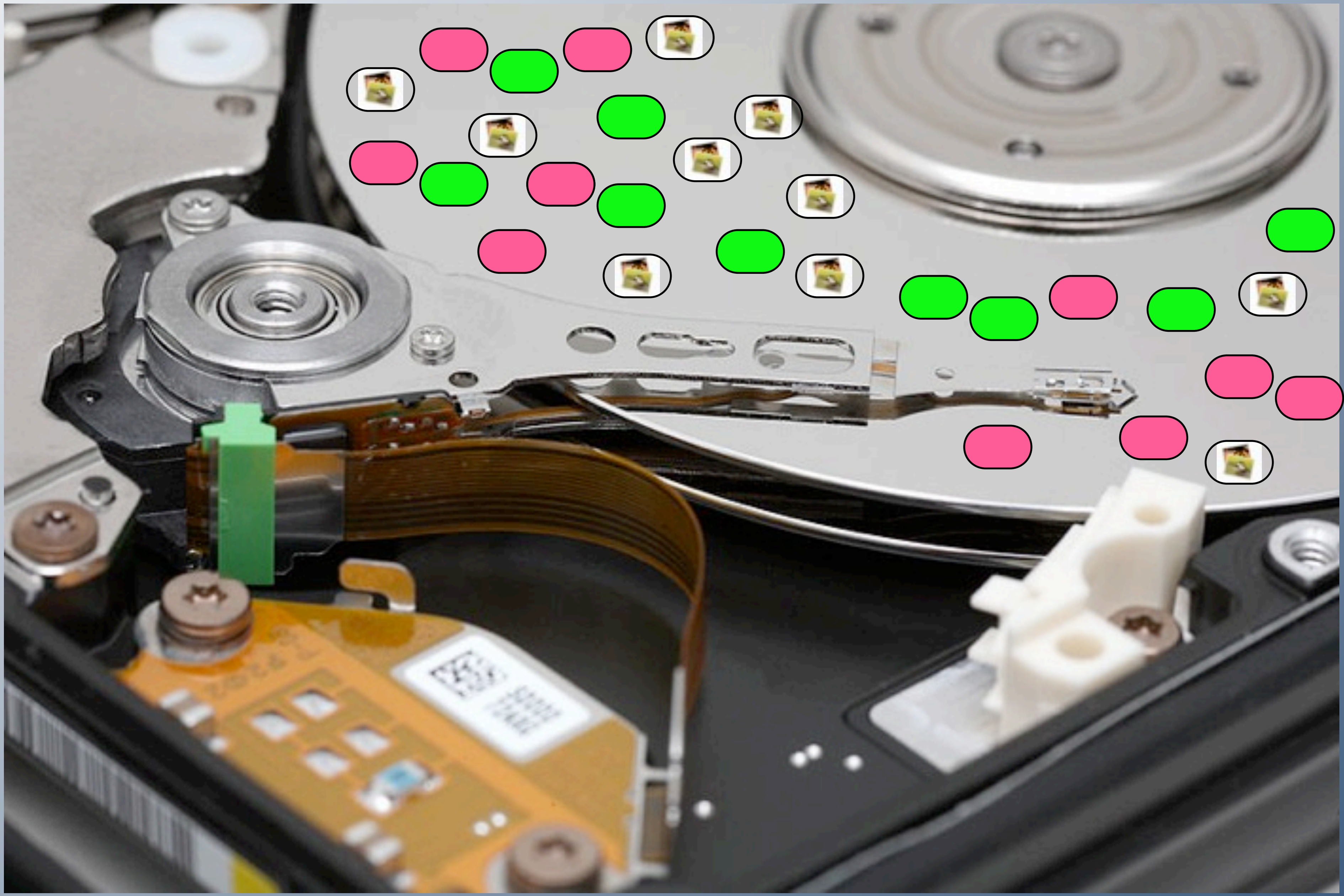


# Instant Drive Forensics with Statistical Sampling

Simson L. Garfinkel
Naval Postgraduate School
simsong@acm.org

# Question: Can we analyze a 1TB drive in a minute?

What if US agents encounter a hard drive at a border crossing?

Or a search turns up a room filled with servers?

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

| |  |  |
|---|---|---|
| Minutes | 208 | 1 |
| Max Data Read | 1 TB | 4.8 GB |
| Max Seeks | 15 million | 17,000<br>(≈3.5msec per seek) |

4.8 GB (0.48%) is a tiny fraction of the disk.

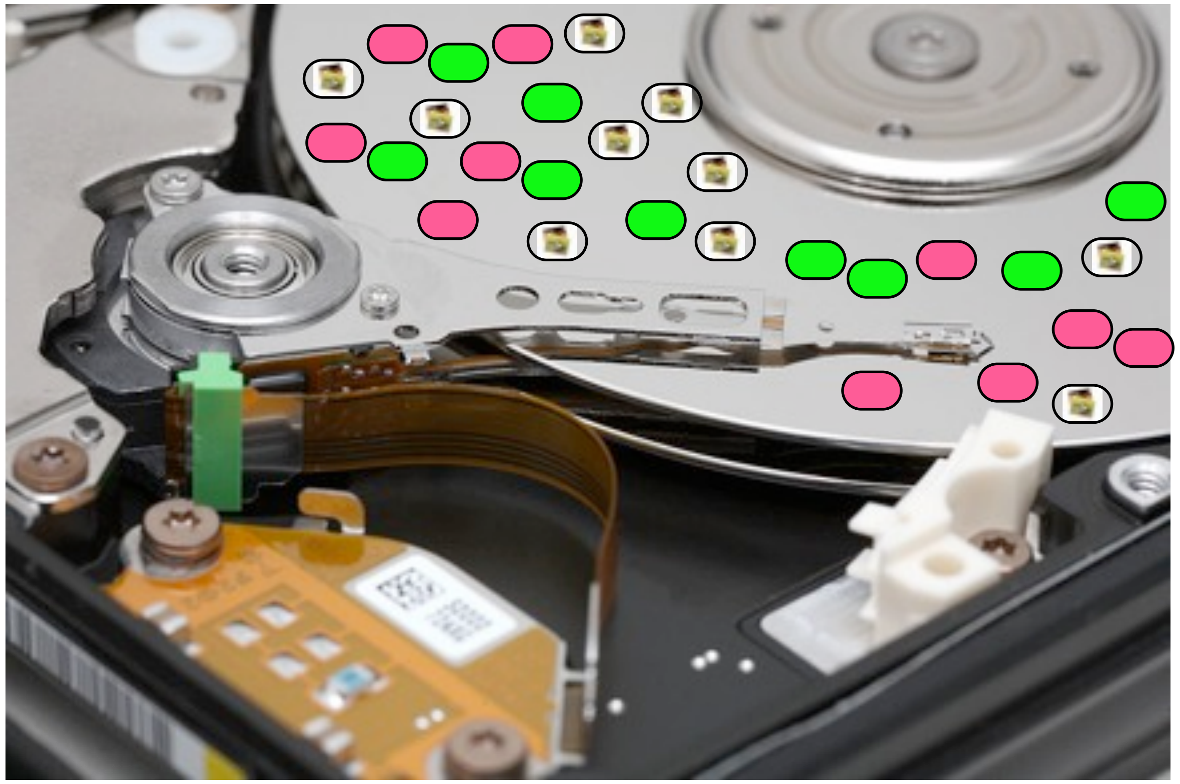
But 4.8 GB is a lot of data!

# Hypothesis: The contents of the disk can be predicted by identifying the contents of randomly chosen sectors.

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

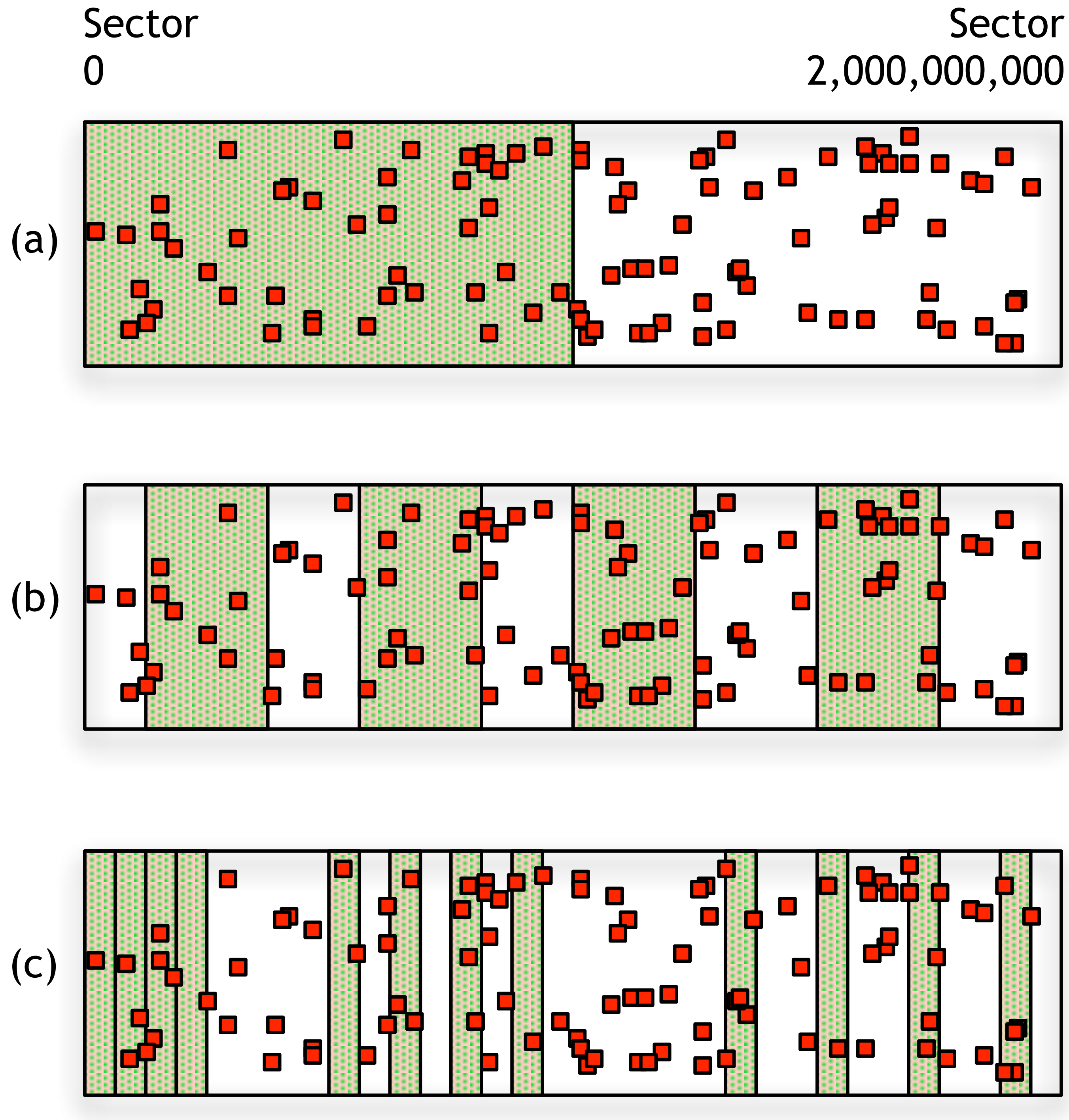

The challenge is identifying *likely voters.*

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



The challenge is *identifying the content* of the sampled sectors.

# We use random sampling; any other approach could be exploited by the enemy.




But sampling has an important limitation...

# Recall data on hard drives divides into three categories:

Resident Data

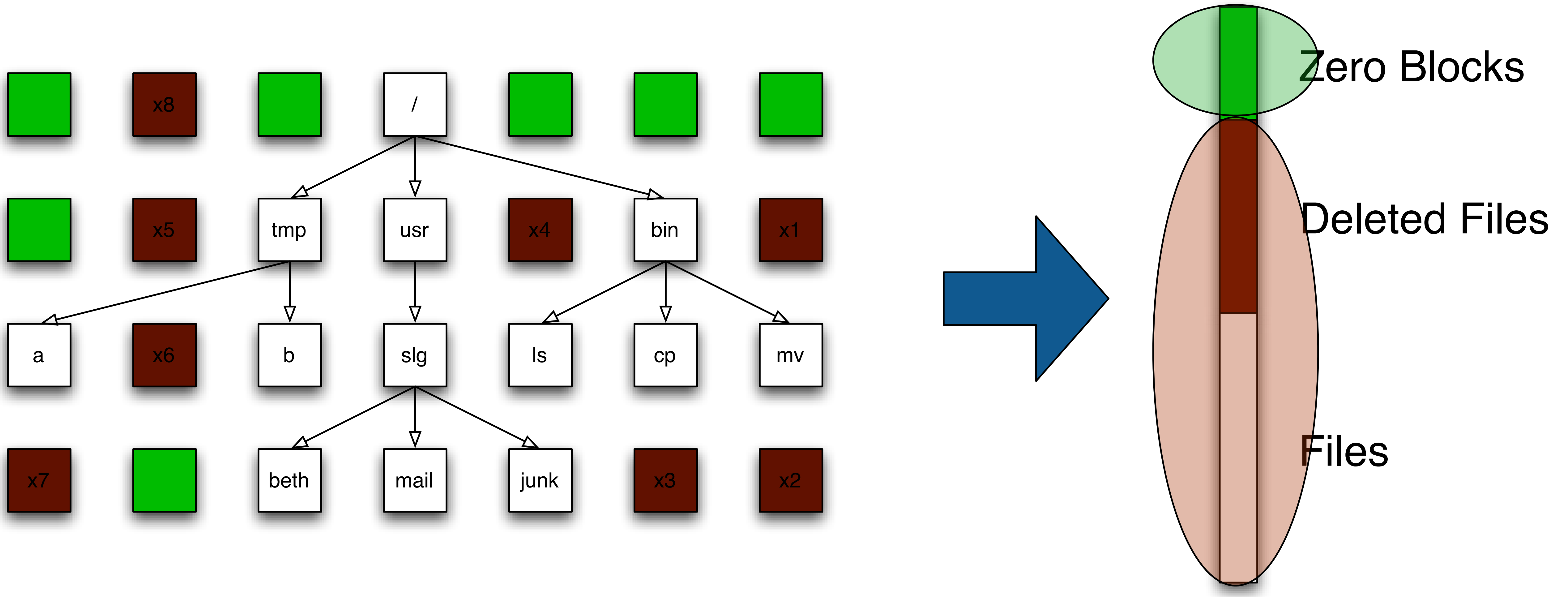
Deleted Data

**user files**
**email messages**
**[temporary files]**

No Data

**blank sectors**

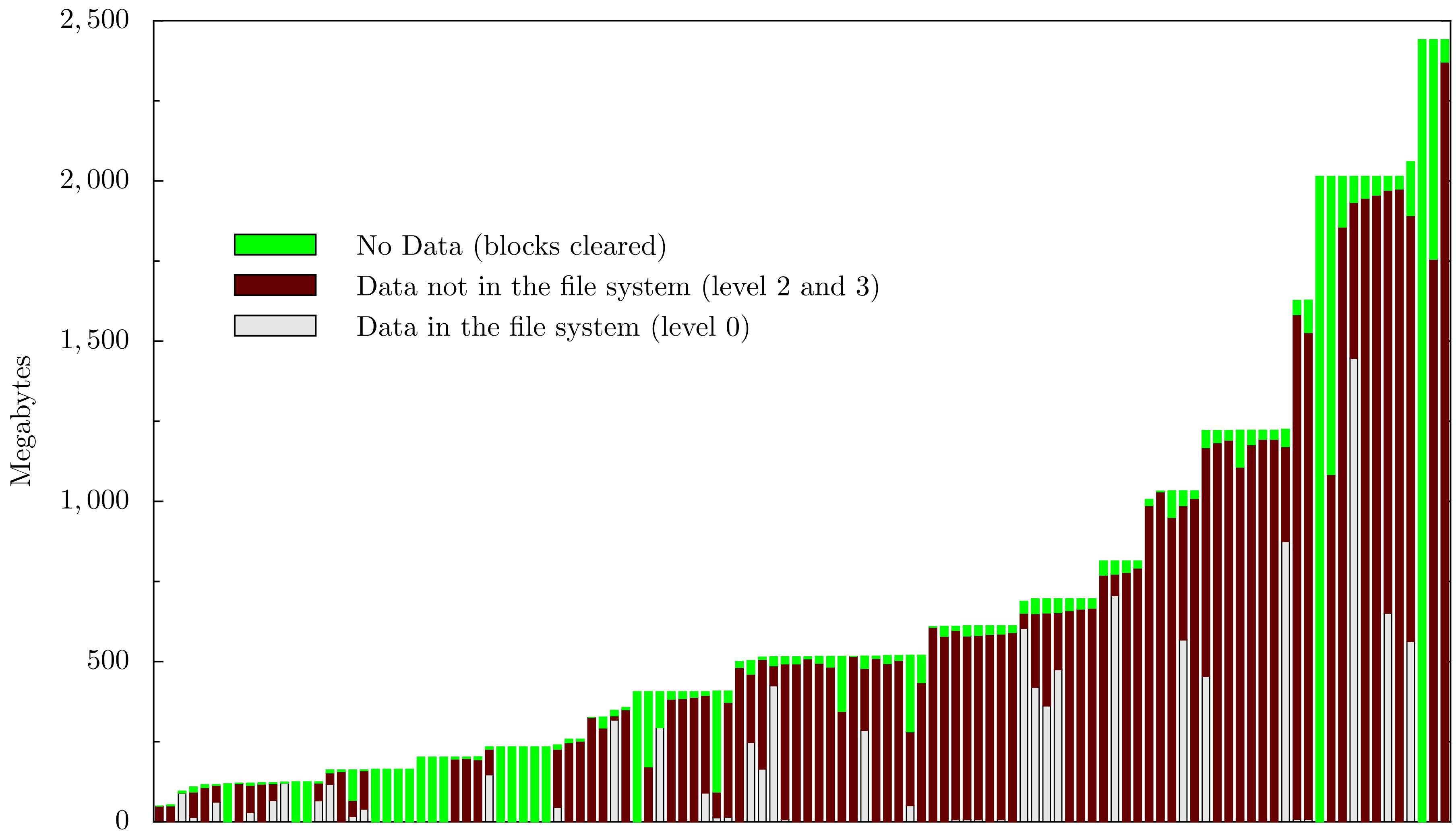
# Sampling can distinguish between "zero" and data.
# It can't distinguish between resident and deleted.



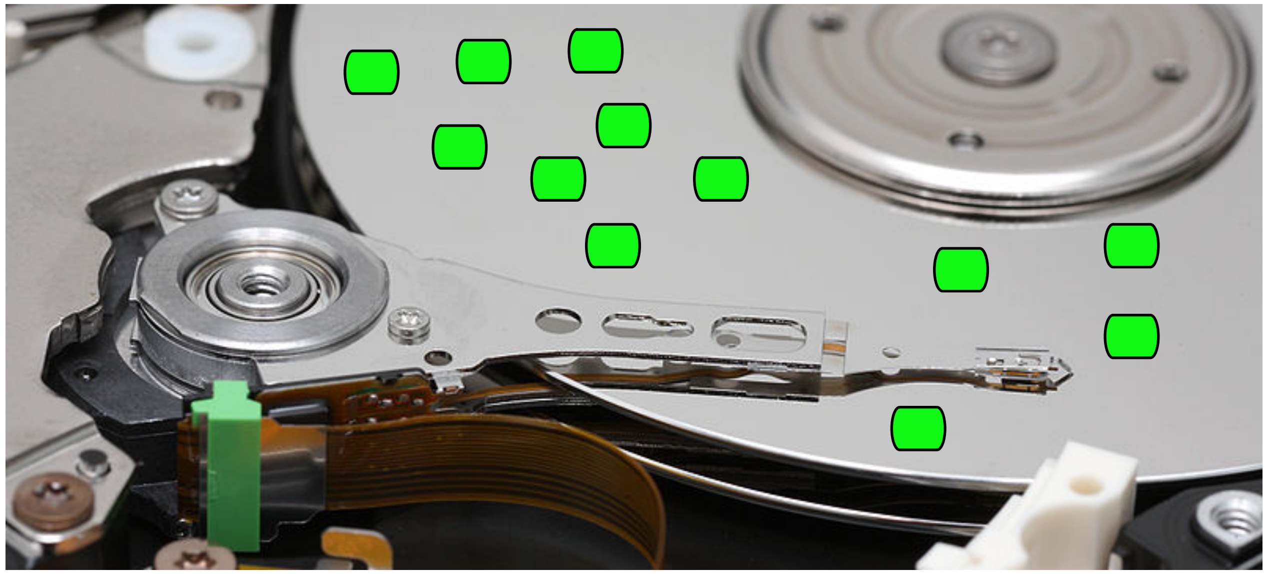
# Let's simplify the problem.
# Can we use statistical sampling to verify wiping?

I bought 2000 hard drives between 1998 and 2006.

Most of were not properly wiped.



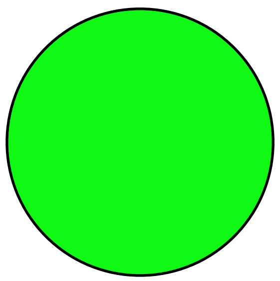

# It should be easy to use random sampling to distinguish a properly cleared disk from one that isn't.



# What does it mean if 10,000 randomly chosen sectors are blank?

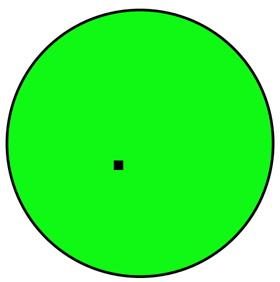
If the disk has 2,000,000,000 blank sectors (0 with data)

- The sample is identical to the population



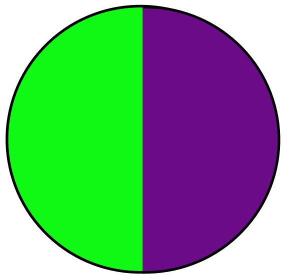
If the disk has 1,999,999,999 blank sectors  (1 with data)

- The sample is representative of the population.
- We will only find that 1 sector using exhaustive search.



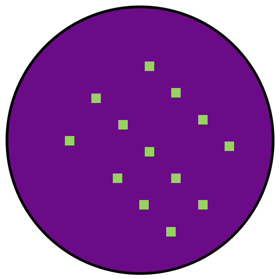
If the disk has 1,000,000,000 blank sectors (1,000,000,000 with data)

- Something about our sampling matched the allocation pattern.
- *This is why we use random sampling.*



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

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



# Rephrase the problem.
# Not a blank disk; a disk with less than 10MB of data.

| | | |
|---|---|---|
| Sectors on disk: | 2,000,000,000 | (1TB) |
| Sectors with data: | 20,000 | (10 MB) |

Chose one sector. Odds of missing the data:

- (2,000,000,000 - 20,000) / (2,000,000,000) = 0.99999
- You are *very likely* to miss one of 20,000 sectors if you pick just one.

Chose a second sector. Odds of missing the data on both tries:

- 0.99999 * (1,999,999,999-20,000) / (1,999,999,999) = .99998
- You are still *very likely* to miss one of 20,000 sectors if you pick two.

But what if you pick 1000? Or 10,000? Or 100,000?

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

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

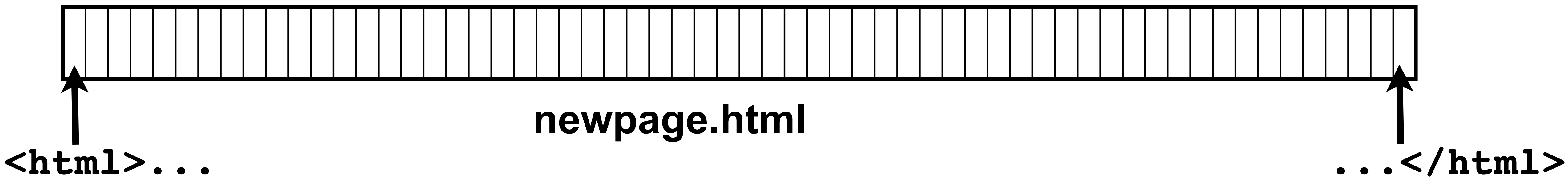
| Sampled sectors | Probability of not finding data |
|---|---|
| 1 | 0.99999 |
| 2 | 0.99998 |
| 100 | 0.99900 |
| 1000 | 0.99005 |
| 10,000 | 0.90484 |
| 20,000 | 0.81873 |
| 40,000 | 0.67032 |
| 60,000 | 0.54881 |
| 80,000 | 0.44932 |
| 100,000 | 0.36787 |
| 150,000 | 0.22312 |
| 200,000 | 0.13532 |
| 300,000 | 0.04978 |
| 400,000 | 0.01831 |
| 500,000 | 0.00673 |

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

500,000 blank randomly chosen sectors should be good enough!

# Part 2: Can we classify files based on a sector?

A file 30K consists of 60 sectors:




Many file types have characteristics headers and footer:

| | header | footer |
|---|---|---|
| HTML | `<html>` | `</html>` |
| JPEG | `<FF><D8><FF><E0>`<br>`<00><10>JFIF<00>` | `<FF><D9>` |
| ZIP | `PK<03><0D>` | `<00><00><00><00>` |

But what about the file in the middle?

# Fragment classification:
# Different file types require different strategies.

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>
```

JPEG files can be identified through the "FF" escape.

- FF must be coded as FF00.
- So if there are a lot of FF00s and few FF01 through FFFF it must be a JPEG.

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.

# This works!
# We identify the *content* of a 160GB iPod in 118 seconds.

## Identifiable:

- Blank sectors
- JPEGs
- Encrypted data
- HTML

## Report:

- Audio Data Reported by iTunes: 2.42GB
- MP3 files reported by file system: 2.39GB
- Estimated MP3 usage:
  - *—2.71GB (1.70%) with 5,000 random samples*
  - *—2.49GB (1.56%) with 10,000 random samples*


## Sampling took 118 seconds.

# Work to date:

## Publications:

- Roussev, Vassil, and Garfinkel, Simson, File Classification Fragment---The Case for Specialized Approaches, Systematic Approaches to Digital Forensics Engineering (IEEE/SADFE 2009), Oakland, California.
- Farrell, P., Garfinkel, S., White, D. Practical Applications of Bloom filters to the NIST RDS and hard drive triage, Annual Computer Security Applications Conference 2008, Anaheim, California, December 2008.

## Work in progress:

- Alex Nelson (PhD Candidate, UCSC) summer project
- Using “Hamming,” our 1100-core cluster for novel SD algorithms.
- Roussev’s Similarity Metric