Moving research results from lab to field is itself a research problem.

In summary, there is an urgent need for fundamental research in automated computer forensics.

Most work to date has been data recovery and reverse engineering.

- User-level file systems
- Recovery of deleted files.

To solve tomorrow's hard problems, we need:

- Algorithms that exploit large data sets (>10TB)
- Machine learning to find *outliers* and *inconsistencies*.
- Algorithms tolerant of data that is *dirty* and *damaged*.

Work in automated forensics is *inherently interdisciplinary*.

- Systems, Security, and Network Engineering
- Machine Learning
- Natural Language Processing
- Algorithms (compression, decompression, big data)
- High Performance Computing
- Human Computer Interactions

There are many opportunities to work outside CS

Need to engage with:

- Policy makers
- Police, Defense & Intelligence Communities.
- Defense Bar

Interesting legal issues.

- Data acquisition.
- Privacy
- Research Oversight (Institutional Review Boards.)

—*For more information, see <http://www.simson.net/> or <http://forensicswiki.org/>*

Questions?

