Finding privacy leaks and stolen data with bulk data analysis and optimistic decoding

Univeristy of Maryland
Wednesday, October 16, 2013 / 11:00am

Simson L. Garfinkel

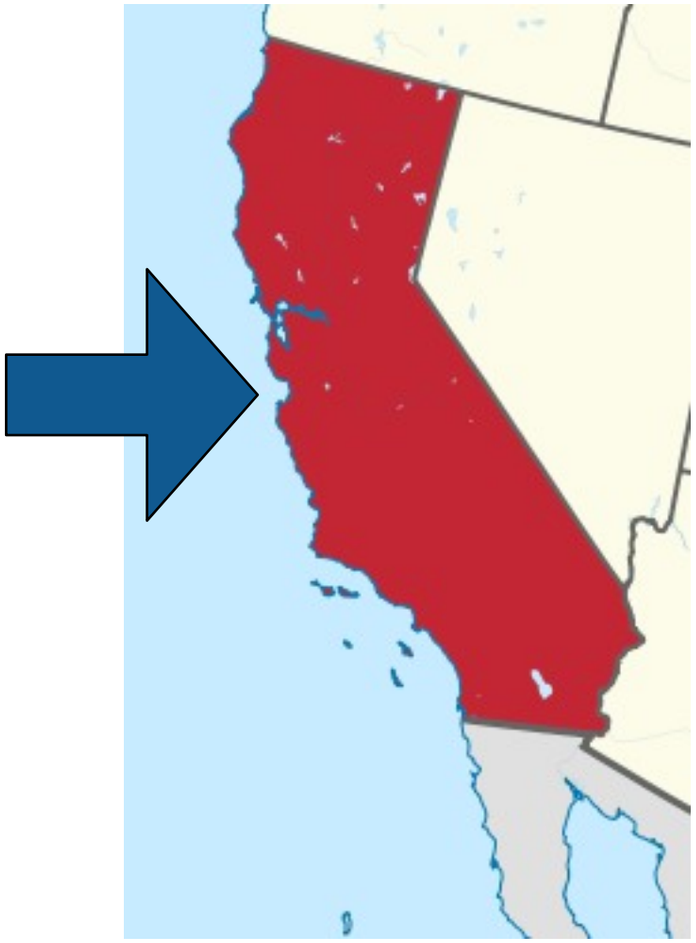
Naval Postgraduate School

http://simson.net/

NPS is the Navy's Research University.

Monterey, CA — 1500 students

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



National Capital Region (NCR) Office

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

Digital information is pervasive in today's society.

Many potential sources of digital information:

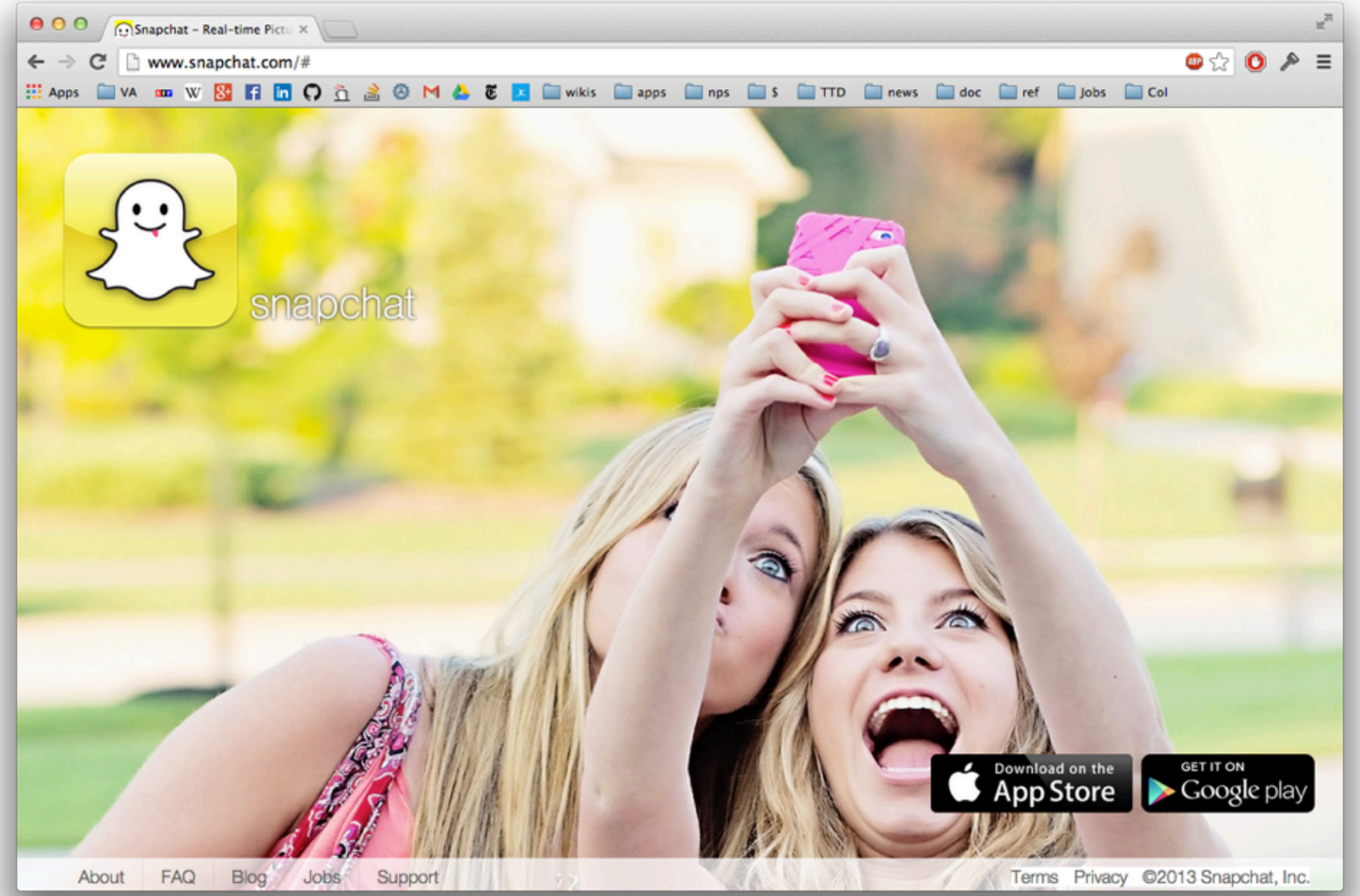
- Desktops; Laptops
- Tablets; Cell Phones
- Internet-Based Services
- Cars

My research makes internal, technical data usable by non-technologists

- Law Enforcement — Document a conspiracy (stock fraud; murder-for-hire; Silk Road)
- DOD — Identify members of a terrorist organization.
- Ordinary people — Recover deleted files.

These tools can also be used to audit software for privacy leaks.

Consider snapchat!



Snapchat promised users that expired images could not be viewed unless “saved.”

Snapchat – Real-time Pictu

www.snapchat.com/#

Apps VA wikis apps nps $ TTD news doc ref Jobs Col

snapchat

Download on the App Store

GET IT ON Google play

About FAQ Blog Jobs Support

be subject to additional fees by your carrier.

How do you send a snap to more than one person?
To send a snap to more than one person, select multiple friends in the "Send To..." screen by tapping the grey circles to the right of their names. Then tap send at the bottom of the screen.

How can I add a caption?
You can add a caption by tapping the preview screen after you have taken a snap.

Is there any way to view an image after the time has expired?
No, snaps disappear after the timer runs out. You can save snaps that you capture by pressing the save button on the preview screen.

What if I take a screenshot?
Screenshots can be captured if you're quick. The sender will be notified if we detect you have taken a screenshot.

Usernames and Passwords

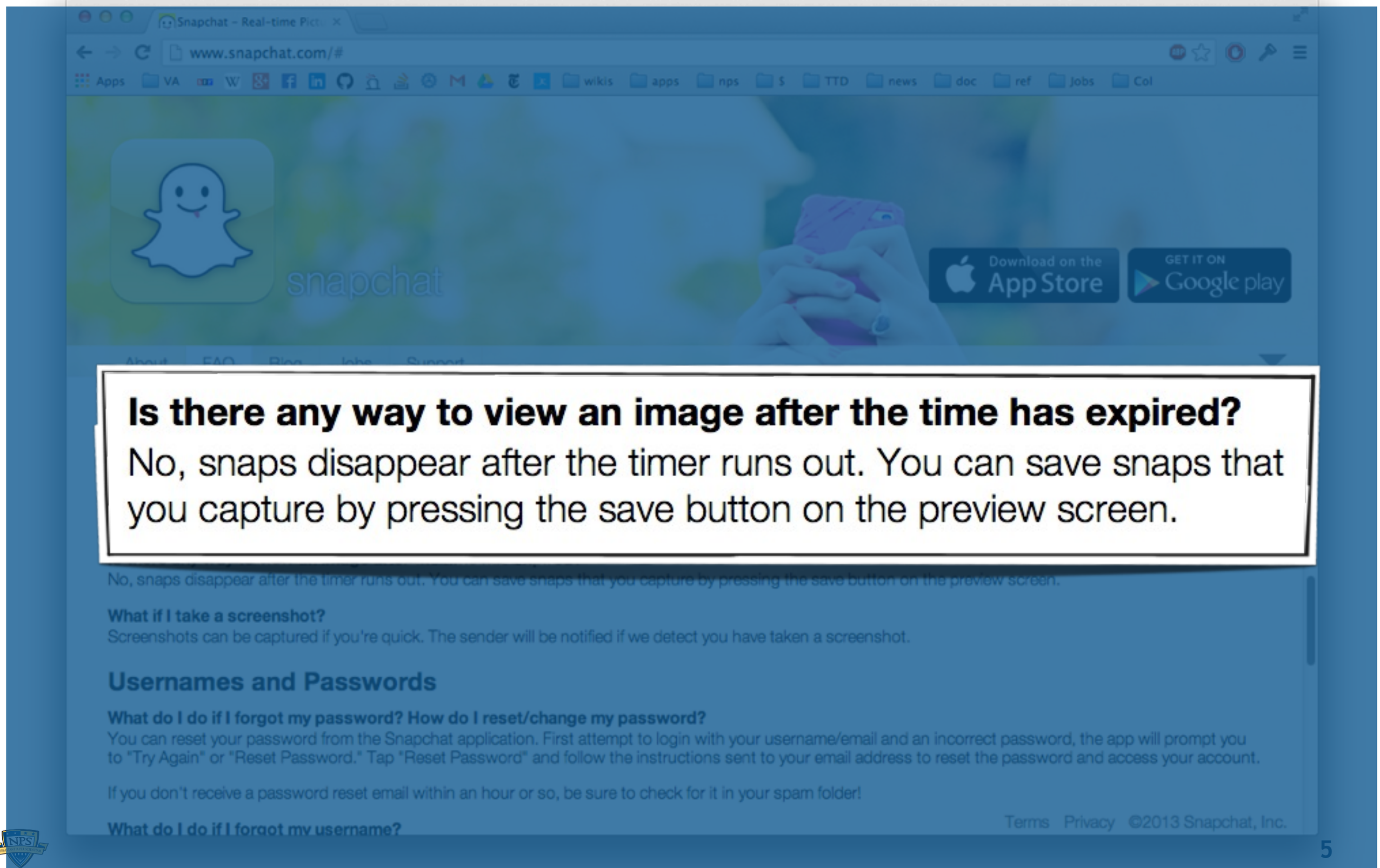
What do I do if I forgot my password? How do I reset/change my password?
You can reset your password from the Snapchat application. First attempt to login with your username/email and an incorrect password, the app will prompt you to "Try Again" or "Reset Password." Tap "Reset Password" and follow the instructions sent to your email address to reset the password and access your account.

If you don't receive a password reset email within an hour or so, be sure to check for it in your spam folder!

What do I do if I forgot my username?

Terms Privacy ©2013 Snapchat, Inc.

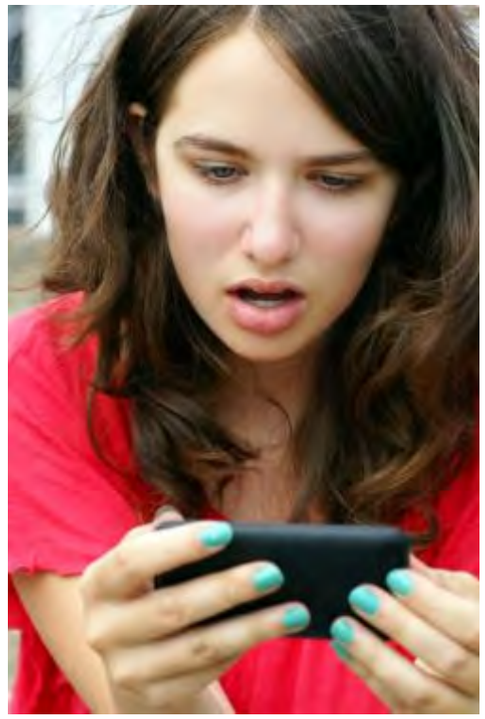
Snapchat promised users that expired images could not be viewed unless “saved.”



OMG! — Expired images not actually deleted. They were just hidden from view.

OMG, "Deleted" Snapchat Sexts Can Actually Be Recovered

By Will Oremus | Posted Thursday, May 9, 2013, at 3:56 PM



Snapchat's users shouldn't be shocked to find that their images can be recovered even after they "self-destruct"—but they will be anyway.
Sylvie Bouchard/Shutterstock.com

The premise of Snapchat is simple: Send a photo or short video to a friend, and it will self-destruct after 10 seconds. That way, it won't wind up on the Internet and ruin anyone's reputation, friendships, or career.

Needless to say, that has made it a wildly popular choice for sexting. But Snapchat's appeal goes far beyond that. In an age in which "privacy" and "technology" have become almost antonymous, it has been billed as the anti-Facebook—a communications tool that deletes your data rather than preserving, analyzing, and trading on it. In short, it's supposed to make messaging fun again.

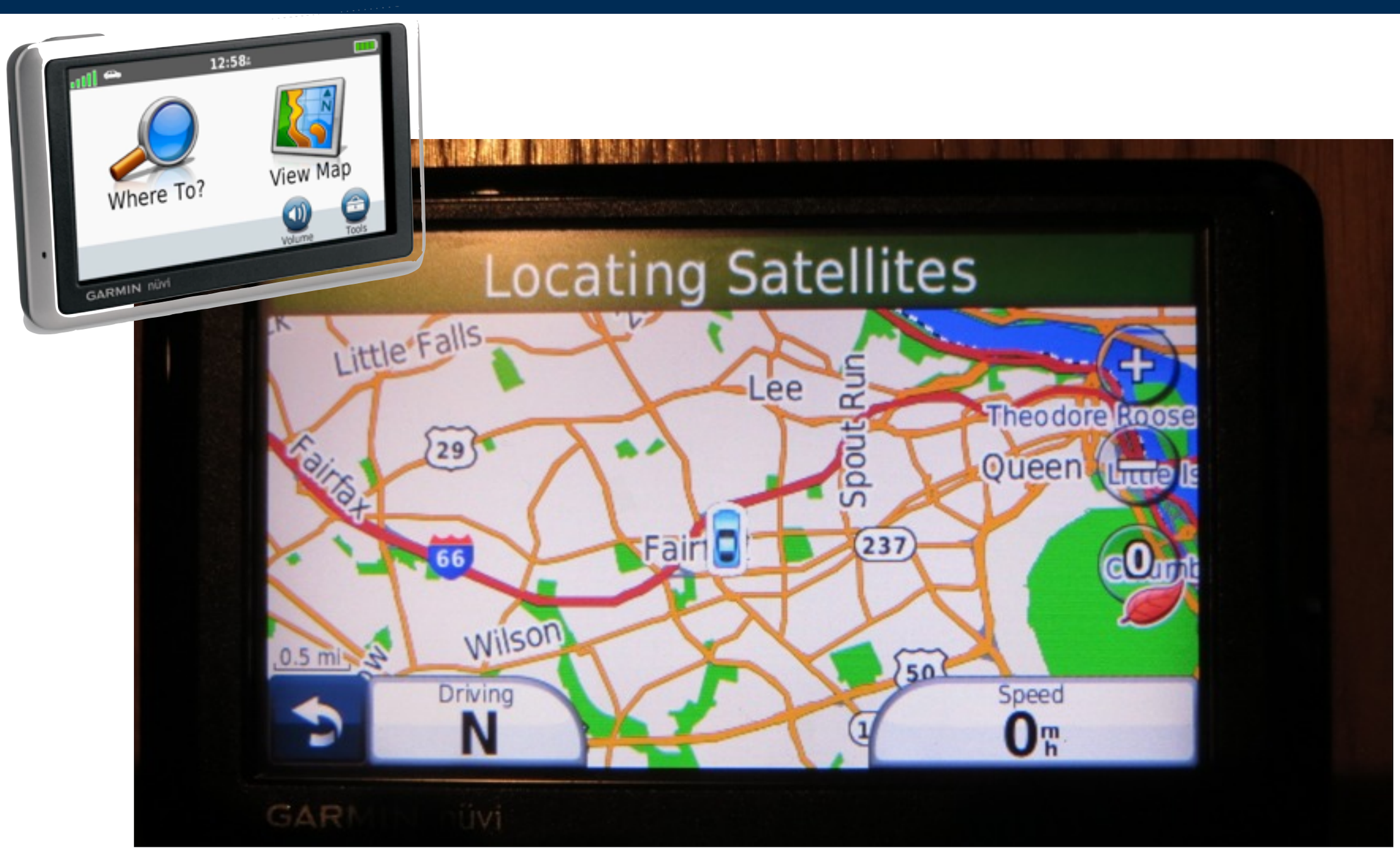
But the app's security has never been ironclad. As the media have repeatedly warned parents, and parents in turn warned their kids, message recipients can still save a compromising image by taking a quick screenshot. But Snapchat tries to mitigate the risk somewhat by automatically notifying the sender when that happens. If someone screenshots you, it's a virtual slap in the face. If they don't, you can assume you're in the clear.

Except that apparently you can't. KSL-TV in Utah reports that an Orem-based firm called Decipher Forensics has figured out a way to recover supposedly deleted images from the recipient's phone. The process isn't simple: 24-year-old Decipher forensics examiner Richard Hickman told the network that it takes him about six hours, on average, to image the phone's data. So far he can only do it with Android devices, though he's working on doing the same for iOS. But his firm is now offering to perform the recovery procedure for anyone who wants it, from parents

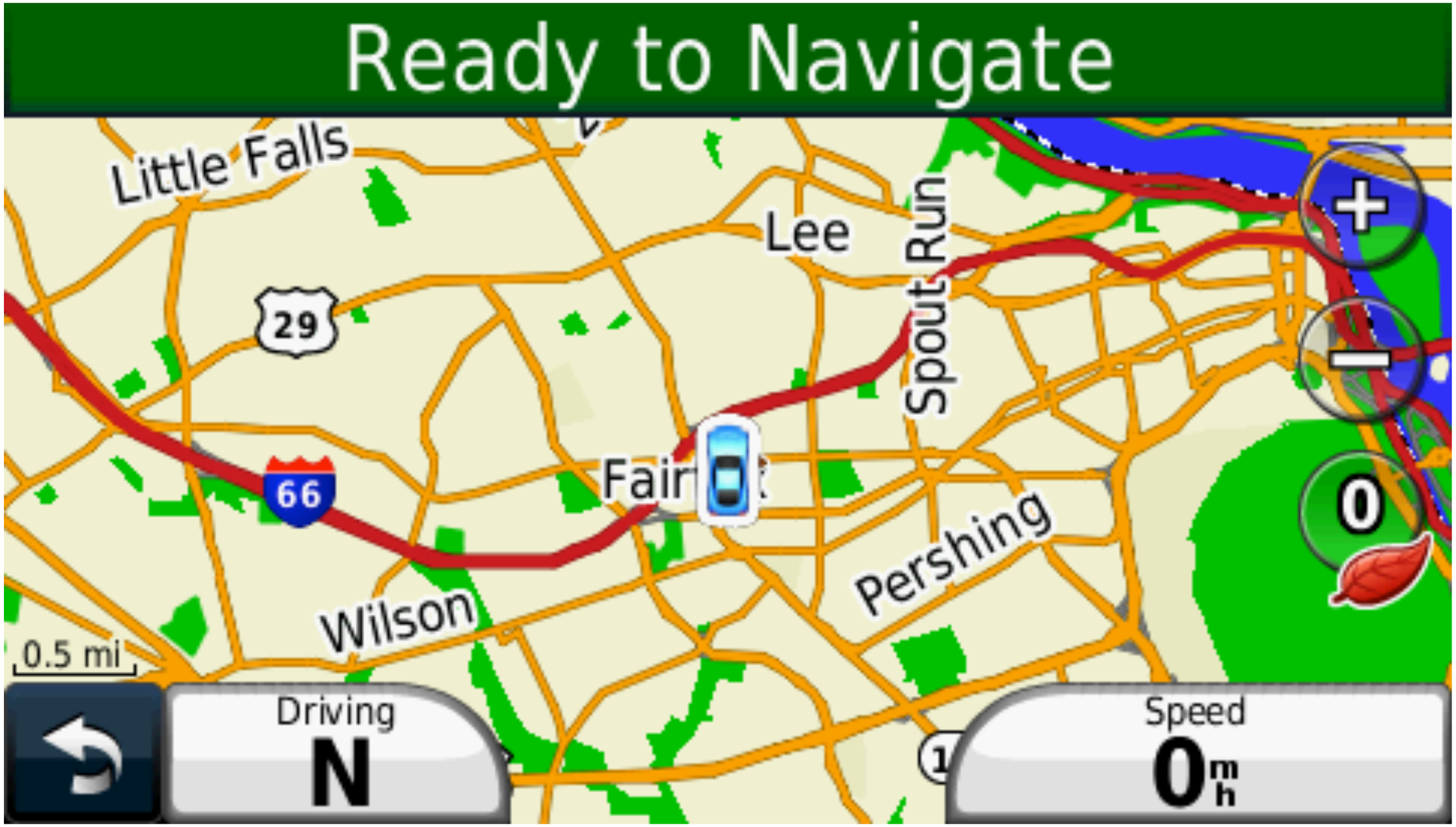
Garmin Nüvi Data Retention

Many devices preserve information in non-obvious ways.

Many devices preserve information in non-obvious ways.



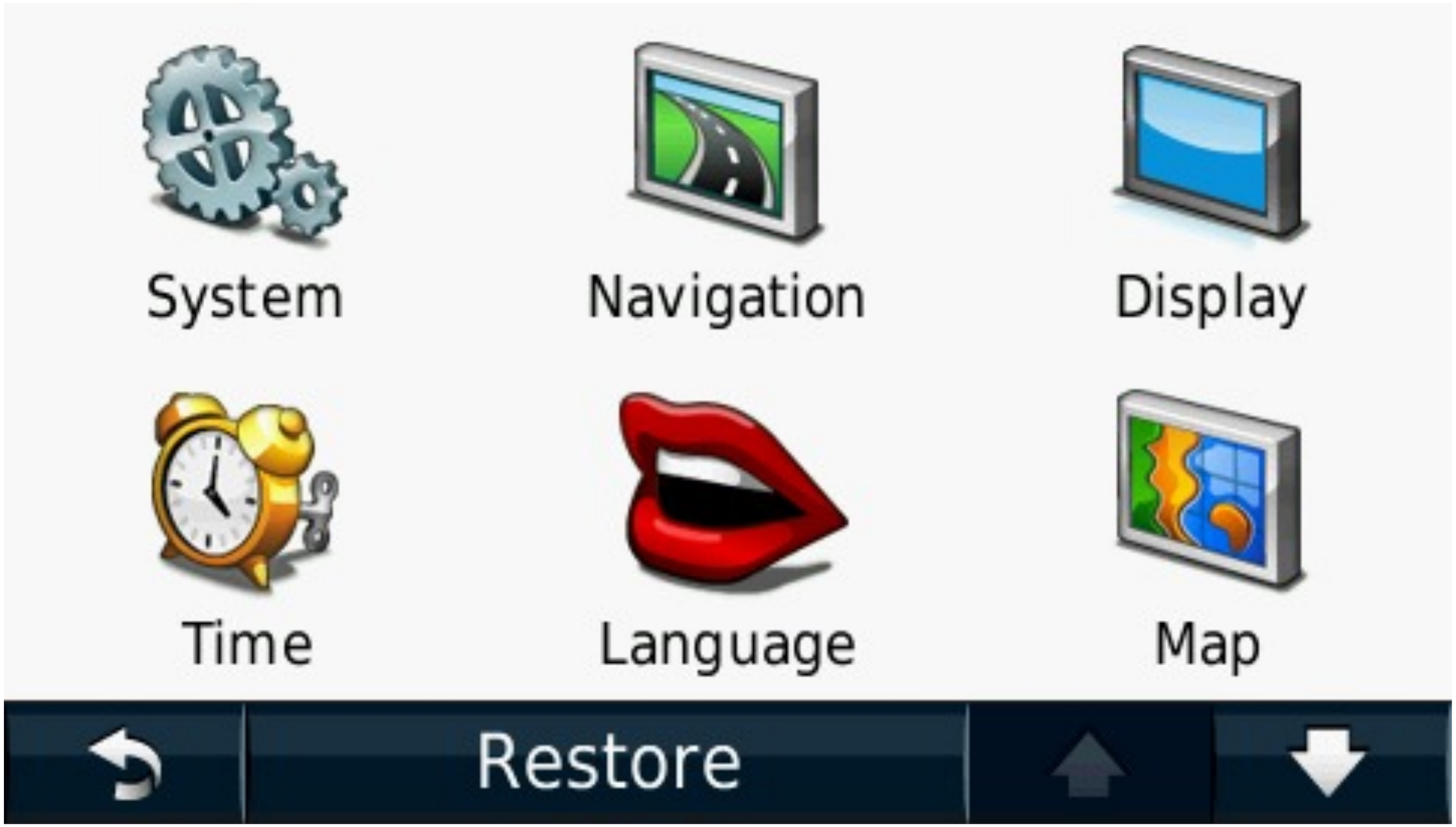
Garmin maps show where you are...

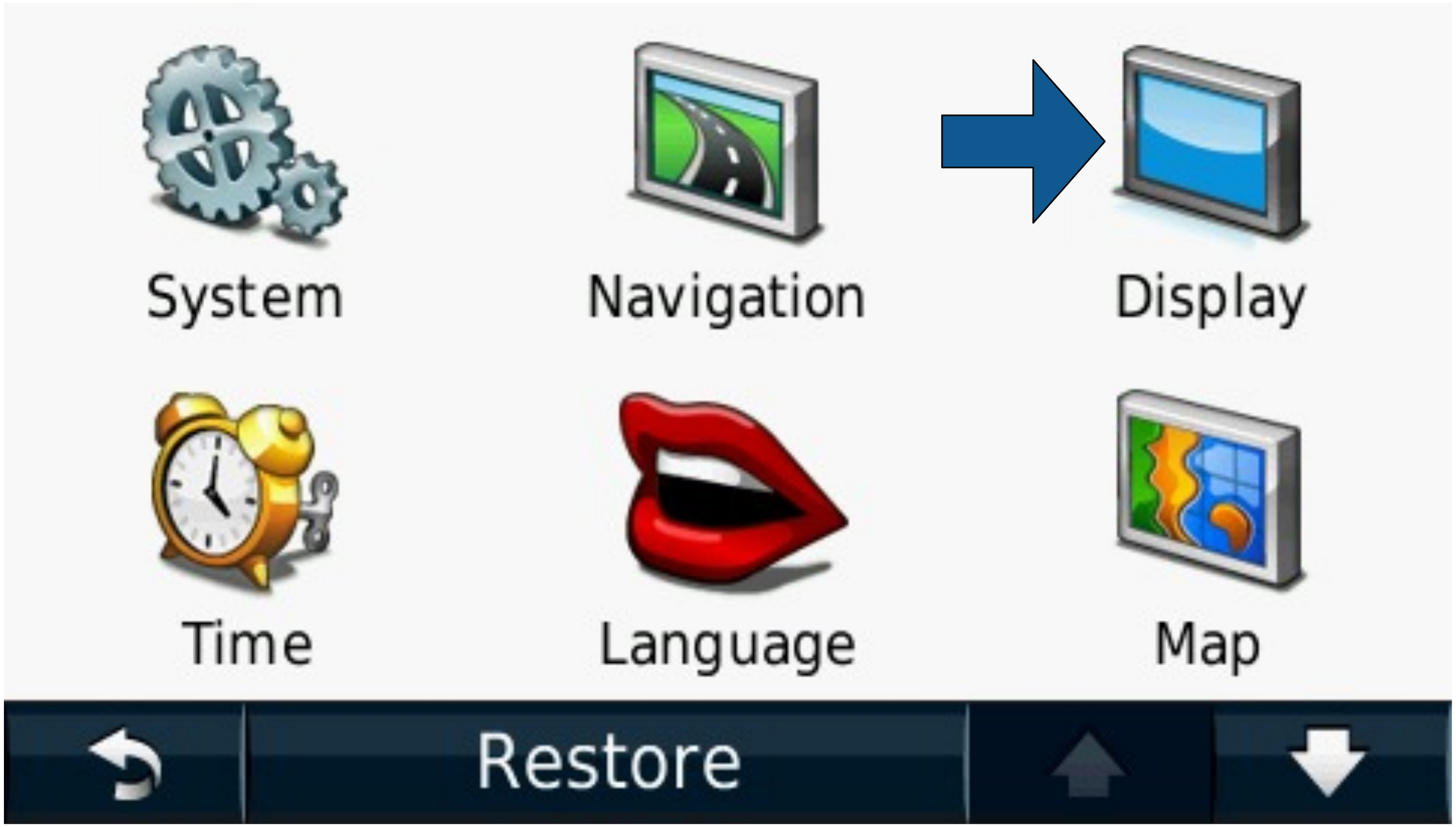


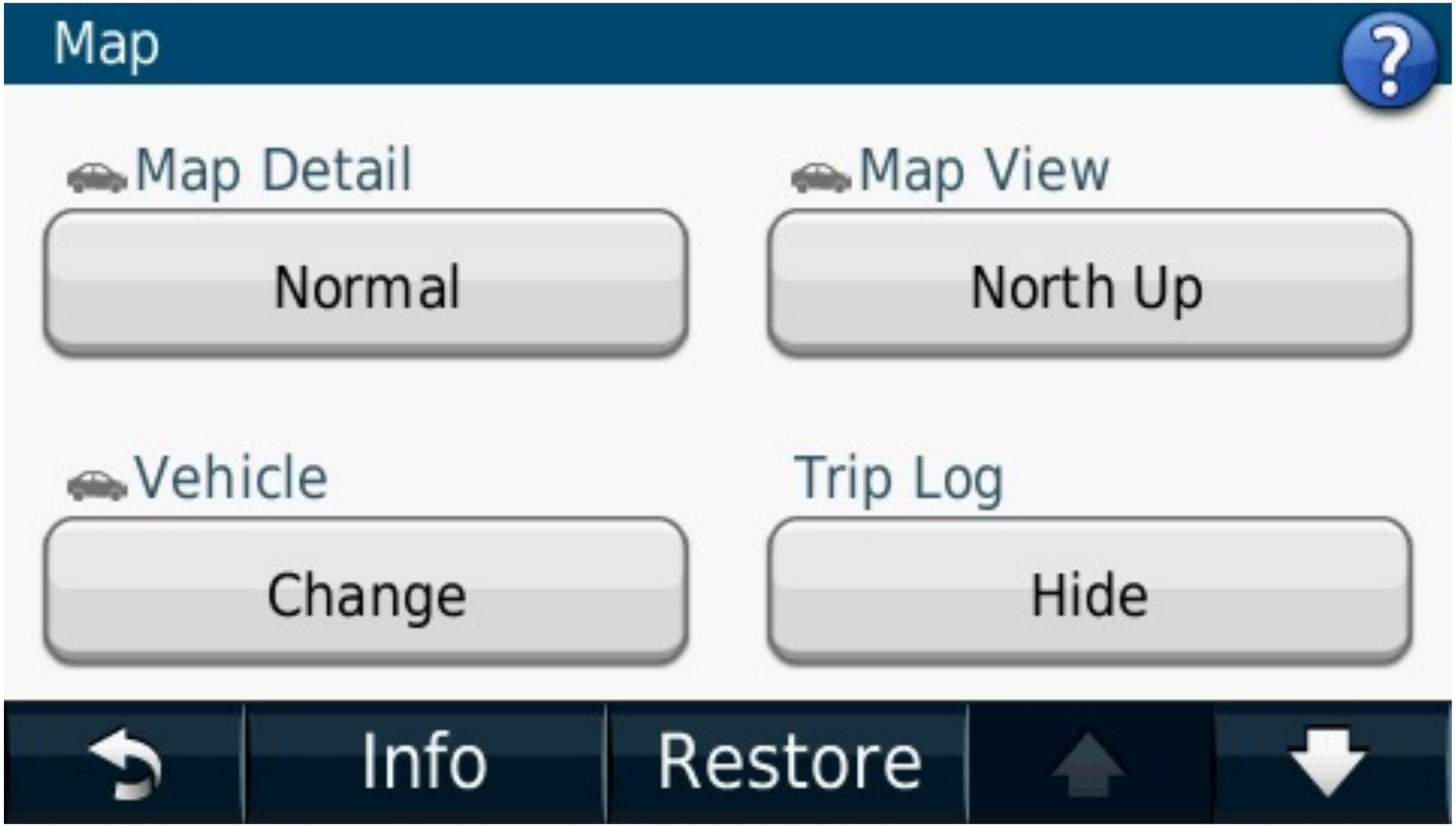
(taken with Garmin's screen capture)

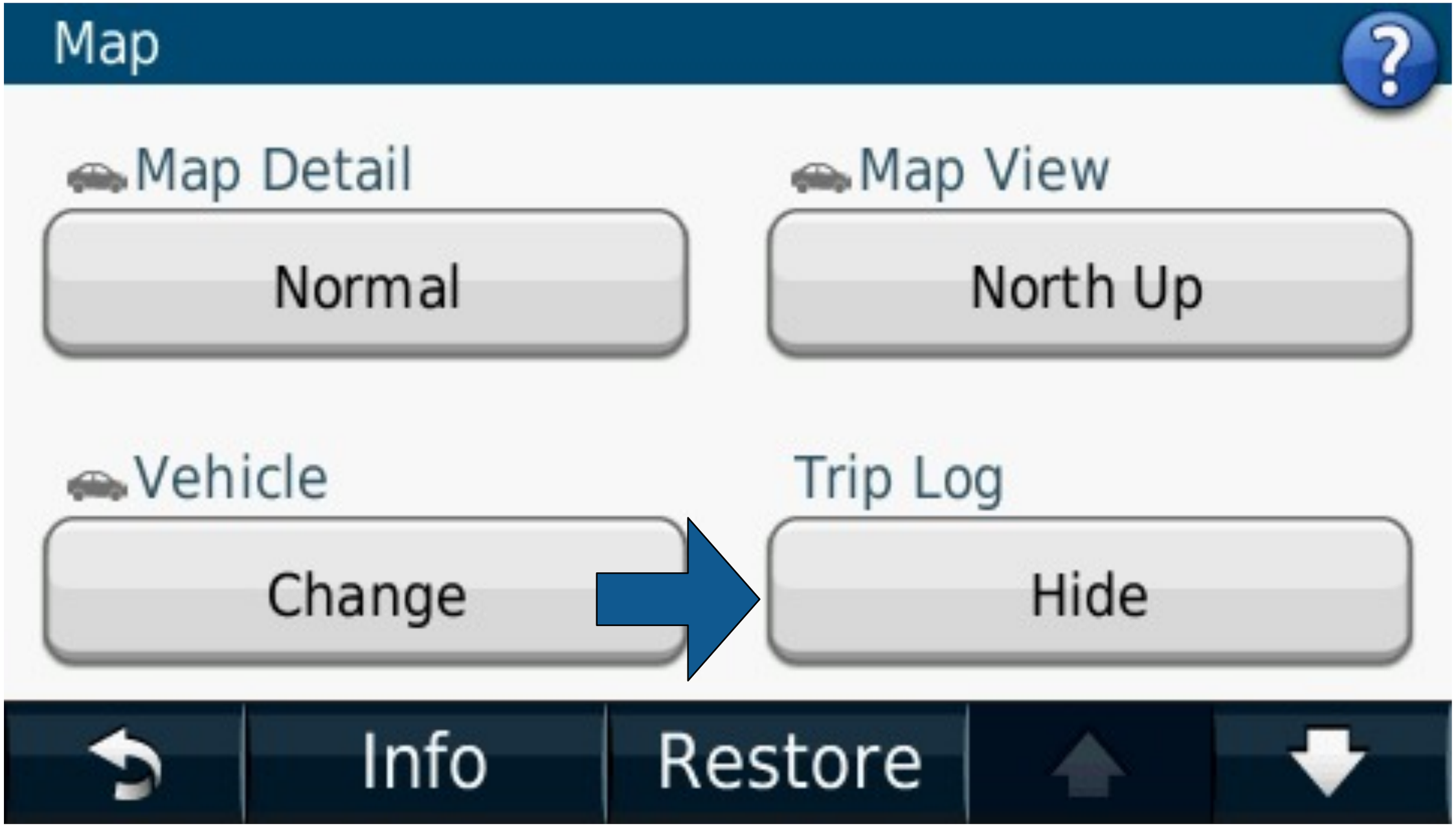

Settings
Where Am I?
Help
ecoRoute™
Picture Viewer
My Data

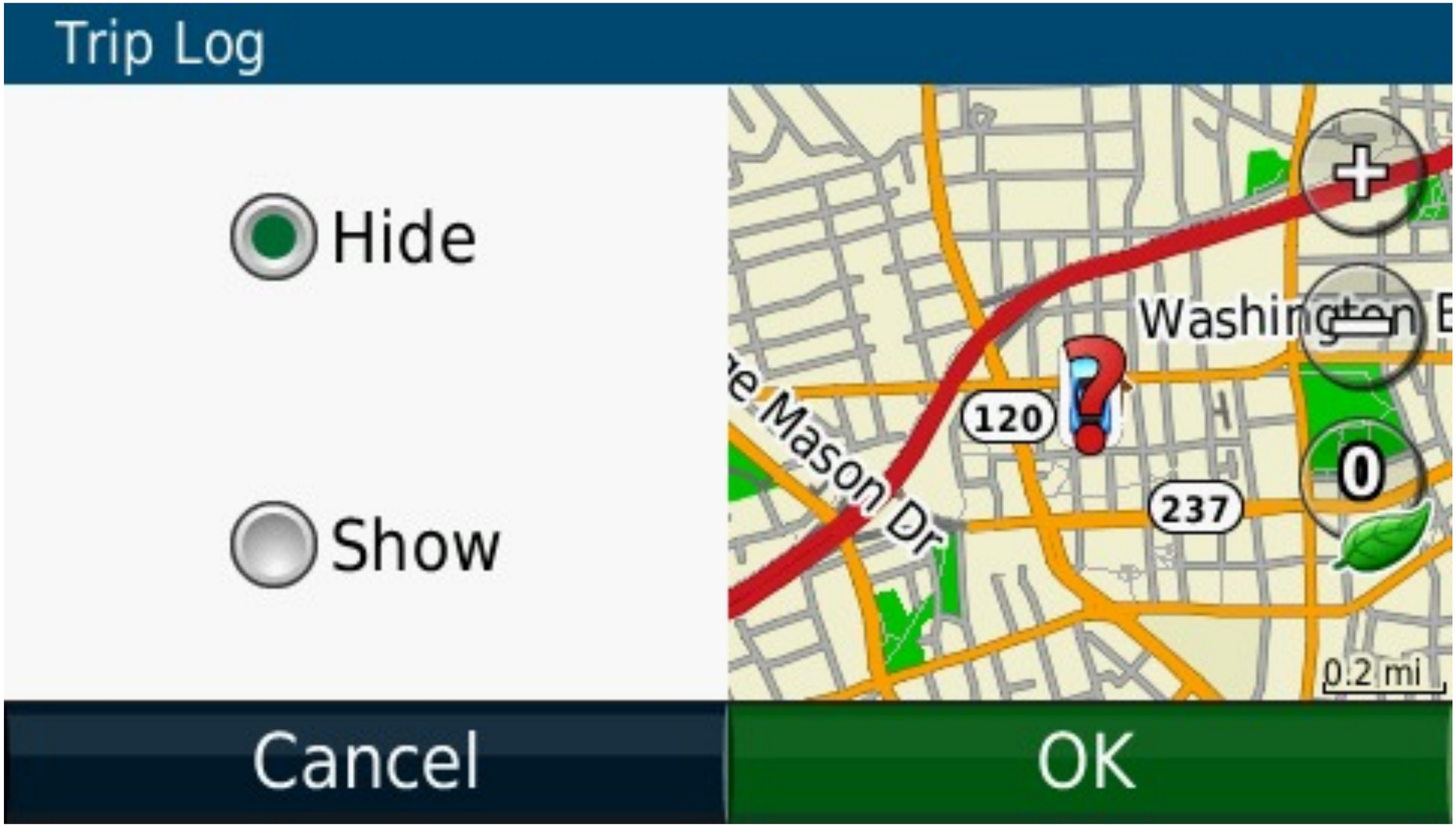

Settings
Where Am I?
Help
ecoRoute™
Picture Viewer
My Data

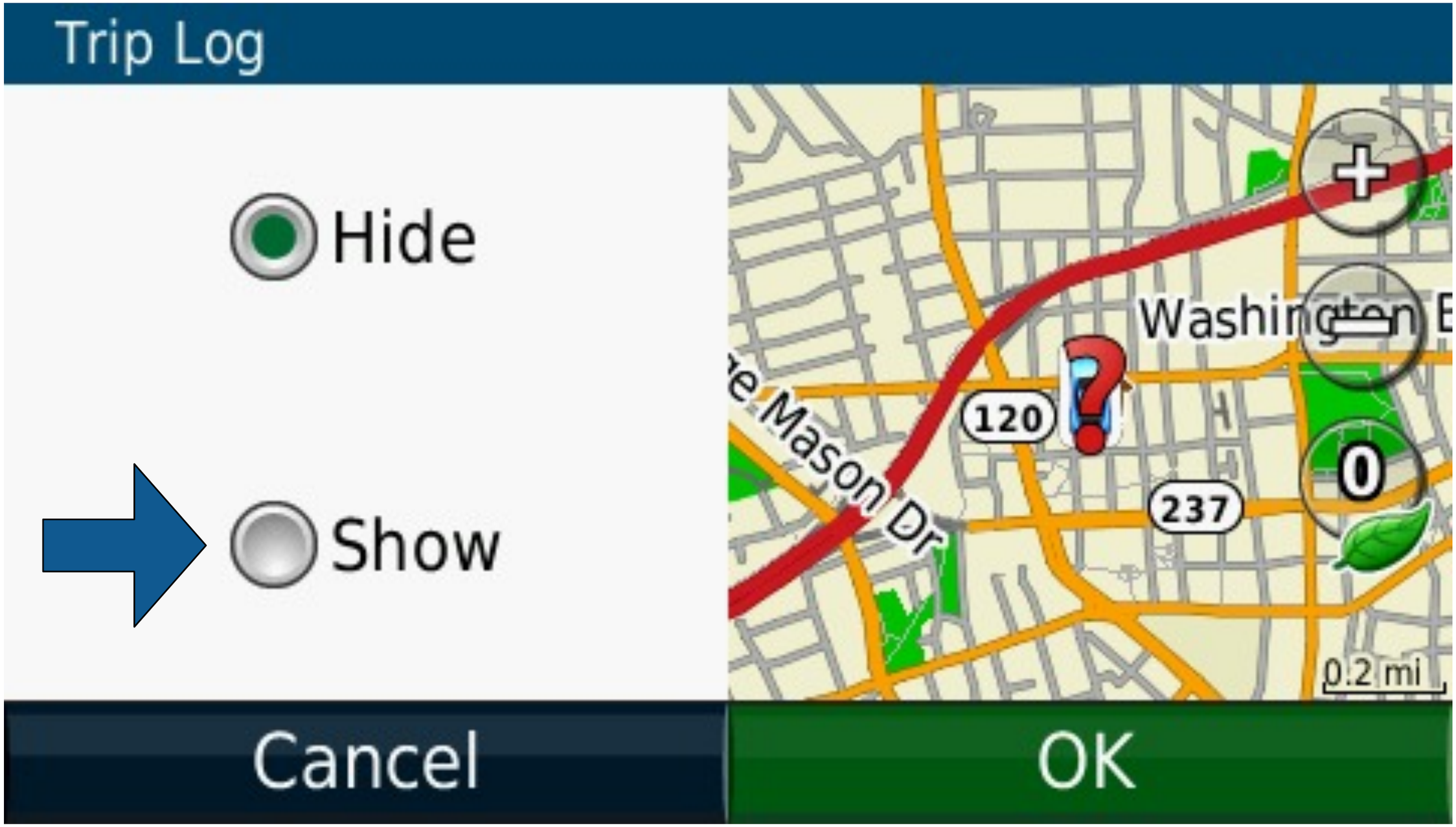

System
Navigation
Display
Time
Language
Map
Restore


System
Navigation
Display
Time
Language
Map
Restore

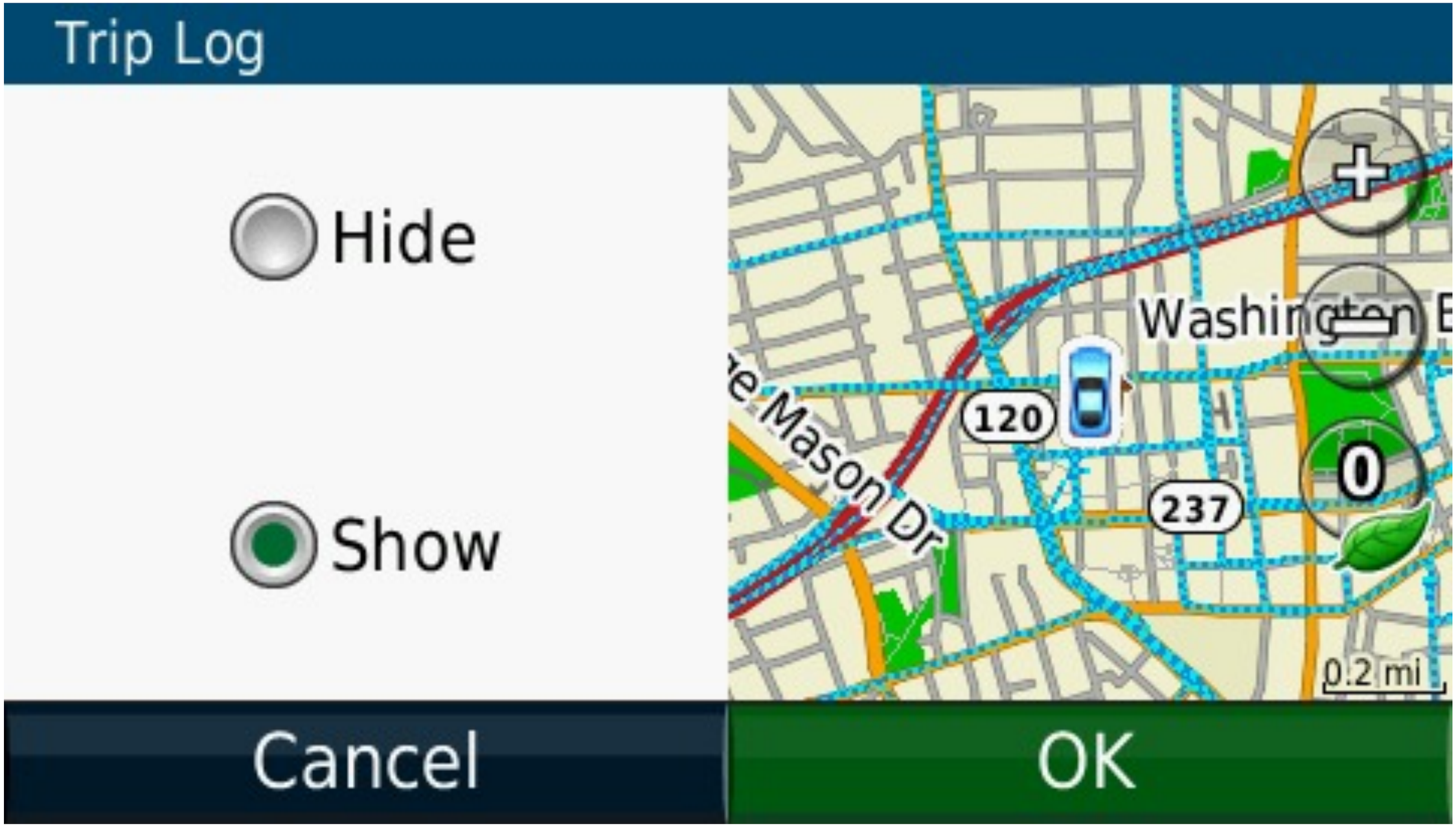

Map
Map Detail
Normal
Map View
North Up
Vehicle
Change
Trip Log
Hide
Info
Restore


Map
Map Detail
Normal
Map View
North Up
Vehicle
Change
Trip Log
Hide
Info
Restore

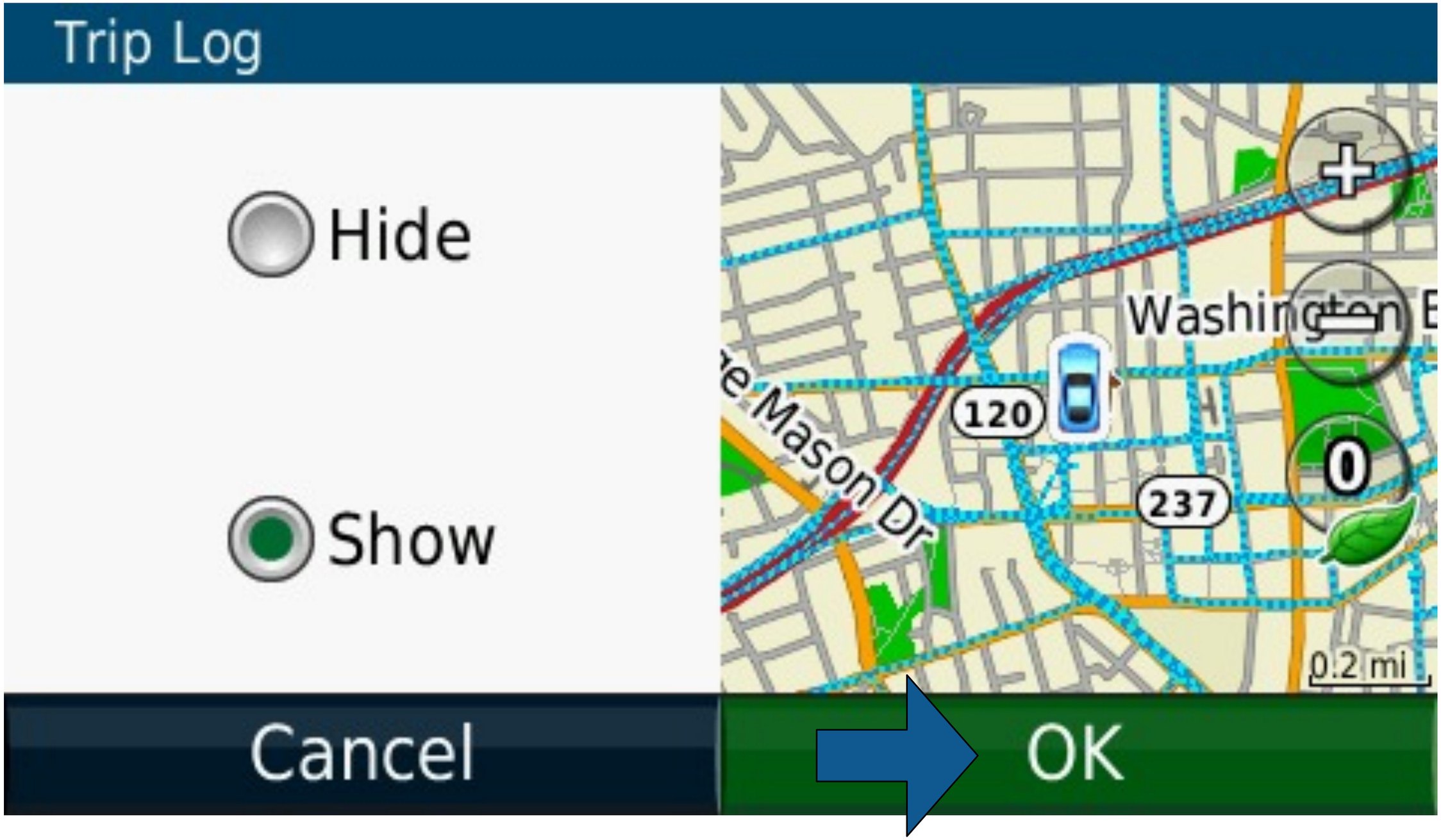

Trip Log
Hide
Show
Washington
Mason Dr
120
237
0
0.2 mi
Cancel
OK


Trip Log
Hide
Show
Washington
Mason Dr
120
237
0
0.2 mi
Cancel
OK

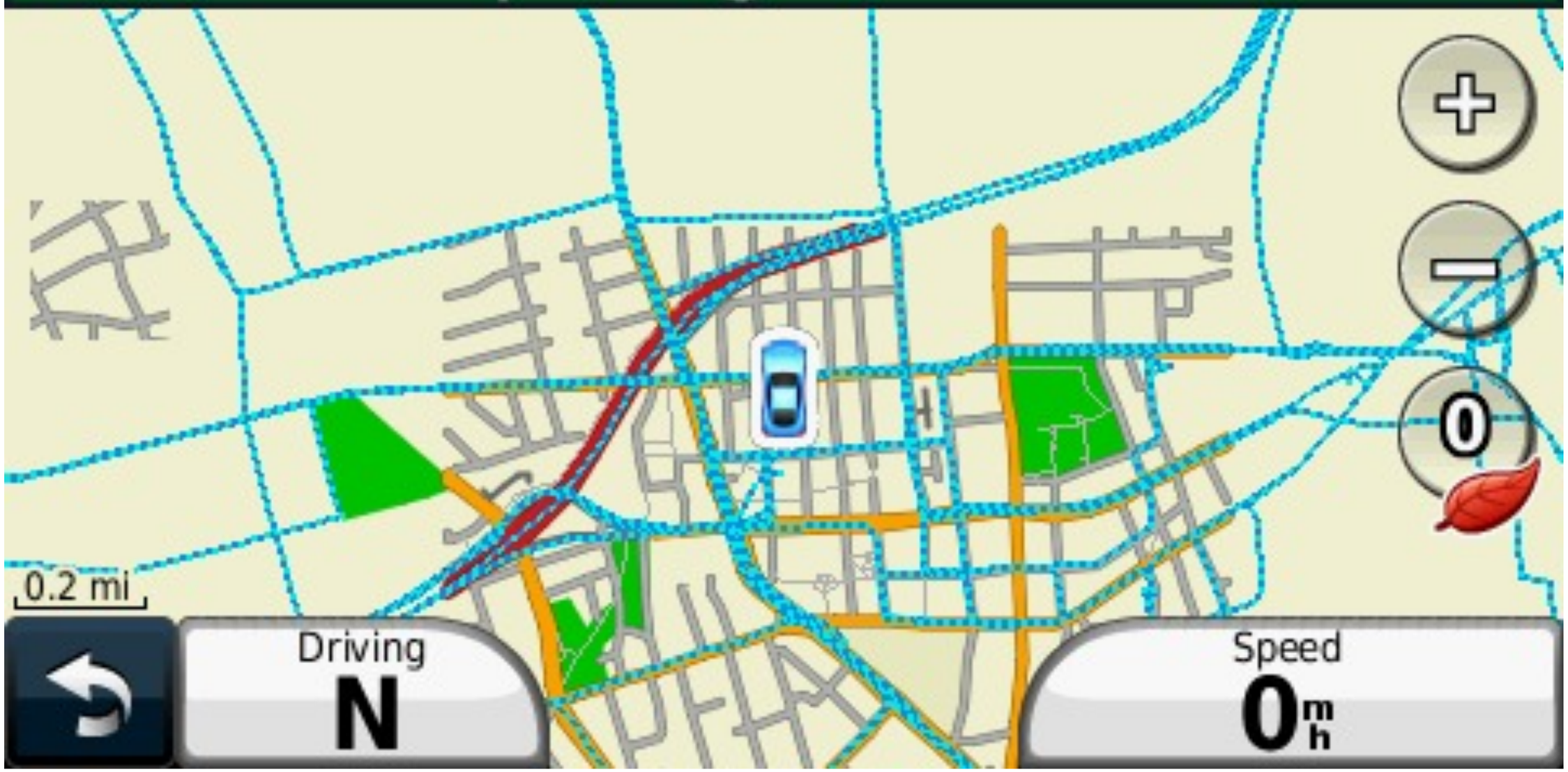
Maps can also show where you have been.



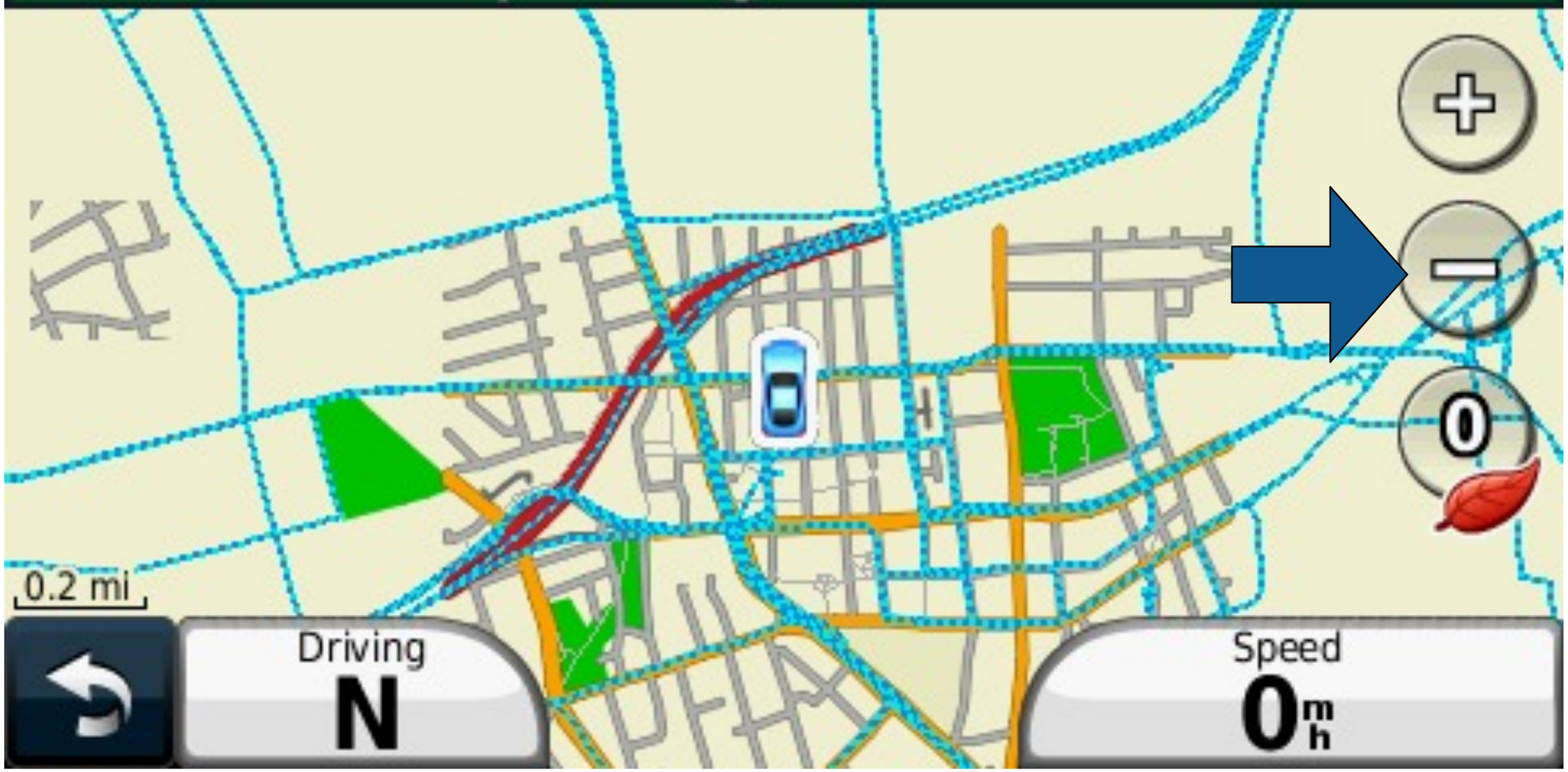
Maps can also show where you have been.



Acquiring Satellites

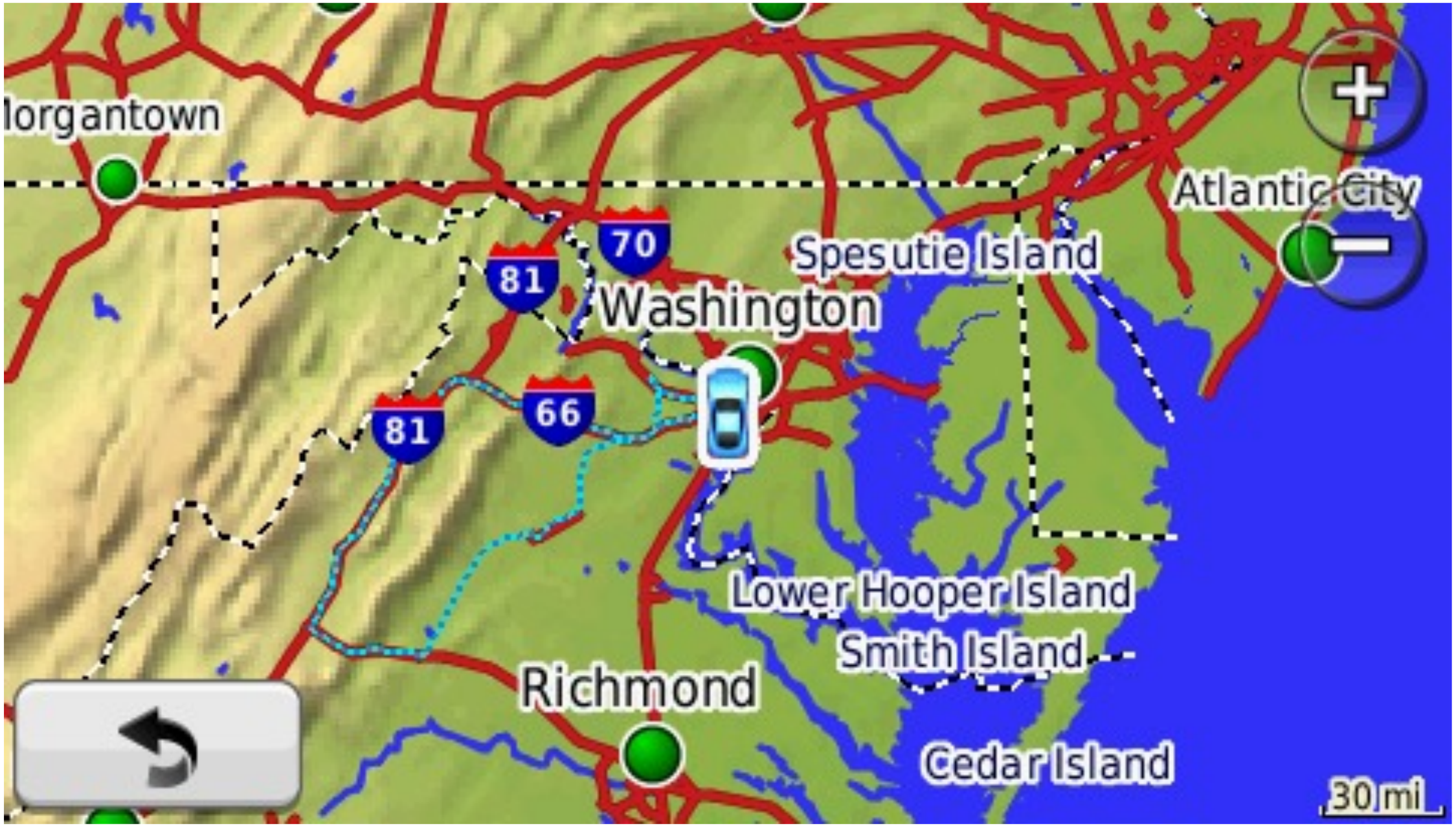


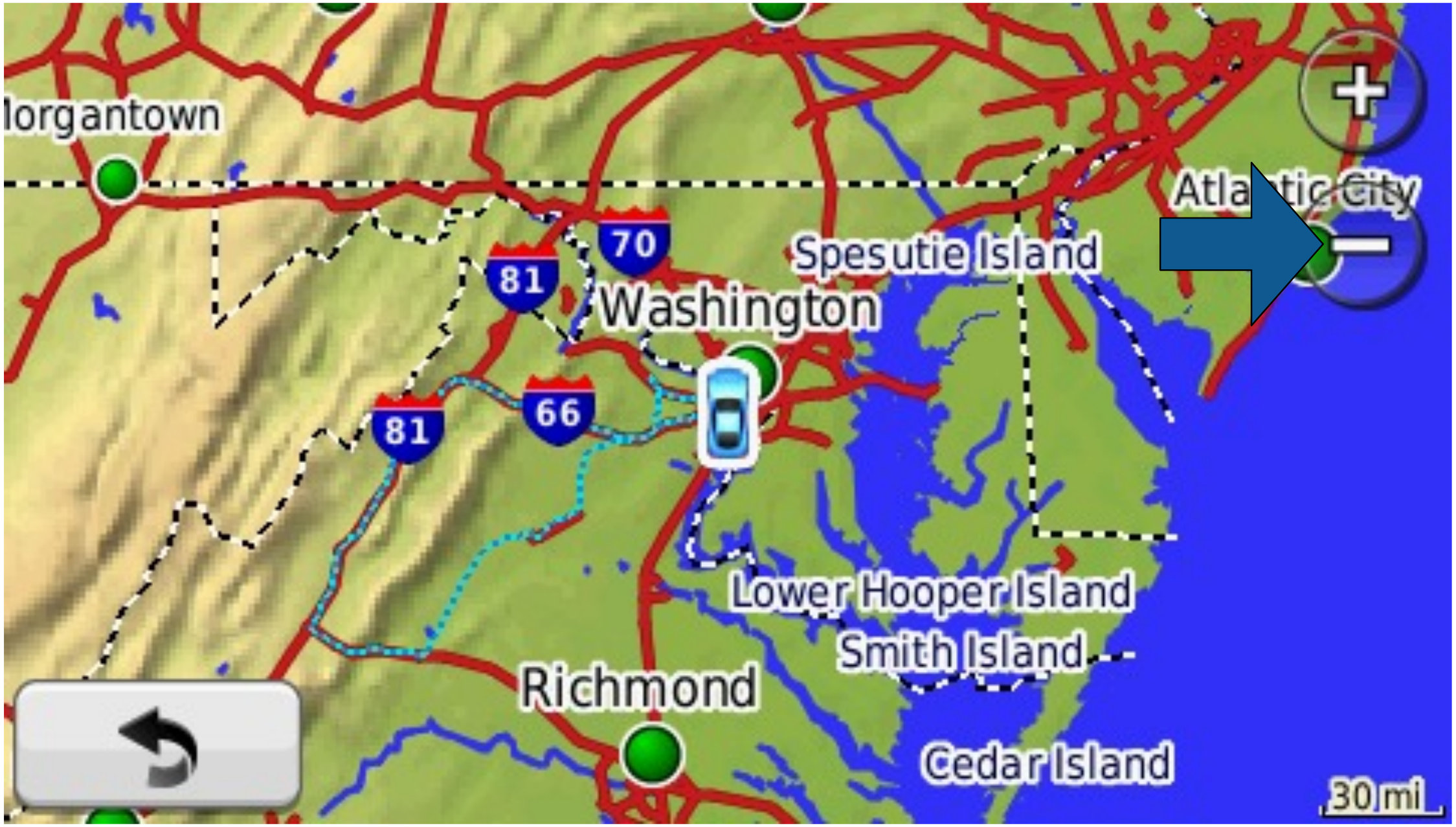
Acquiring Satellites




Ready to Navigate
0.5 mi
Driving
N
Speed
0
m
h


Ready to Navigate
0.5 mi
Driving
N
Speed
0
m
h


Morgantown
Atlantic City
70
81
Spesutie Island
Washington
81
66
Lower Hooper Island
Smith Island
Richmond
Cedar Island
30 mi


Morgantown
Atlantic City
Spesutie Island
Washington
70
81
66
81
Lower Hooper Island
Smith Island
Richmond
Cedar Island
30 mi

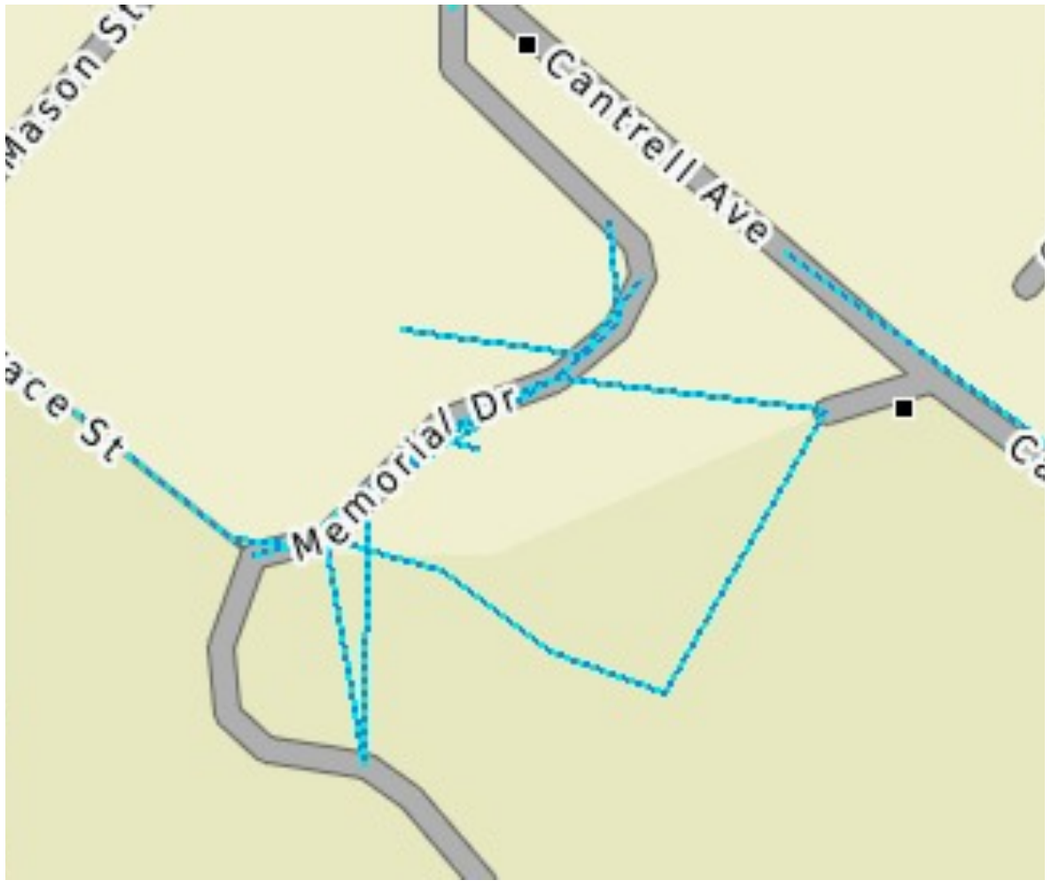

Harrisonburg
Culpeper
Staunton
Charlottesville
8 mi


Harrisonburg
Culpeper
Staunton
Charlottesville
8 mi


Harrisonburg
Massanutten
33
Port Republic
Save
Go!
2 mi

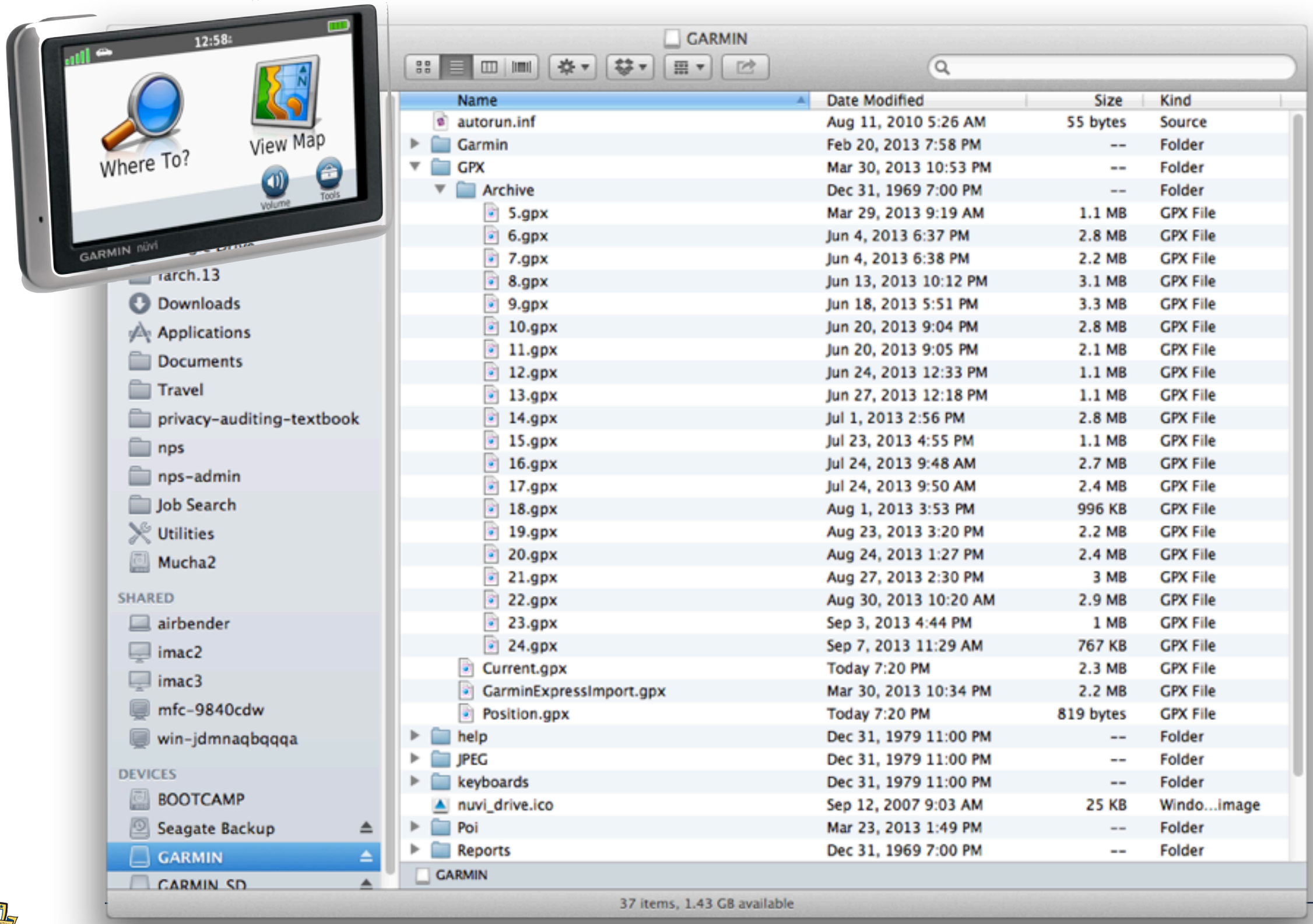

Harrisonburg
Massanutten
33
Port Republic
Save
Go!
2 mi


Save
Go!
S Mason St
Cantrell Ave
Memorial Dr
Cantrell Ave
120 ft

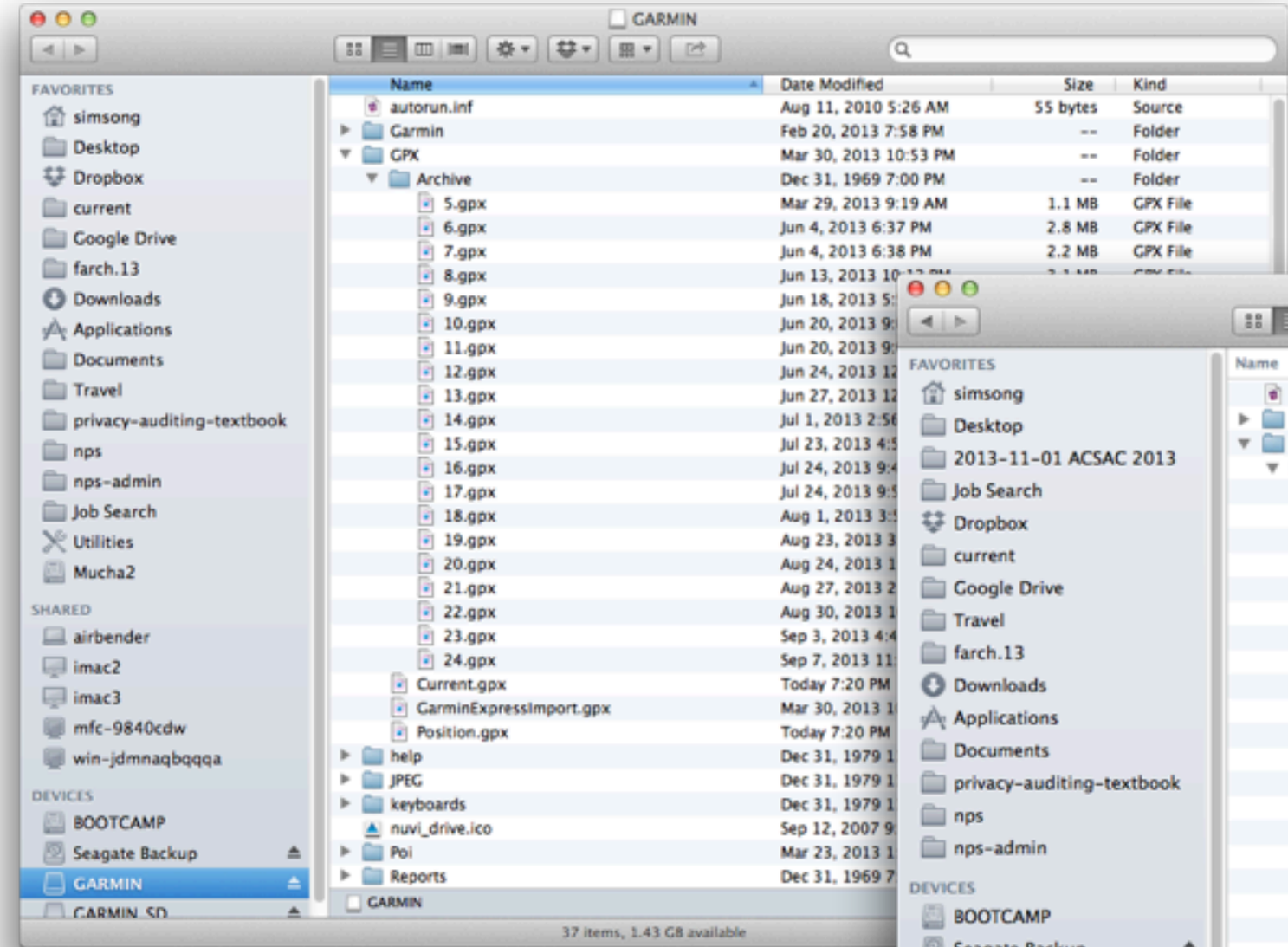

Cantrell Ave
Memorial Dr
ace St


S Mason St
Cantrell Ave
Cantrell Ave
E Grace St
E Grace St

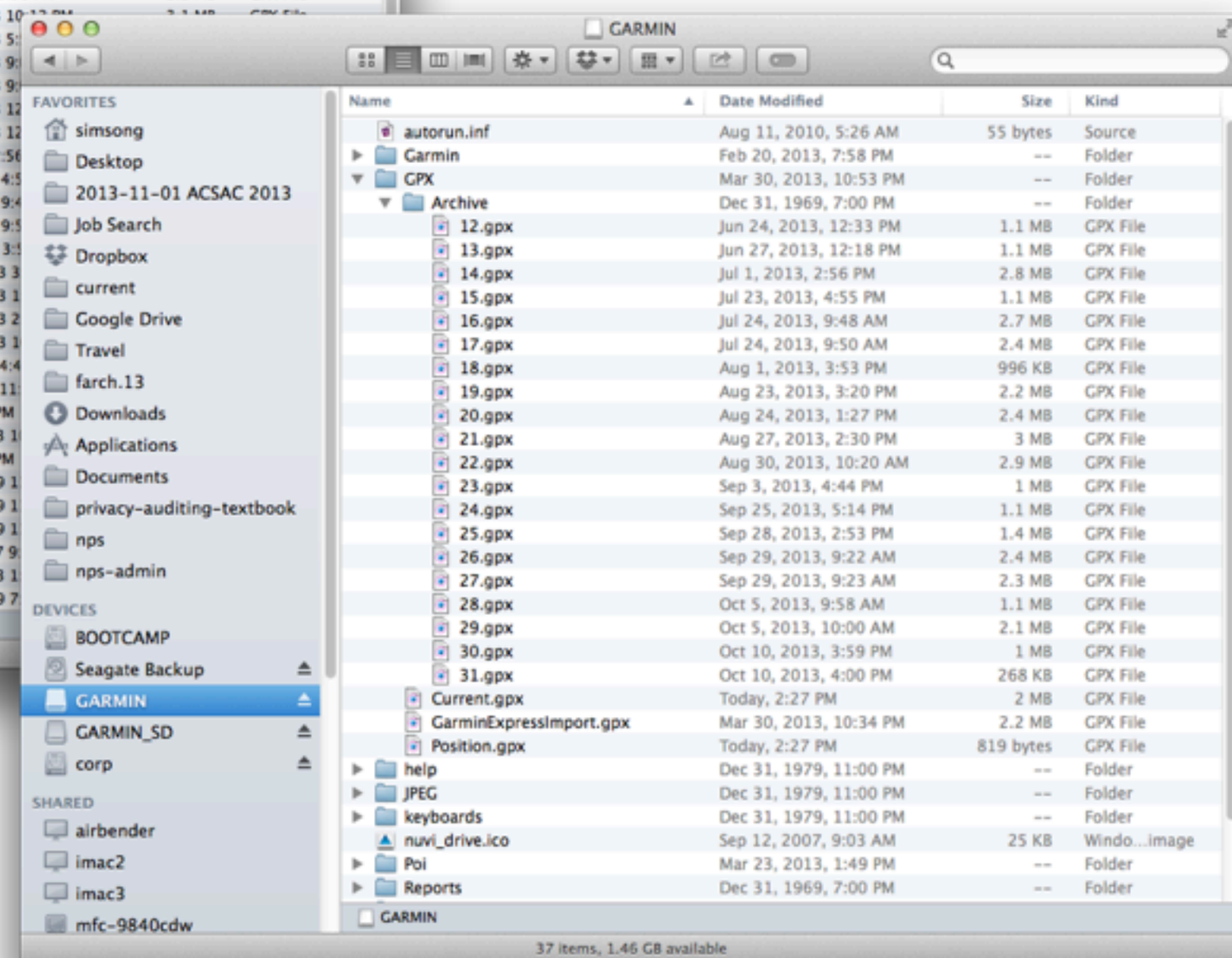
The Garmin GPS appears as an external USB storage device with directories and files.



My Nüvi *always* has 20 files in the /GPS/Archive directory



Sep 7, 2013



Oct 12, 2013

Is there a privacy leak?
Technical Privacy Auditing (TAP) is a way to find out.

Questions:

- Is there information present that we can't see?
- Is this information potentially damaging?

Technical Privacy Auditing

1 TARGET — What are we examining? Why?

2 RESEARCH — What is the format? Are there available tools?

3 COLLECT — Obtain exemplars, ideally from multiple instances

4 ANALYZE — Extract Features; Look for oddities & outliers

5 EXPERIMENT — Show that you understand what is happening

6 REPLICATE — Ideally on another system

7 REPORT — Share results in a concise & understandable form

The file 24.gpx contains track information in XML format.

23.gpx	Sep 3, 2013 4:44 PM	1 MB	GPX File
24.gpx	Sep 7, 2013 11:29 AM	767 KB	GPX File
Current.gpx	Today 7:20 PM	2.3 MB	GPX File


```
om/
xmlschemas/GpxExtensions/v3 http://www.garmin.com/xmlschemas/GpxExtensionsv3.xsd http://www.garmin.com/xmlschemas/
TrackPointExtension/v2 http://www.garmin.com/xmlschemas/TrackPointExtensionv2.xsd"><metadata><link href="http://
www.garmin.com"><text>Garmin International</text></link><time>2013-09-03T20:44:09Z</time>
</metadata><trk><name>ACTIVE LOG: 23 AUG 2013 10:47</name><trkseg><trkpt lat="38.885255" lon="-77.114185"><ele>99.63</
ele><time>2013-08-23T14:47:26Z</time><extensions><gpxtpx:TrackPointExtension><gpxtpx:course>0.00</gpxtpx:course></
gpxtpx:TrackPointExtension></extensions></trkpt><trkpt lat="38.885206" lon="-77.114113"><ele>89.54</
ele><time>2013-08-23T14:47:49Z</time><extensions><gpxtpx:TrackPointExtension><gpxtpx:speed>1.37</
gpxtpx:speed><gpxtpx:course>0.00</gpxtpx:course></gpxtpx:TrackPointExtension></extensions></trkpt><trkpt
lat="38.885058" lon="-77.114068"><ele>88.58</ele><time>2013-08-23T14:47:54Z</
time><extensions><gpxtpx:TrackPointExtension><gpxtpx:speed>5.49</gpxtpx:speed><gpxtpx:course>180.71</gpxtpx:course></
gpxtpx:TrackPointExtension></extensions></trkpt><trkpt lat="38.885008" lon="-77.114062"><ele>88.10</
ele><time>2013-08-23T14:47:55Z</time><extensions><gpxtpx:TrackPointExtension><gpxtpx:speed>5.49</
gpxtpx:speed><gpxtpx:course>177.88</gpxtpx:course></gpxtpx:TrackPointExtension></extensions></trkpt><trkpt
lat="38.884954" lon="-77.114056"><ele>88.10</ele><time>2013-08-23T14:47:56Z</
time><extensions><gpxtpx:TrackPointExtension><gpxtpx:speed>6.86</gpxtpx:speed><gpxtpx:course>179.29</gpxtpx:course></
gpxtpx:TrackPointExtension></extensions></trkpt><trkpt lat="38.884389" lon="-77.113991"><ele>85.69</
ele><time>2013-08-23T14:48:05Z</time><extensions><gpxtpx:TrackPointExtension><gpxtpx:speed>6.86</
gpxtpx:speed><gpxtpx:course>182.12</gpxtpx:course></gpxtpx:TrackPointExtension></extensions></trkpt><trkpt
lat="38.884175" lon="-77.113967"><ele>85.21</ele><time>2013-08-23T14:48:09Z</
time><extensions><gpxtpx:TrackPointExtension><gpxtpx:speed>5.49</gpxtpx:speed><gpxtpx:course>184.94</gpxtpx:course></
gpxtpx:TrackPointExtension></extensions></trkpt><trkpt lat="38.883740" lon="-77.114060"><ele>82.33</
ele><time>2013-08-23T14:48:19Z</time><extensions><gpxtpx:TrackPointExtension><gpxtpx:speed>5.49</
gpxtpx:speed><gpxtpx:course>254.12</gpxtpx:course></gpxtpx:TrackPointExtension></extensions></trkpt><trkpt
lat="38.883736" lon="-77.114121"><ele>82.33</ele><time>2013-08-23T14:48:20Z</
time><extensions><gpxtpx:TrackPointExtension><gpxtpx:speed>5.49</gpxtpx:speed><gpxtpx:course>266.82</gpxtpx:course></
gpxtpx:TrackPointExtension></extensions></trkpt><trkpt lat="38.883690" lon="-77.114860"><ele>80.41</
ele><time>2013-08-23T14:48:31Z</time><extensions><gpxtpx:TrackPointExtension><gpxtpx:speed>4.12</
gpxtpx:speed><gpxtpx:course>265.41</gpxtpx:course></gpxtpx:TrackPointExtension></extensions></trkpt><trkpt
lat="38.883641" lon="-77.115661"><ele>81.37</ele><time>2013-08-23T14:48:44Z</
time><extensions><gpxtpx:TrackPointExtension><gpxtpx:speed>5.49</gpxtpx:speed><gpxtpx:cour[Mucha ~/Desktop/Garmin]$
```

(Reformatted for improved readability)

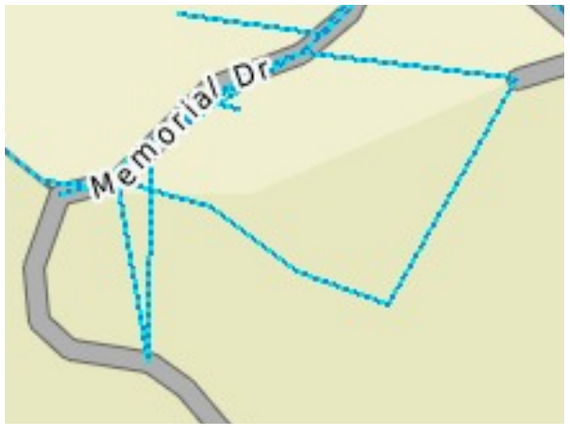
```
<trkpt lat="38.885058" lon="-77.114068">
       <ele>88.58</ele><time>2013-08-23T14:47:54Z</time>
       <gpxtpx:speed>5.49</gpxtpx:speed>
       <gpxtpx:course>180.71</gpxtpx:course>
</trkpt>

<trkpt lat="38.885008" lon="-77.114062">
       <ele>88.10</ele><time>2013-08-23T14:47:55Z</time>
       <gpxtpx:speed>5.49</gpxtpx:speed>
       <gpxtpx:course>177.88</gpxtpx:course>
</trkpt>
```

Accuracy of 0.000001º lat is ≈ 40,000 km ÷ 360 x .000001 ≈ 0.1 m

≈ 10cm

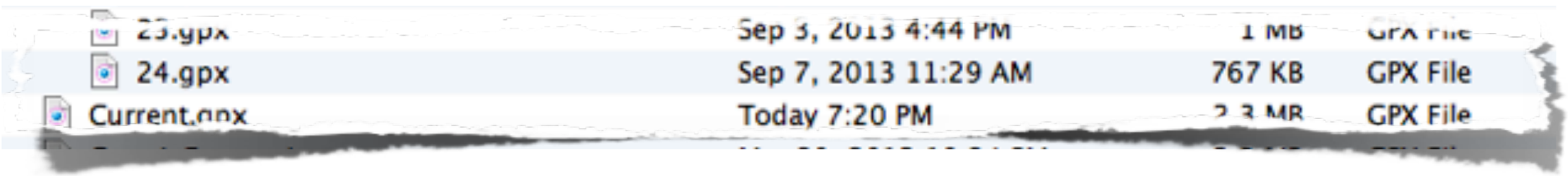
GPS accuracy is 7.8 meters w/ 95% confidence level

- http://www.gps.gov/systems/gps/performance/accuracy/



Each GPX XML record is roughly 266 bytes

```
<ele>88.58</ele><time>2013-08-23T14:47:54Z</
time><extensions><gpxtpx:TrackPointExtension><gpxtpx:speed>5.
49</gpxtpx:speed><gpxtpx:course>180.71</gpxtpx:course></
gpxtpx:TrackPointExtension></extensions></trkpt><trkpt
lat="38.885008" lon="-77.114062"><ele>88.10</ele>
```



The file 24.gpx has 766,553 ÷ 266 ≈ 2,900 tracking points in it.

A simple program gets stats from the GPX files:

```
import xml.sax, os, glob

class SpeedReader(xml.sax.ContentHandler):
    def __init__(self):
        xml.sax.ContentHandler.__init__(self)
        self.cdata = ""
        self.speeds = []

    def startElement(self,name,attrs):
        self.inElement = name
        self.cdata     = ""

    def characters(self,content):
        self.cdata += content

    def endElement(self,name):
        if name=="gpxtpx:speed":
            self.speeds.append(float(self.cdata))

def get_gpx_speeds(fn):
    sr = SpeedReader()
    try:
        xml.sax.parse(open(fn,"rb"),sr)
    except xml.sax._exceptions.SAXParseException:
        pass
    print(fn,min(sr.speeds),max(sr.speeds),len(sr.speeds))


if __name__=="__main__":
    for fn in glob.glob("*.gpx"):
        get_gpx_speeds(fn)
```

```
$ python crunch.py
12.gpx 1.37 28.82 4188
13.gpx 1.37 28.82 4015
14.gpx 1.37 30.2 10095
15.gpx 1.37 28.82 3827
16.gpx 1.37 34.31 9830
17.gpx 1.37 34.31 8664
18.gpx 1.37 26.08 3405
19.gpx 1.37 34.31 8082
20.gpx 1.37 32.94 9023
21.gpx 1.37 32.94 11081
22.gpx 1.37 32.94 10880
23.gpx 1.37 32.94 3822
24.gpx 1.37 32.94 3964
25.gpx 1.37 34.31 5045
26.gpx 1.37 34.31 8826
27.gpx 1.37 34.31 8649
28.gpx 1.37 34.31 3907
29.gpx 1.37 28.82 7700
30.gpx 1.37 28.82 3774
31.gpx 1.37 17.84 996
```

129,773 records
20 files

The Sleuth Kit (TSK) show no deleted files in the /GPX/Archive directory.

```
$ fls -r SLG-GARMAIN.E01  28
d/d 28:   GPX
+ r/r 54419718:   Current.gpx
+ d/d 54419720:   Archive
++ r/r 57598982:  21.gpx
++ r/r 57598984:  22.gpx
++ r/r 57598986:  23.gpx
++ r/r 57598988:  24.gpx
++ r/r 57598990:  25.gpx
++ r/r 57598992:  26.gpx
++ r/r 57598994:  27.gpx
++ r/r 57598996:  28.gpx
++ r/r 57598998:  29.gpx
++ r/r 57599000:  30.gpx
++ r/r 57599002:  31.gpx
++ r/r 57599004:  12.gpx
++ r/r 57599006:  13.gpx
++ r/r 57599008:  14.gpx
++ r/r 57599010:  15.gpx
++ r/r 57599012:  16.gpx
++ r/r 57599014:  17.gpx
++ r/r 57599016:  18.gpx
++ r/r 57599018:  19.gpx
++ r/r 57599020:  20.gpx
```

```
+++++ d/d 23676-144-1:   CrashReports
+++++ r/d * 23442-48-2(realloc):  GoogleUpdateHelper.msi
+++++ r/d * 23442-144-1(realloc): GoogleUpdateHelper.msi
+++++ r/d * 23439-48-2(realloc):  goopdate.dll
+++++ r/d * 23439-144-1(realloc): goopdate.dll
+++++ r/d * 23441-48-2(realloc):  GoopdateBho.dll
+++++ r/d * 23441-144-1(realloc): GoopdateBho.dll
+++++ r/d * 23443-48-2(realloc):  goopdateres_ar.dll
+++++ r/d * 23443-144-5(realloc): goopdateres_ar.dll
+++++ r/d * 23444-48-2(realloc):  goopdateres_bg.dll
+++++ r/d * 23444-144-1(realloc): goopdateres_bg.dll
```

(deleted files on another volume)

It’s unclear when the files get rolled over...
... or where the data goes when it is rolled over.

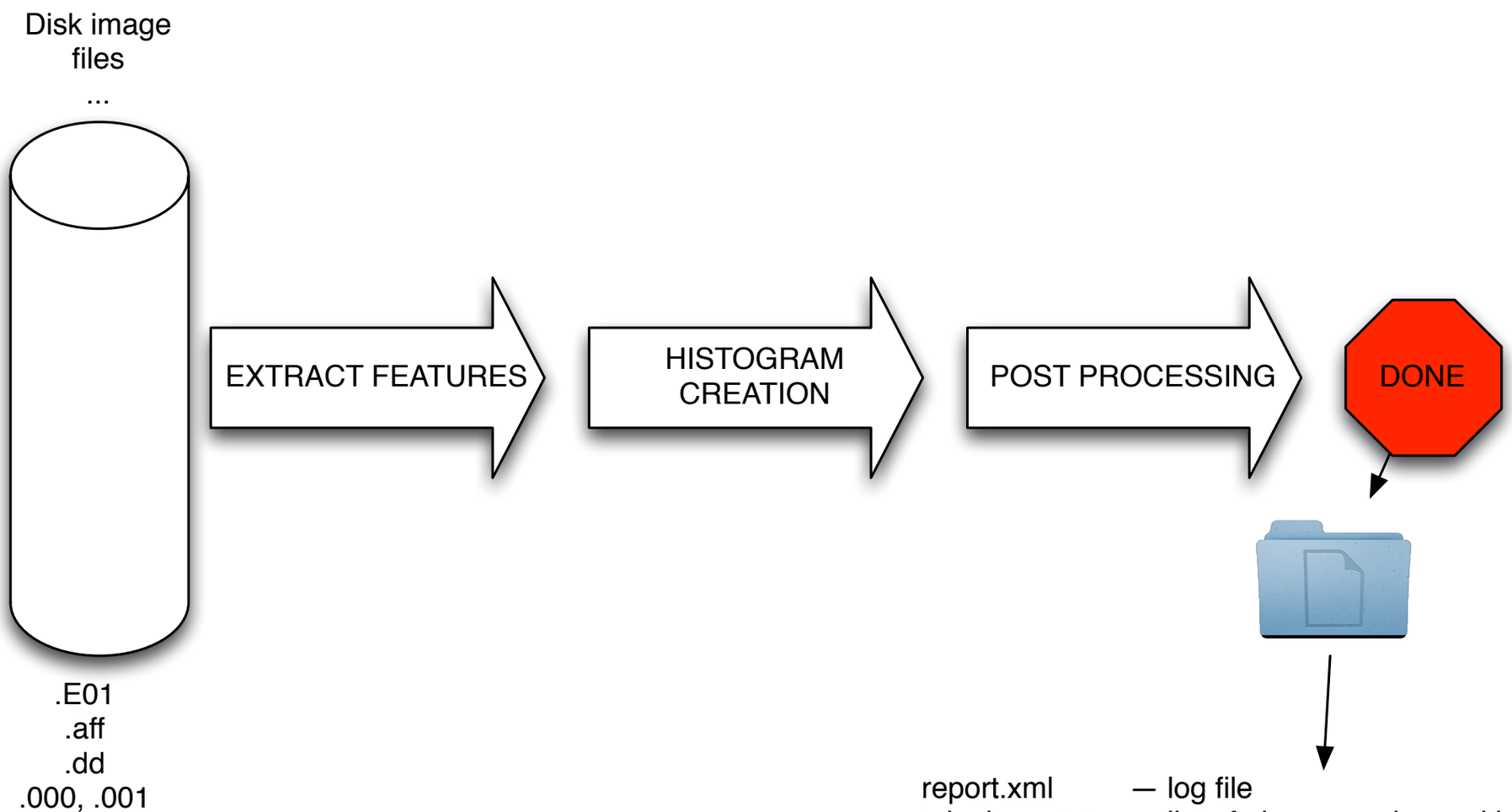
- Archive/5.gpx — 2013-03-07
- ...
- Archive/12.gpx — 2013-06-11 – 2013-06-16
- Archive/13.gpx — 2013-06-17 – 2013-06-21
- Archive/14.gpx — 2013-06-21 – 2013-07-01
- Current.gpx — 2013-10-01 – 2013-10-11

Current also has waypoints and phone numbers...

```
<wpt lat="38.861390" lon="-77.056519"><ele>-0.11</ele><name>Costco</name>
  <desc>1200 S Fern St\nArlington, VA 22202</desc>
  <sym>Department Store</sym>
  <extensions>
    <gpxx:WaypointExtension>
     <gpxx:Categories><gpxx:Category>Shopping</gpxx:Category>
                      </gpxx:Categories>
     <gpxx:Address><gpxx:StreetAddress>1200 S Fern St
                   </gpxx:StreetAddress>
     <gpxx:City>Arlington</gpxx:City><gpxx:State>VA</gpxx:State>
     <gpxx:PostalCode>22202</gpxx:PostalCode></gpxx:Address>
     <gpxx:PhoneNumber>1 7034132324</gpxx:PhoneNumber>
    </gpxx:WaypointExtension>
   </extensions>
</wpt>
```

Where did this come from?

bulk_extractor is a stream forensics program. It finds and extracts “features” from bulk data.



report.xml — log file
telephone.txt — list of phone numbers with context
telephone_histogram.txt — histogram of phone numbers
vcard/ — directory of VCARDs
...

Output is a *directory* containing:

- feature files; histograms; carved objects
- Mostly in UTF-8; some XML
- Can be bundled into a ZIP file and process with bulk_extractor_reader.py

A hex dump shows GPX fragments on sector boundaries.

```
  2510fe0: d0f0 f0d0 f0f0 d0f0 f0d0 f0f0 d0f0 f0d0  ................
2510ff0: f0f0 d0f0 f0d0 f0f0 d0f0 f0d0 f0f0 d0f0  ................
2511000: 6f69 6e74 4578 7465 6e73 696f 6e3e 3c67  ointExtension><g
2511010: 7078 7470 783a 7370 6565 643e 3238 2e38  pxtpx:speed>28.8
2511020: 323c 2f67 7078 7470 783a 7370 6565 643e  2</gpxtpx:speed>
2511030: 3c67 7078 7470 783a 636f 7572 7365 3e32  <gpxtpx:course>2
2511040: 3234 2e34 373c 2f67 7078 7470 783a 636f  24.47</gpxtpx:co
2511050: 7572 7365 3e3c 2f67 7078 7470 783a 5472  urse></gpxtpx:Tr
2511060: 6163 6b50 6f69 6e74 4578 7465 6e73 696f  ackPointExtensio
2511070: 6e3e 3c2f 6578 7465 6e73 696f 6e73 3e3c  n></extensions><
2511080: 2f74 726b 7074 3e3c 7472 6b70 7420 6c61  /trkpt><trkpt la
2511090: 743d 2233 392e 3936 3538 3637 2220 6c6f  t="39.965867" lo
25110a0: 6e3d 222d 3734 2e39 3132 3837 3622 3e3c  n="-74.912876"><
25110b0: 656c 653e 3235 2e31 333c 2f65 6c65 3e3c  ele>25.13</ele><
25110c0: 7469 6d65 3e32 3031 332d 3038 2d31 3854  time>2013-08-18T
25110d0: 3230 3a32 353a 3038 5a3c 2f74 696d 653e  20:25:08Z</time>
25110e0: 3c65 7874 656e 7369 6f6e 733e 3c67 7078  <extensions><gpx
25110f0: 7470 783a 5472 6163 6b50 6f69 6e74 4578  tpx:TrackPointEx
2511100: 7465 6e73 696f 6e3e 3c67 7078 7470 783a  tension><gpxtpx:
```

bulk_extractor has a feature extractor for GPX XML.

```
$ ls -l | cut -c 30- | grep . | grep -v ' 0 '
   133867 Oct 12 18:15 domain.txt
      728 Oct 12 18:15 domain_histogram.txt
     1359 Oct 12 18:15 email.txt
      434 Oct 12 18:15 email_histogram.txt
 12598874 Oct 12 18:15 gps.txt
   188394 Oct 12 18:15 hex.txt
     3102 Oct 12 18:15 json.txt
    28954 Oct 12 18:15 report.xml
    16791 Oct 12 18:15 telephone.txt
     1309 Oct 12 18:15 telephone_histogram.txt
   225582 Oct 12 18:15 url.txt
     3436 Oct 12 18:15 url_histogram.txt
      552 Oct 12 18:15 url_services.txt
  1306682 Oct 12 18:15 windirs.txt
$
$ more gps.txt
# BULK_EXTRACTOR-Version: 1.4.1 ($Rev: 10844 $)
# Feature-Recorder: gps
# Filename: SLG-GARMAIN.E01
# Feature-File-Version: 1.1
83886185        ,,,,,111.53
83886435        2013-06-29T17:31:34Z,38.886777,-77.144822,99.15,10.98,114.3
83886685        2013-06-29T17:31:50Z,38.886160,-77.143053,100.60,8.24,115.7
83886934        2013-06-29T17:32:02Z,38.885877,-77.142233,99.63,5.49,115.76
83887183        2013-06-29T17:32:04Z,38.885841,-77.142125,99.15,4.12,131.29
83887432        2013-06-29T17:32:05Z,38.885827,-77.142082,99.15,4.12,151.06
83887681        2013-06-29T17:32:06Z,38.885727,-77.142003,98.67,5.49,168.00
```

GPS data from raw disk partition

Time | Lat | Lon | Elv | Sp | Hd

Simple string operations on “gps.txt” feature file shows range of private data leakage.

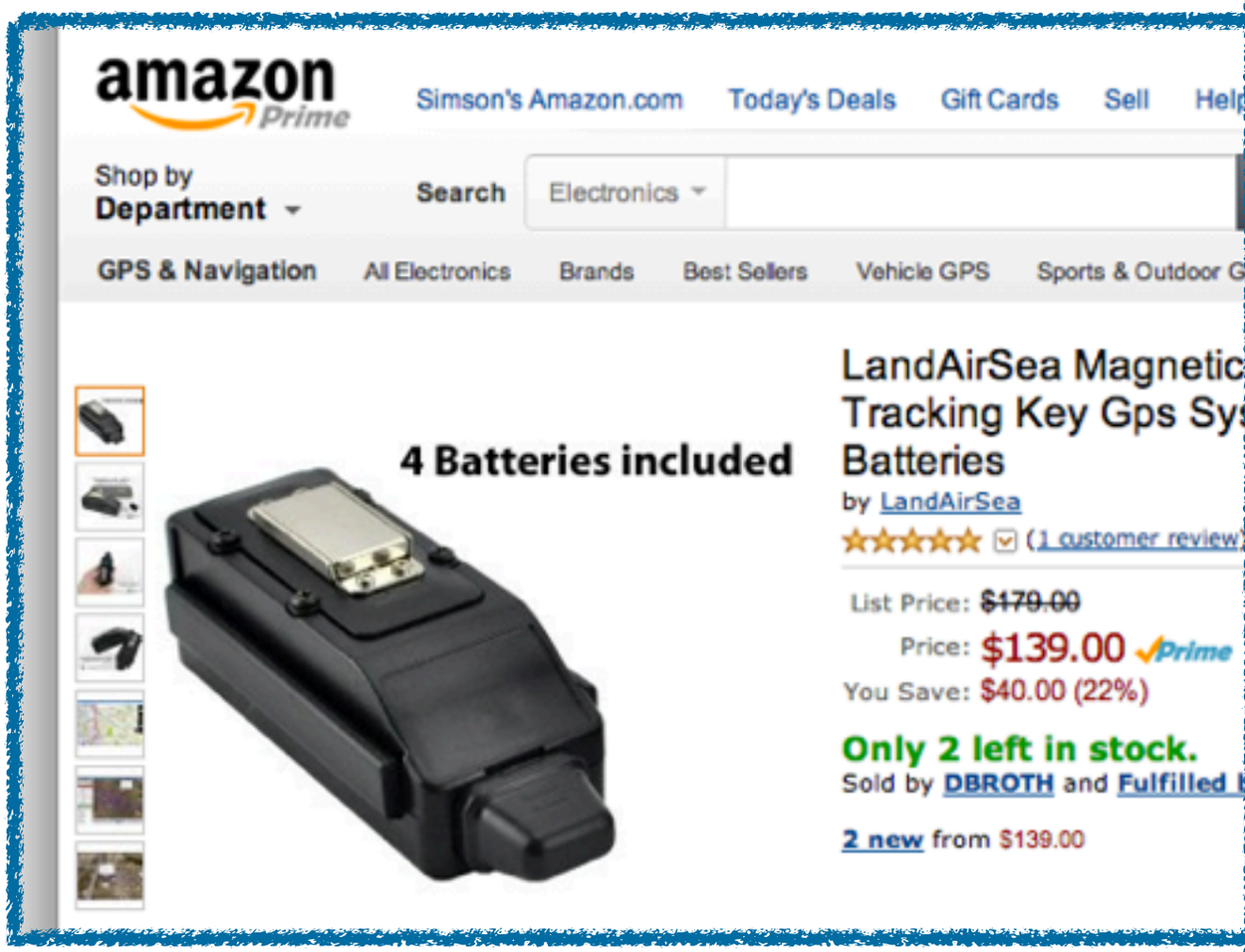
```
$ cut -f 2 gps.txt |grep 2013 | sort | head -5
2013-02-10T22:21:43Z,38.884835,-77.159409,92.42,4.12,354.35
2013-02-10T22:21:57Z,38.885625,-77.159516,94.83,6.86,357.18
2013-02-10T22:22:05Z,38.885845,-77.159546,96.75,2.75,59.29
2013-02-10T22:22:08Z,38.885863,-77.159336,96.75,4.12,104.47
2013-02-10T22:22:09Z,38.885845,-77.159282,96.75,5.49,108.71

$ cut -f 2 gps.txt |grep 2013 | sort | tail -5
2013-10-11T22:32:34Z,38.885033,-77.114435,105.40,,314.82
2013-10-11T22:33:06Z,38.885033,-77.114435,105.40,,132.71
2013-10-12T18:27:26Z,,,,,
2013-10-12T18:27:26Z,,,-0.11,,

$ cut -f 2 gps.txt |grep 2013 | wc -l
  180589
```

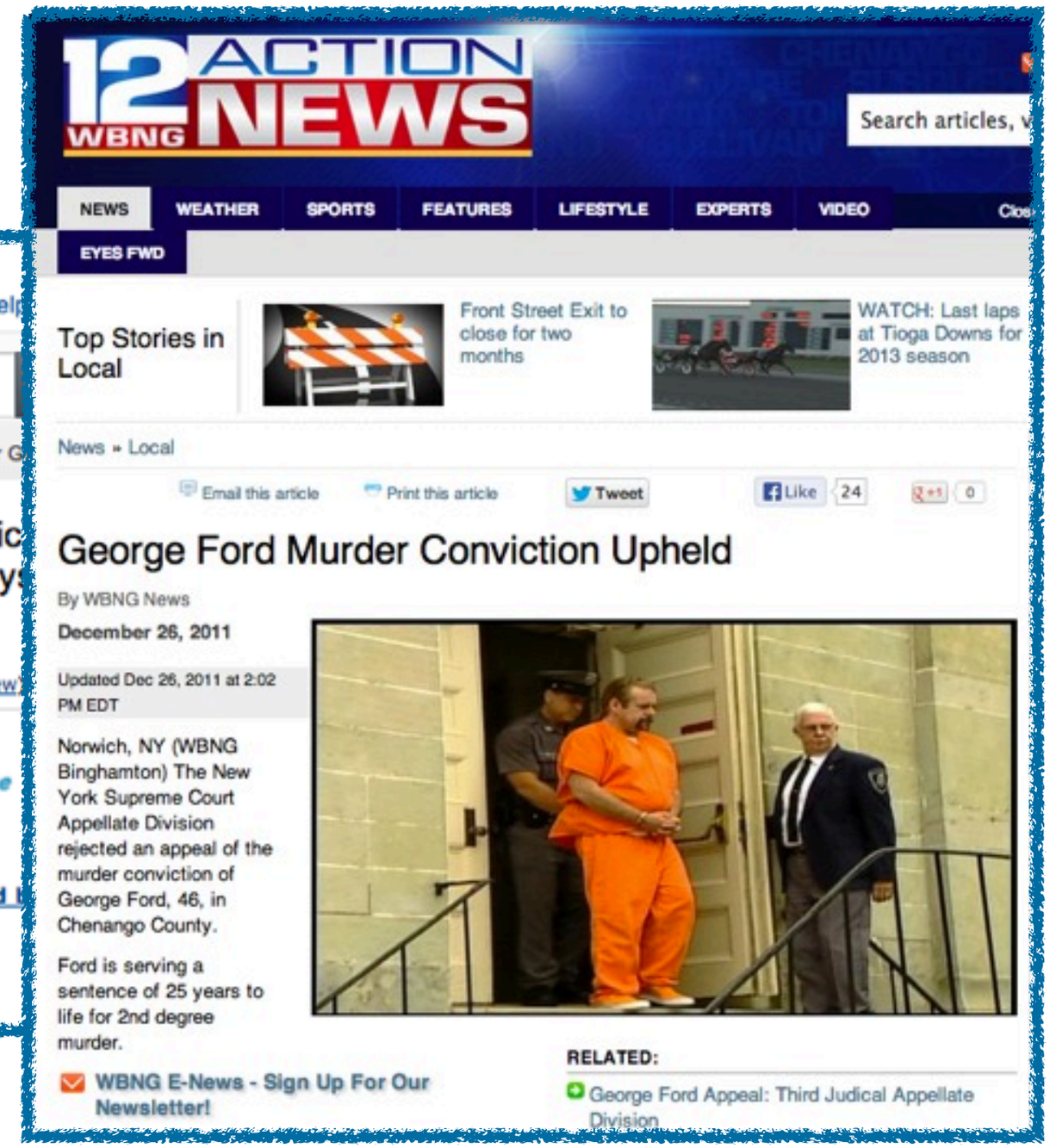
	Allocated	Bulk Data Analysis
Date Range	Feb 10 - Oct 12	June 24 - Oct 12
# Entries	129,773	180,589

Privacy problems for consumers can benefit law enforcement.

George Ford Murder trial.

- Killing of Shyanne Somers, 12, on July 8, 2007
- Convicted on 2009
- Key evidence: GPS tracking data





My research focus: better tools and algorithms for triage.

Identification of high-value data.

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

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

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

Presentation and Integration:

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

Today's tools frequently miss case-critical data.

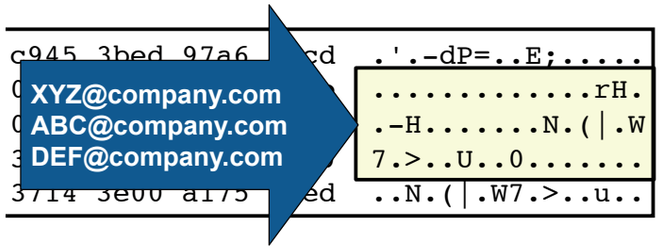
Like GPS information, email addresses are of significant forensic interest.

38.885034,-77.114424

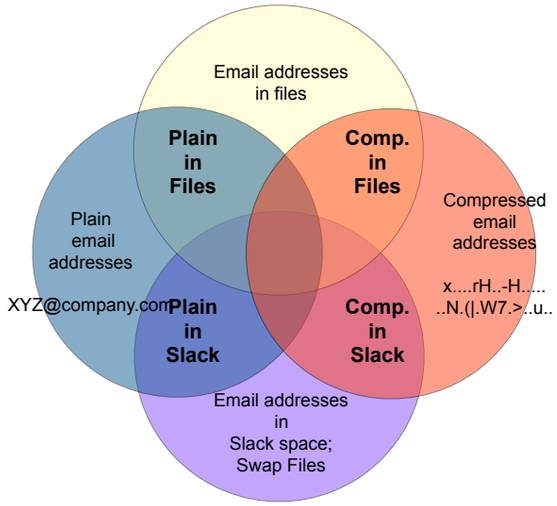
ABC@company.com

Email addresses can be compressed.

Popular forensic tools do not optimistically decompress.



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



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

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

Email addresses can link devices together

Email addresses can reveal:

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

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

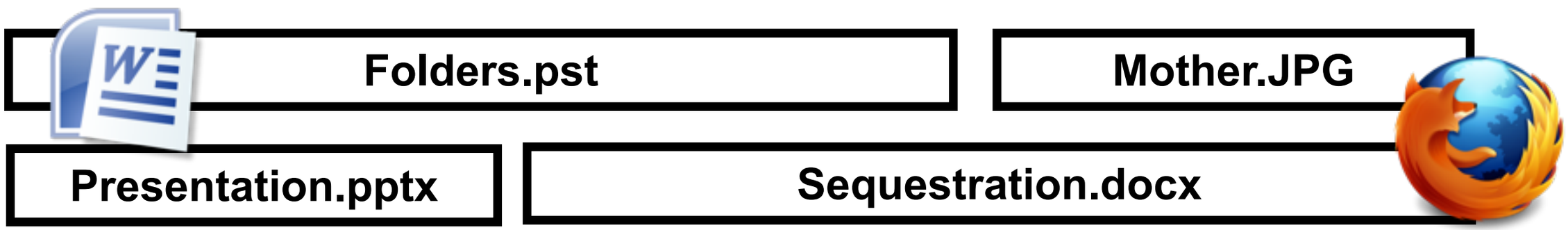
Correlation requires feature extraction

Today's forensic tools implement two strategies for extracting email addresses.

1. *Text extraction from files*
2. *Text extraction from bulk data*

Email addresses are extracted from *document files* by converting to text then scanning with regular expressions.

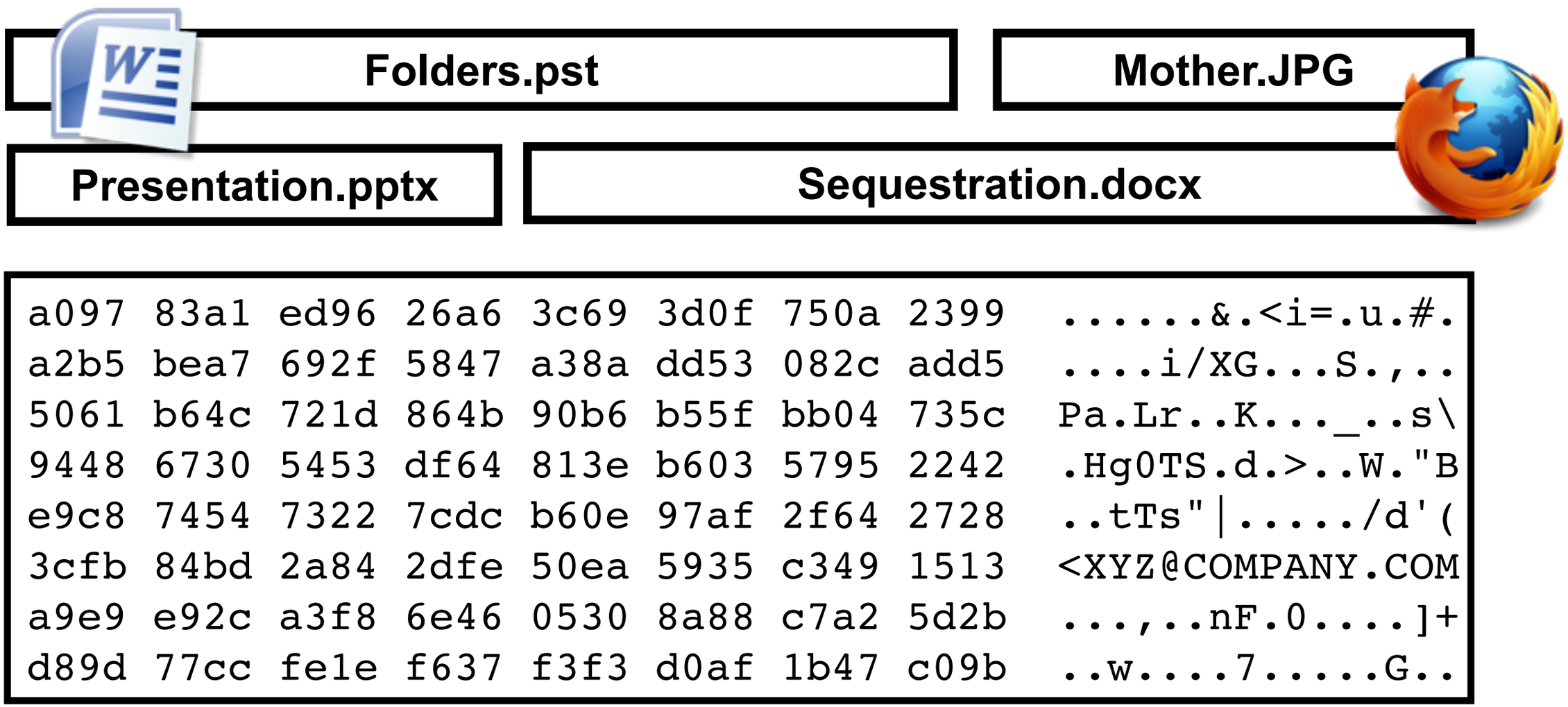
File ➛ Text ➛ RegEx ➛ Email Addresses



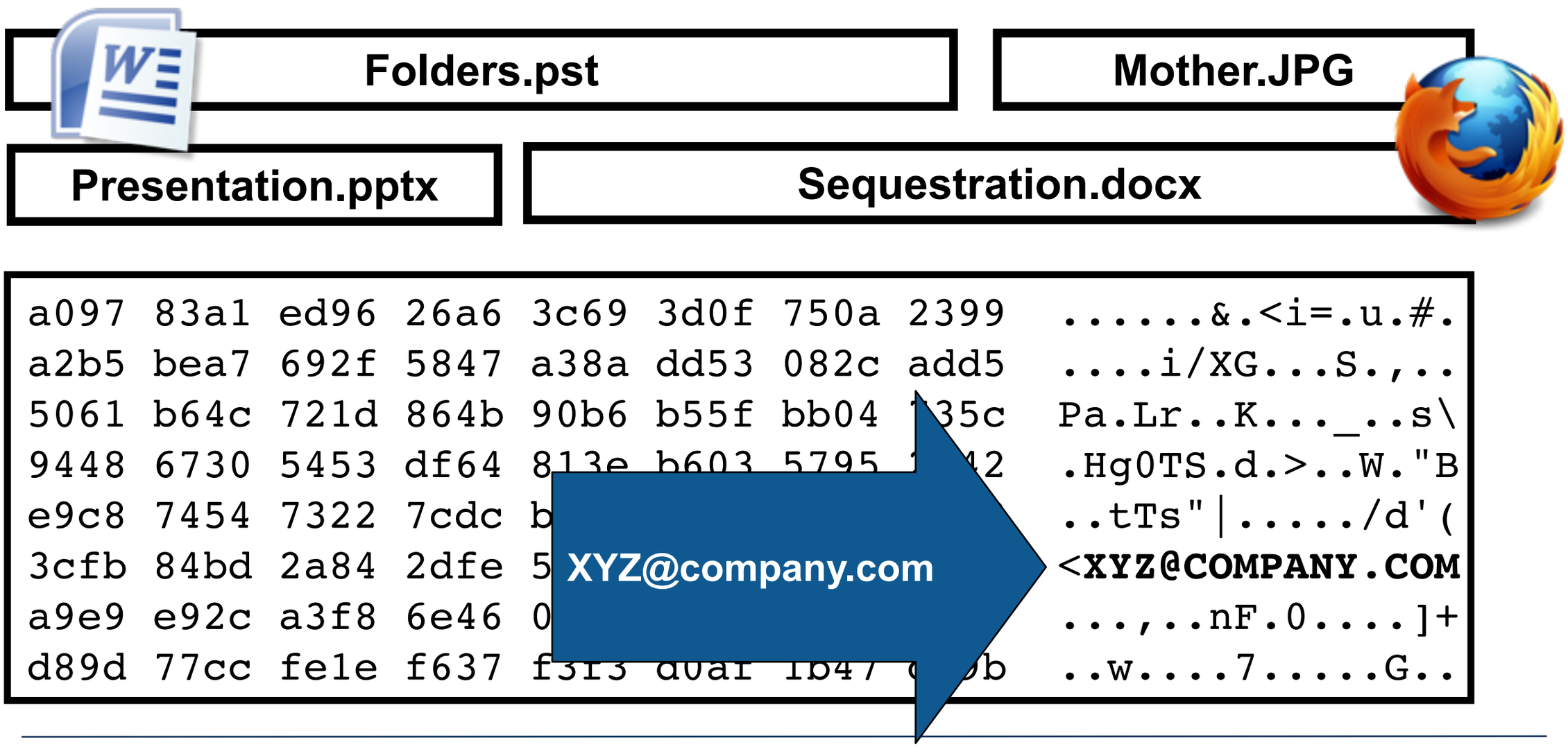
➛ TEXT TEXT TEXT **XYZ@COMPANY.COM** TEXT TEXT TEXT TEXT ➛ **XYZ@COMPANY.COM**

Regular expressions can also extract email addresses from data not in files — “bulk data.”

[bulk data] ➽ RegEx ➽ Email Addresses



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



Every email address is a sequence of bytes.

A simple email address:

```
XYZ@company.com
```

Stored on disk / in memory as 15 bytes:

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

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

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

Normally bytes are displayed in hexadecimal notation:

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

This is UNICODE

Every email address is a sequence of bytes.

A simple email address:

XYZ@company.com

Stored on disk / in memory as 15 bytes:

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

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

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

Normally bytes are displayed in hexadecimal notation:

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

This is UNICODE

Byte sequences can be encoded in many ways.

XYZ@company.com

- Unicode: “XYZ@company.com”

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

- Base 16: “58595a40636f6d70616e792e636f6d0a”

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

- Base 64: “WFlaQGNvbXBhbnkuY29tCg===”

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

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

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

Compression works by eliminating repeated sequences:

Computers use compression to save memory:

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

Compressed with “gzip:”

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

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

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

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

```
e327 962d 6450 3d91 c945 3bed 97a6 a4cd   .'.-dP=..E;.....
1[illegible]b 0800 0000 0000 0203 8b88 8c72 48ce   .............rH.
[illegible]8cc abd4 03d2 0a8e 4ece 287c 1757   .-H.......N.(|.W
[illegible]714 3e00 b455 c1c5 3000 0000 0000 0000   7.>..U..0.......
0a8e 4ece 287c 1757 3714 3e00 a175 10ed   ..N.(|.W7.>..u..
```

Folders.pst

Mother.JPG

Presentation.pptx

Sequestration.docx

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

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

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

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

Folders.pst

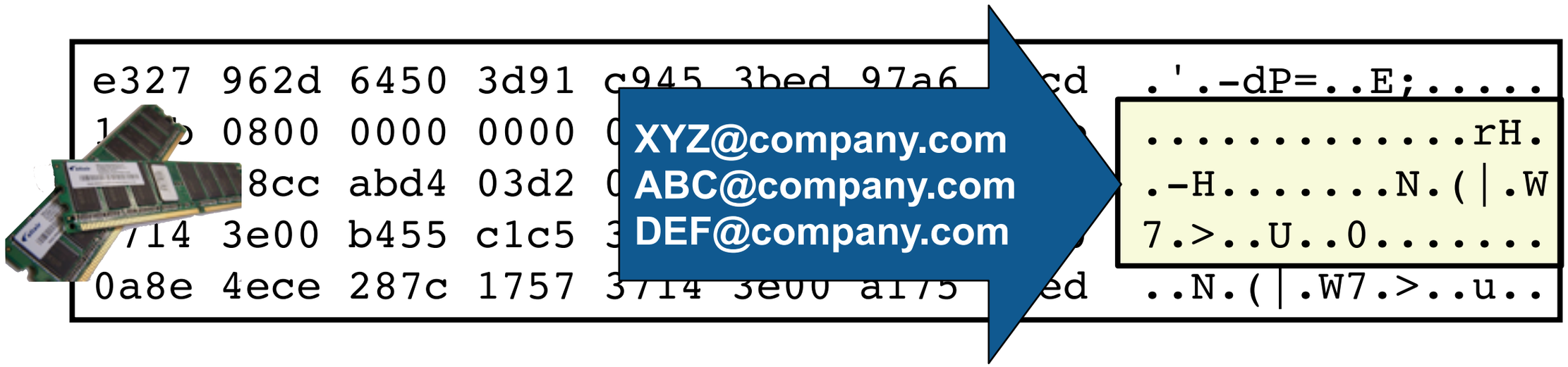
Mother.JPG

Presentation.pptx

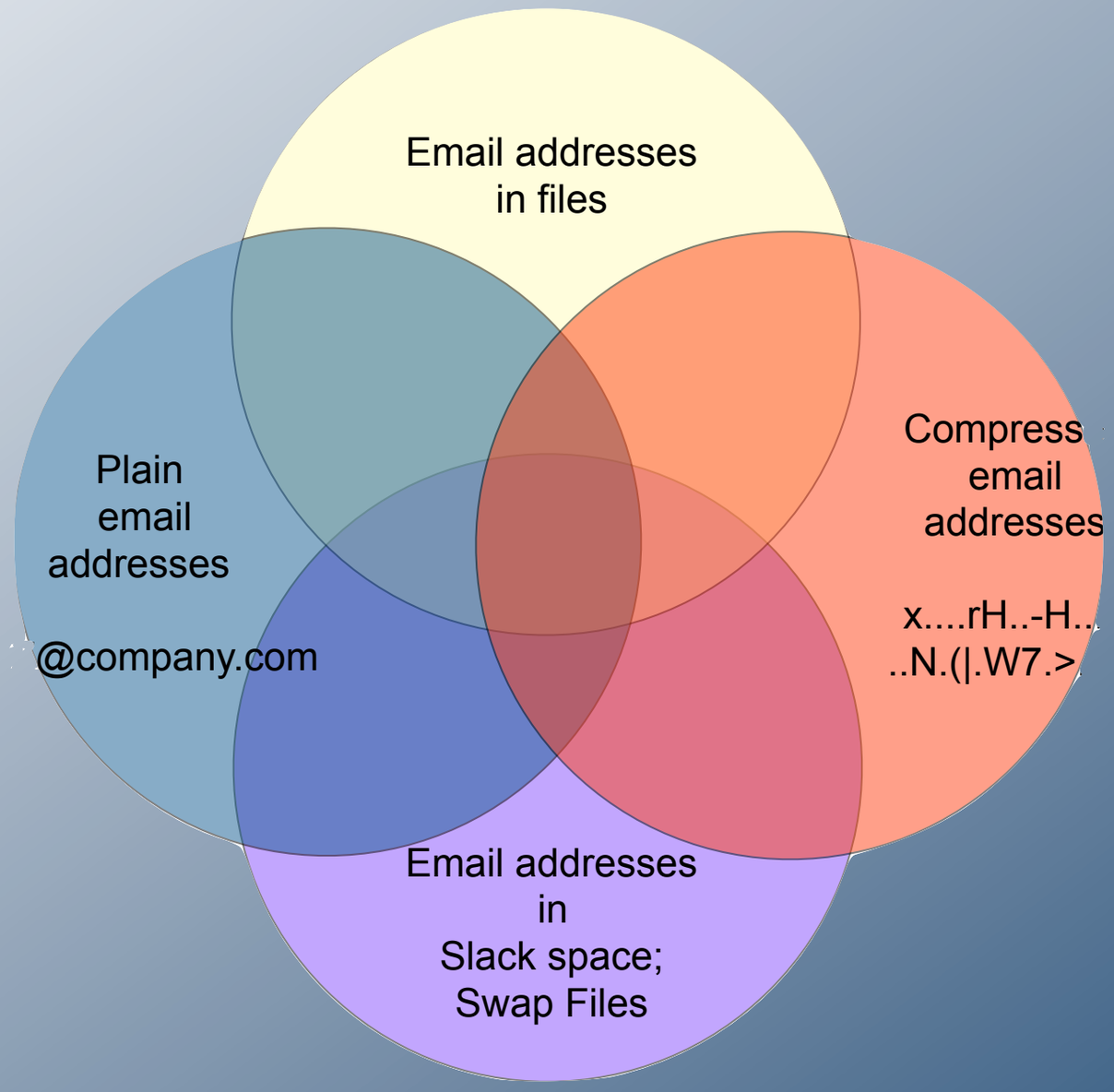
Sequestration.docx

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

Existing commercial digital forensic tools ignore compressed email addresses in bulk data.



This is a serious problem.



How many encoded features are conventional tools missing?

Email addresses can be in files

Files

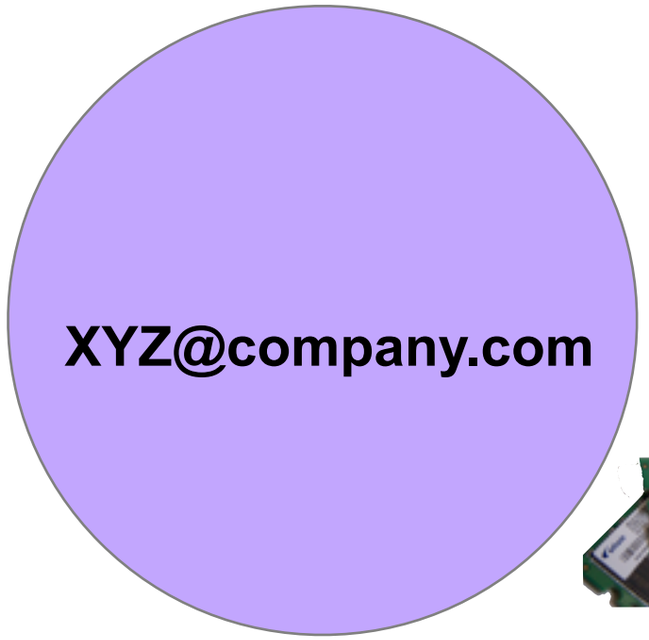
- Documents
- Address book
- Email messages

ABC@company.com
DEF@company.com

Browser Cache:

- Web mail
- Facebook Data

Email addresses can be in non-file disk sectors



Swap Files
Hibernation Files
File fragments

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

ABC@company.com
DEF@company.com

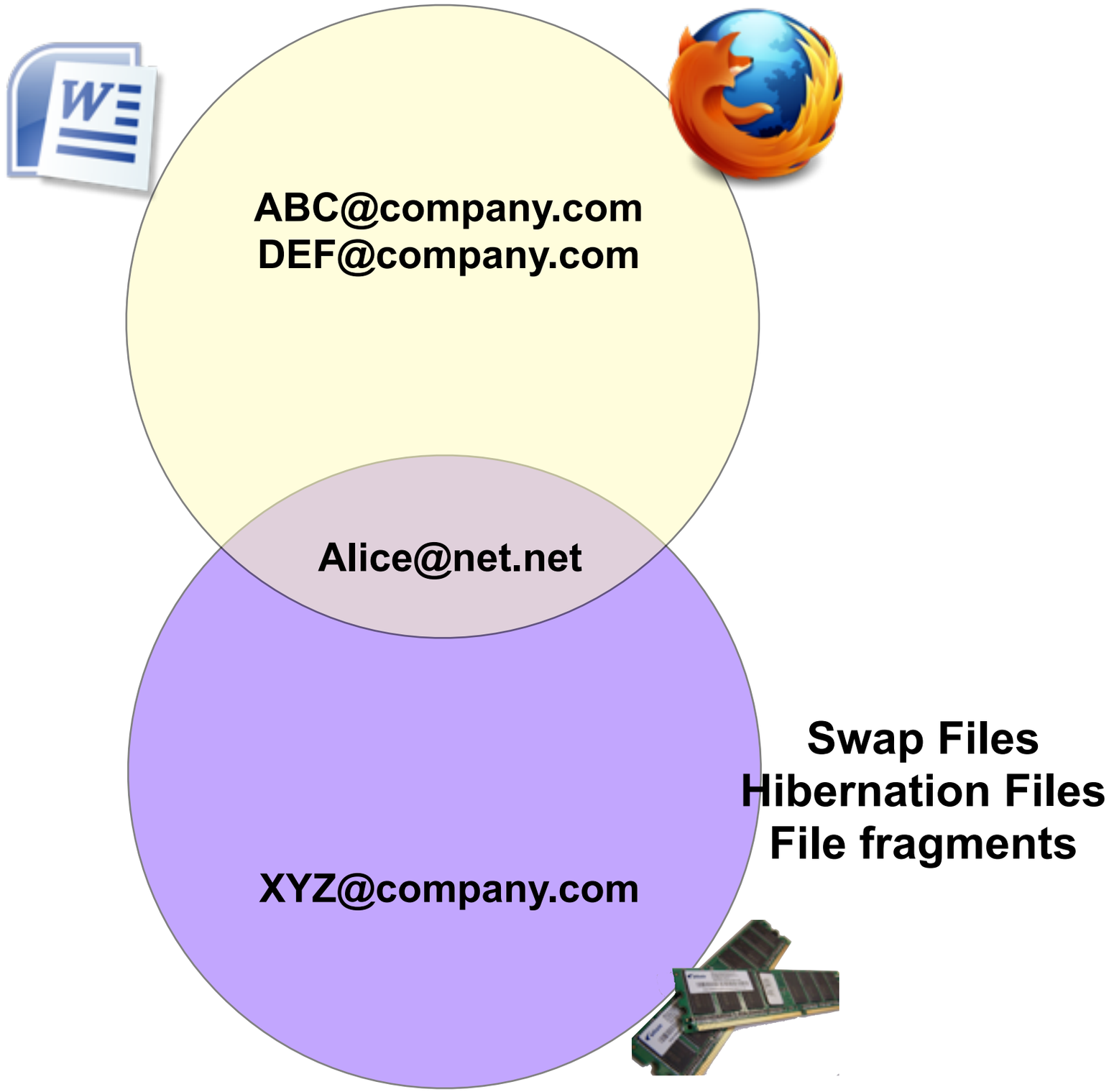
Alice@net.net

Alice@net.net

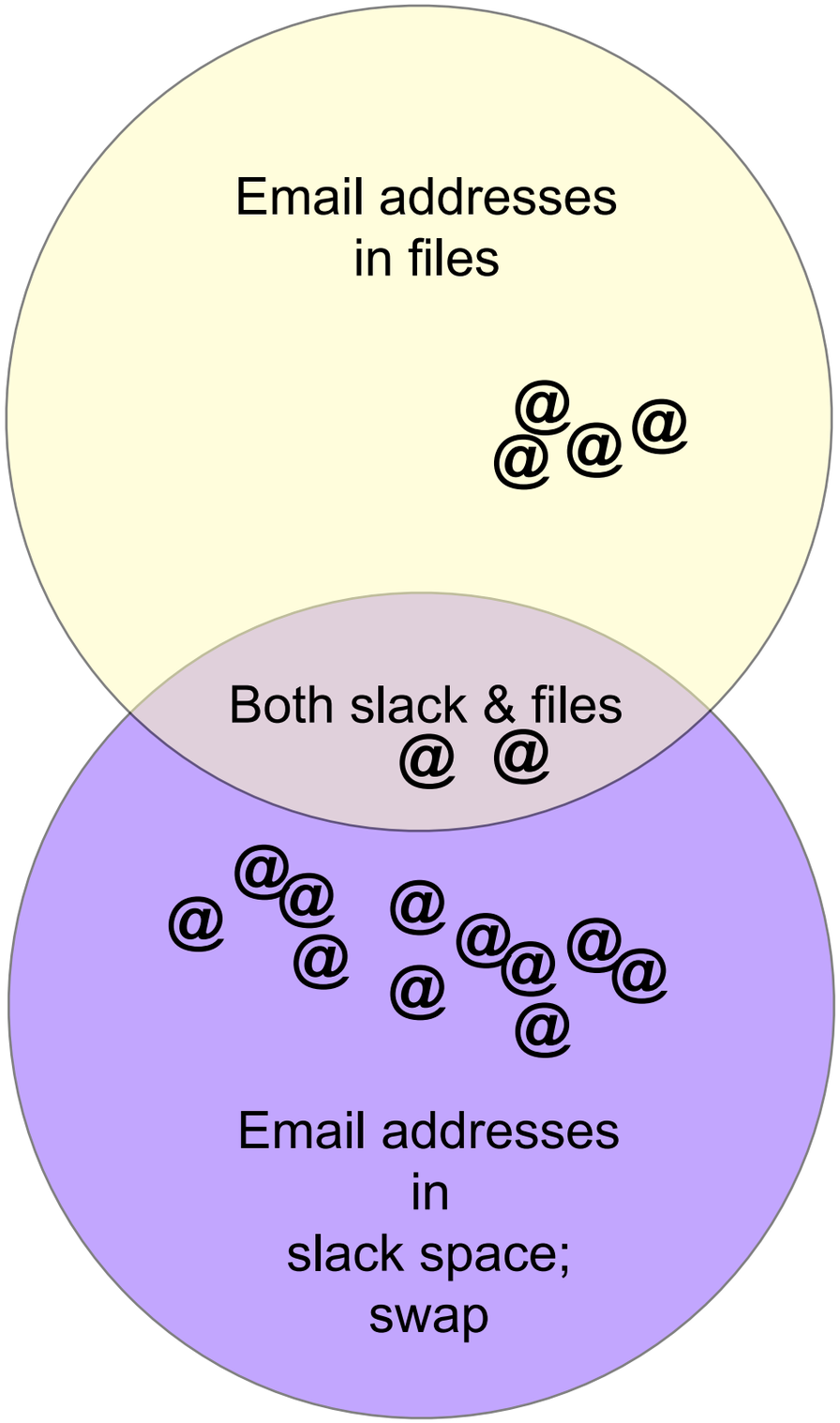
XYZ@company.com

Swap Files
Hibernation Files
File fragments

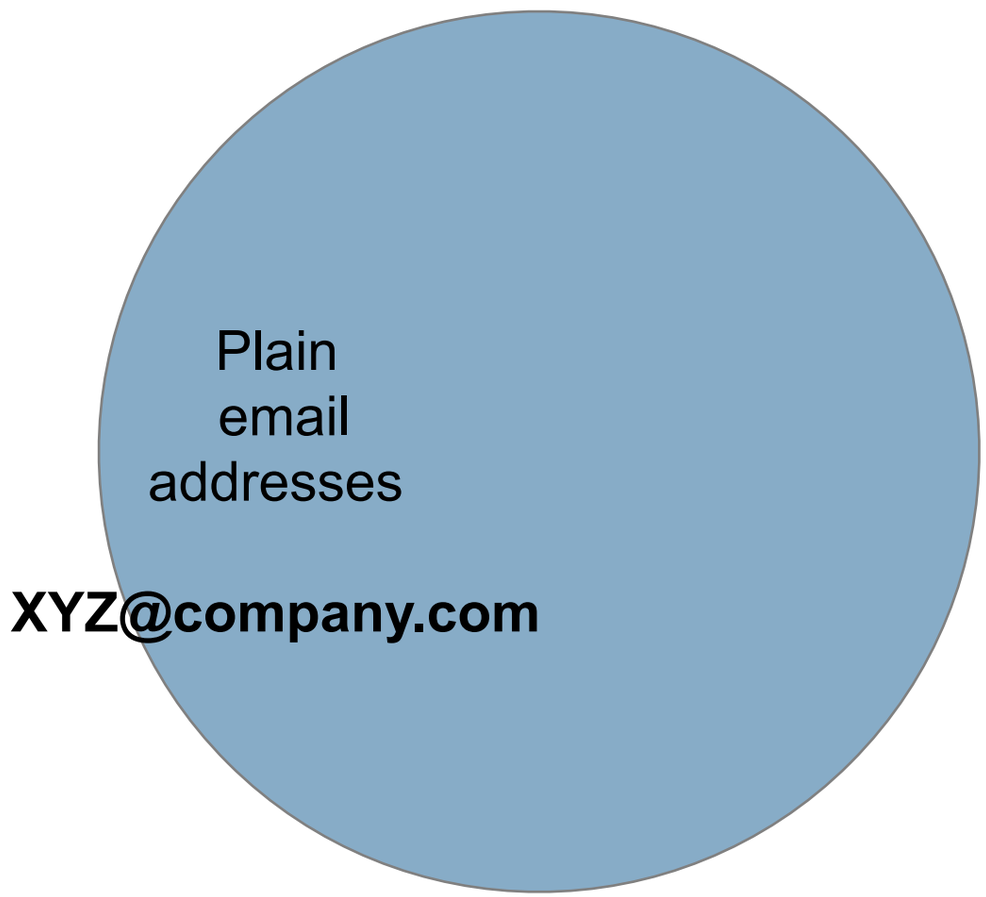
This Venn Diagram represents email addresses on media.



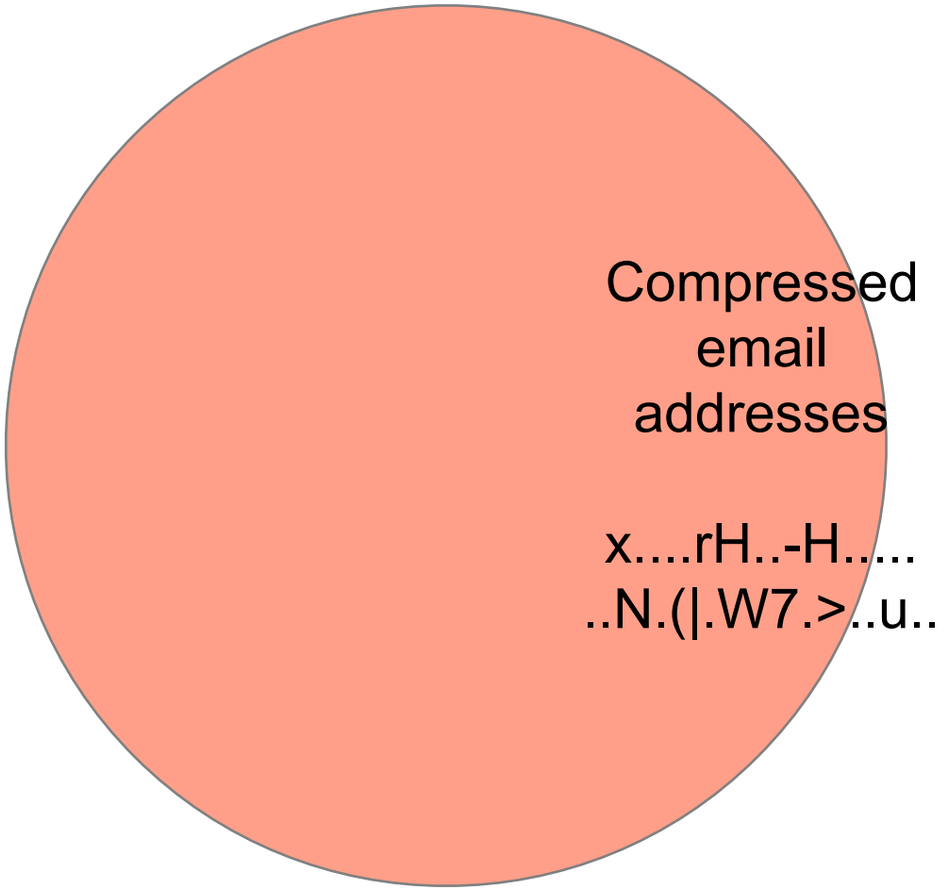
The number of email addresses in each region depends on the media.



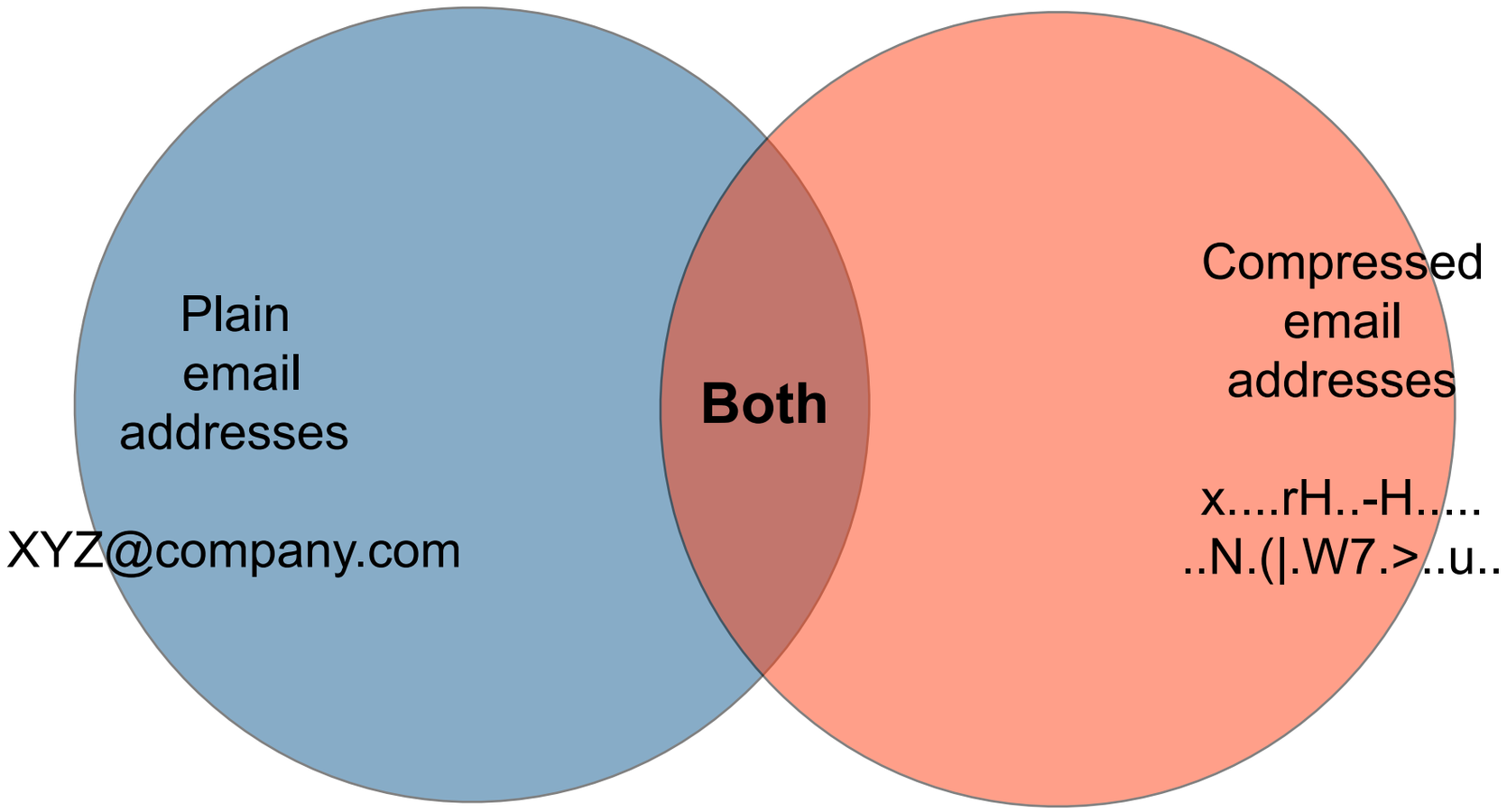
Email addresses can be plain text. “XYZ@company.com”



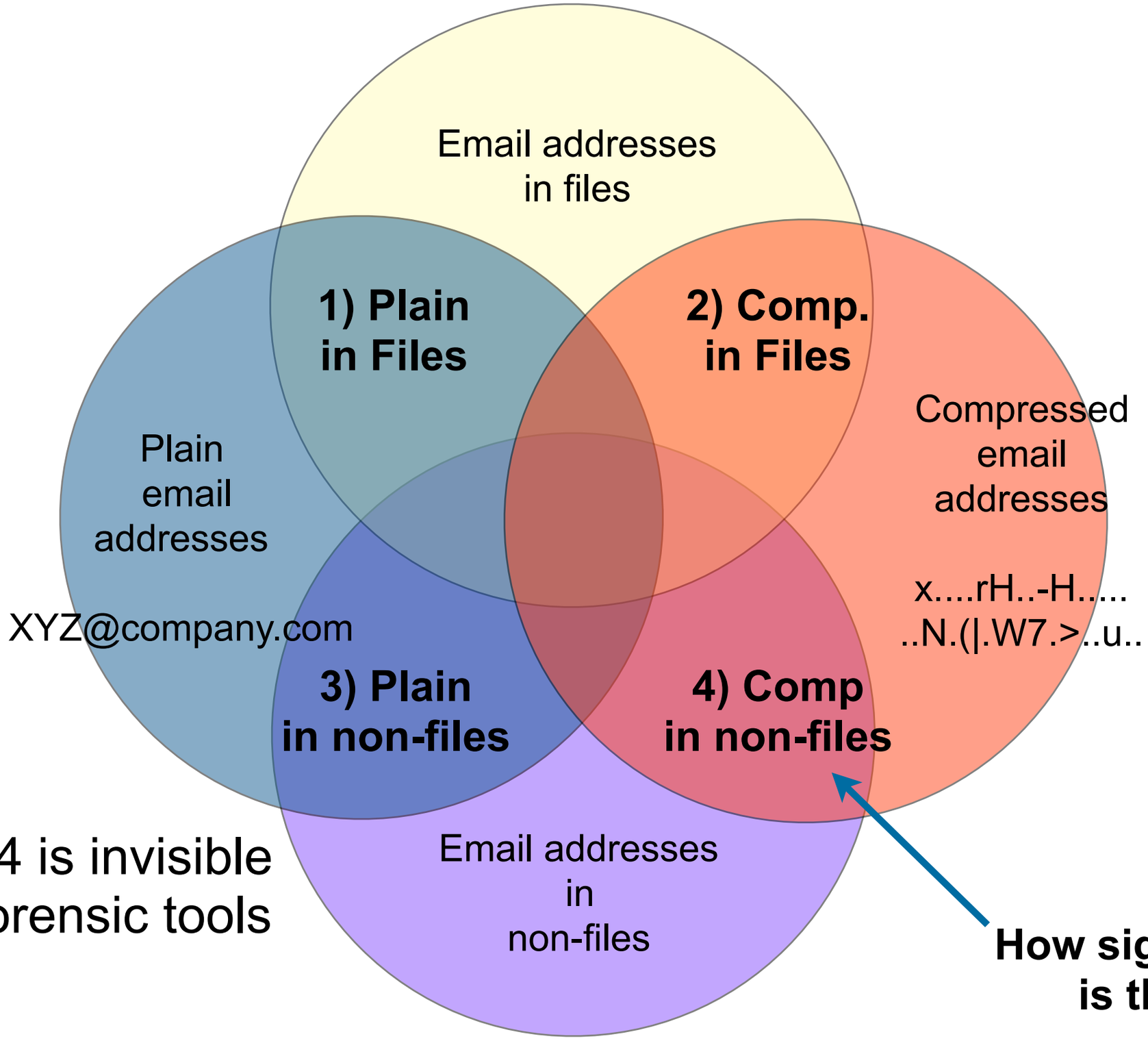
Email addresses can be compressed or encoded.
“x....rH..-H.....N.(|.W7.>..u..”



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



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

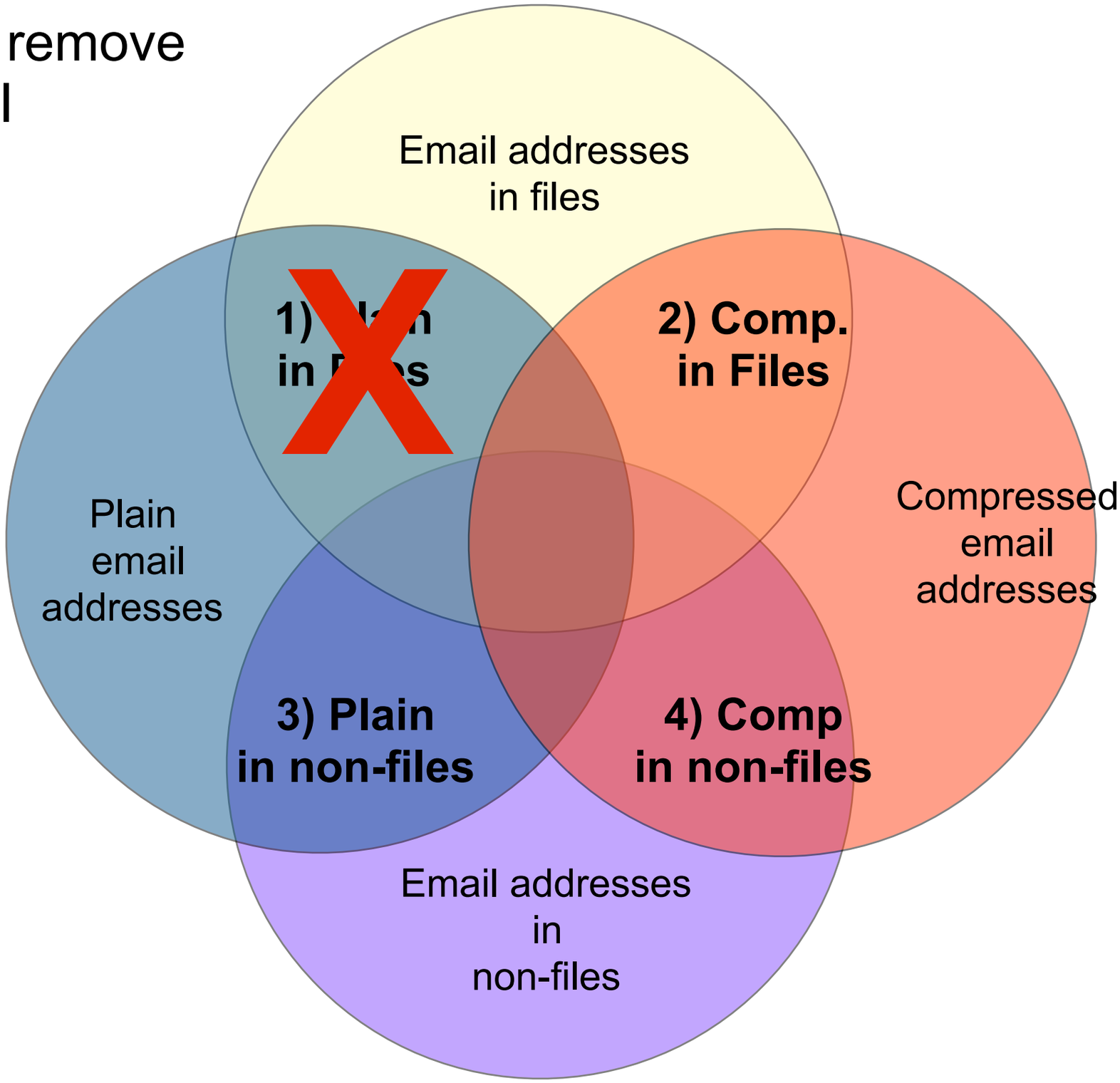


Condition #4 is invisible to today's forensic tools

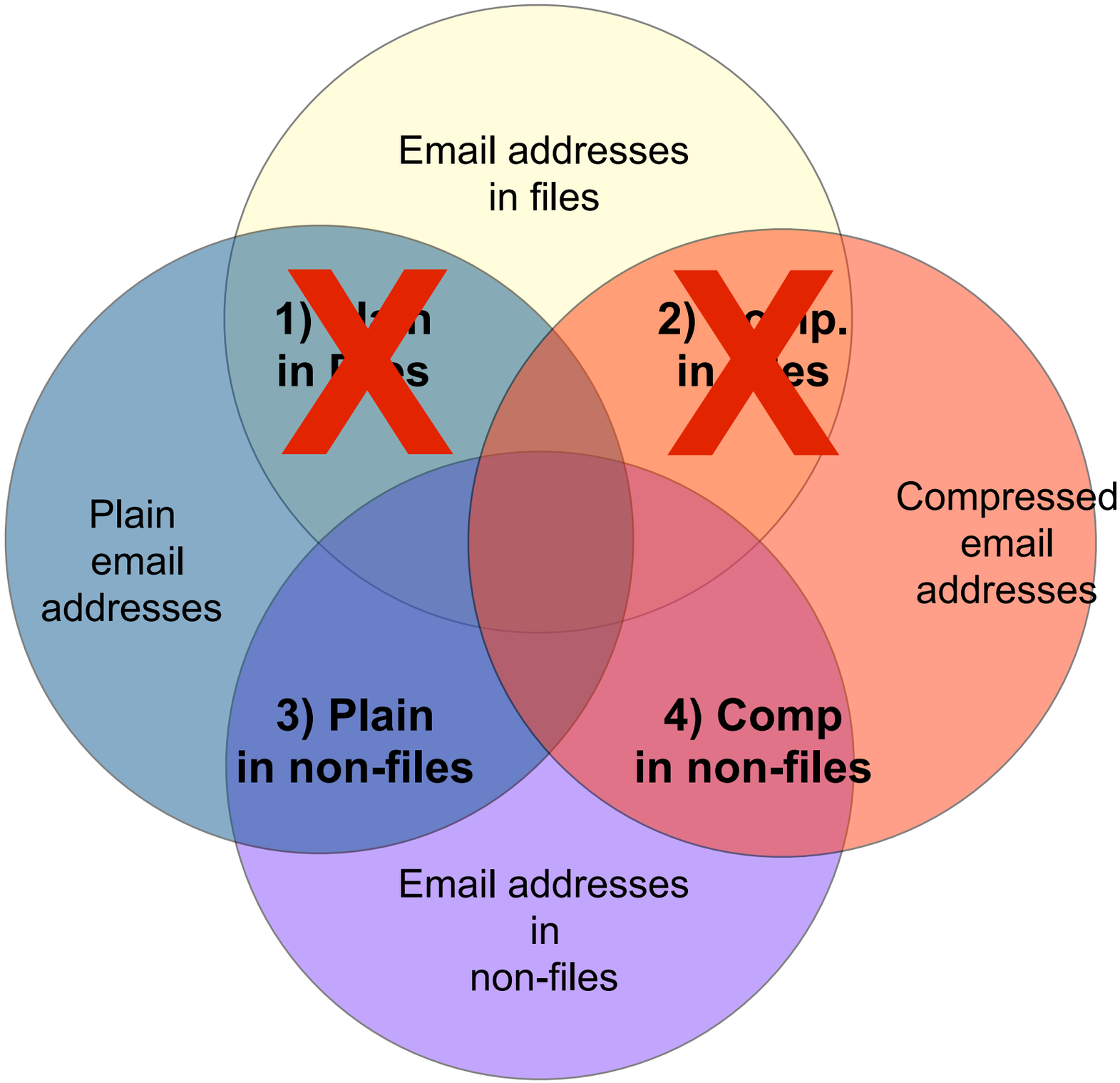
How significant is this?

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

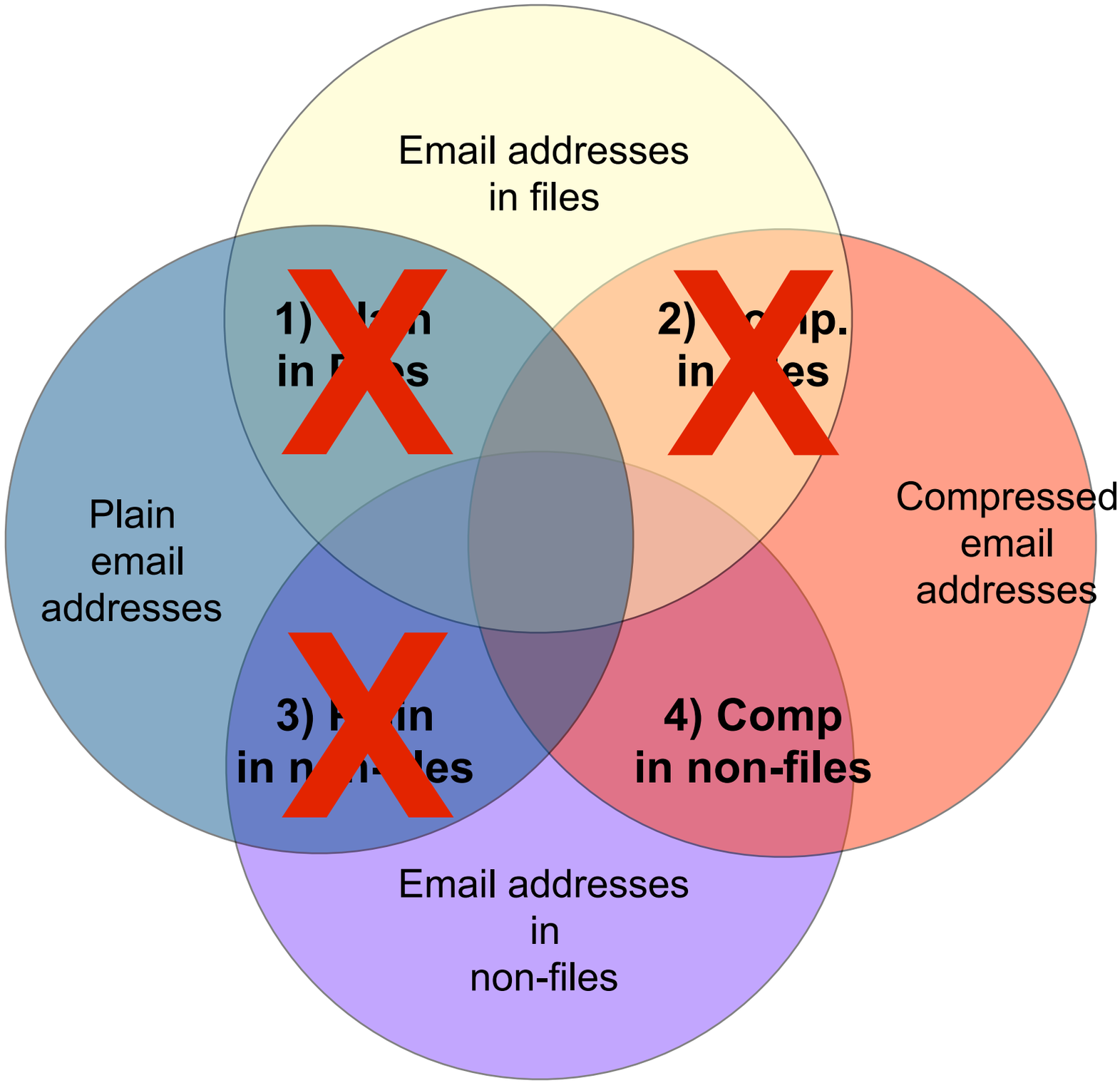
First, find and remove the plain email addresses in files.



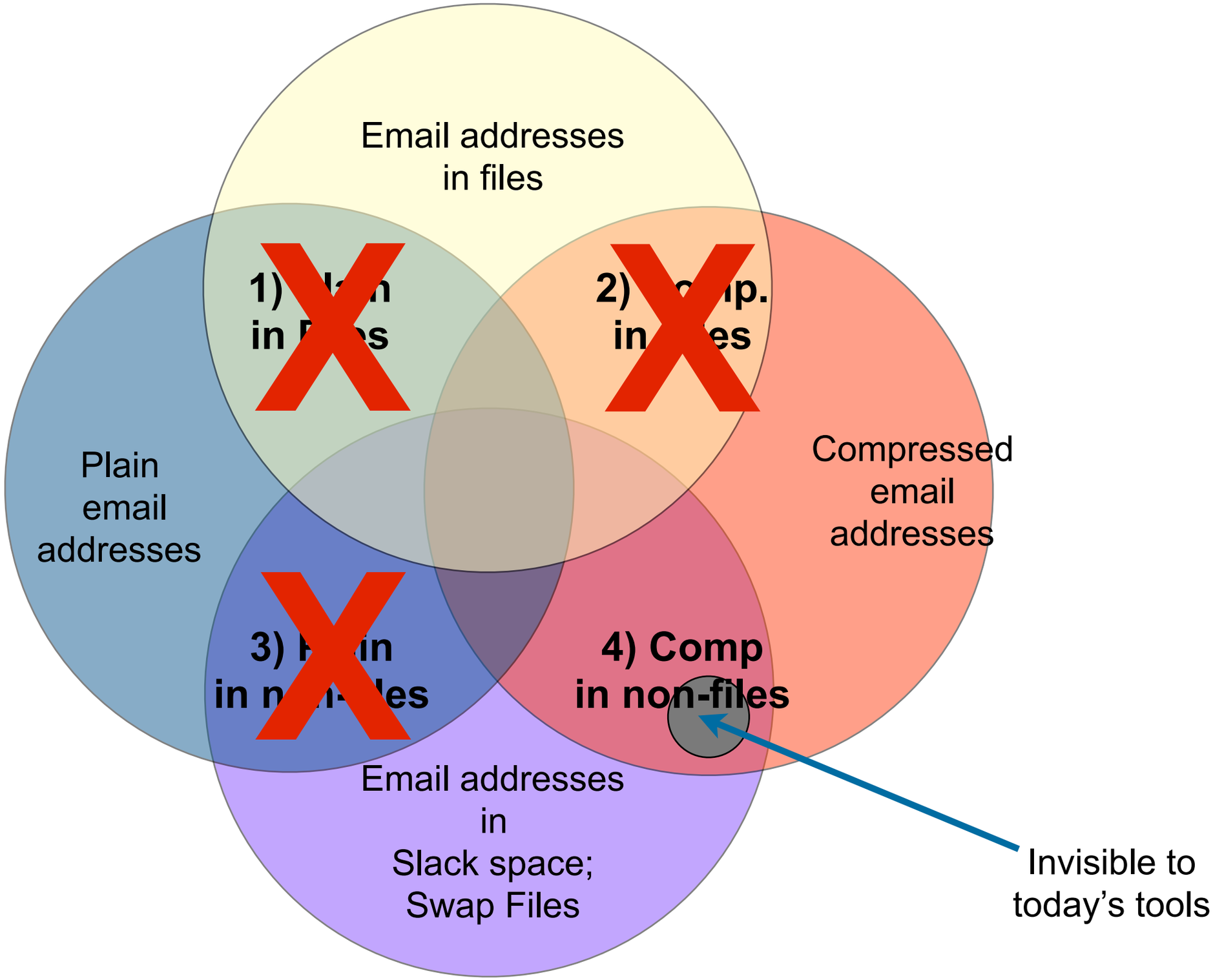
...Remove the addresses compressed and in files....



...Remove email addresses that are not compressed.

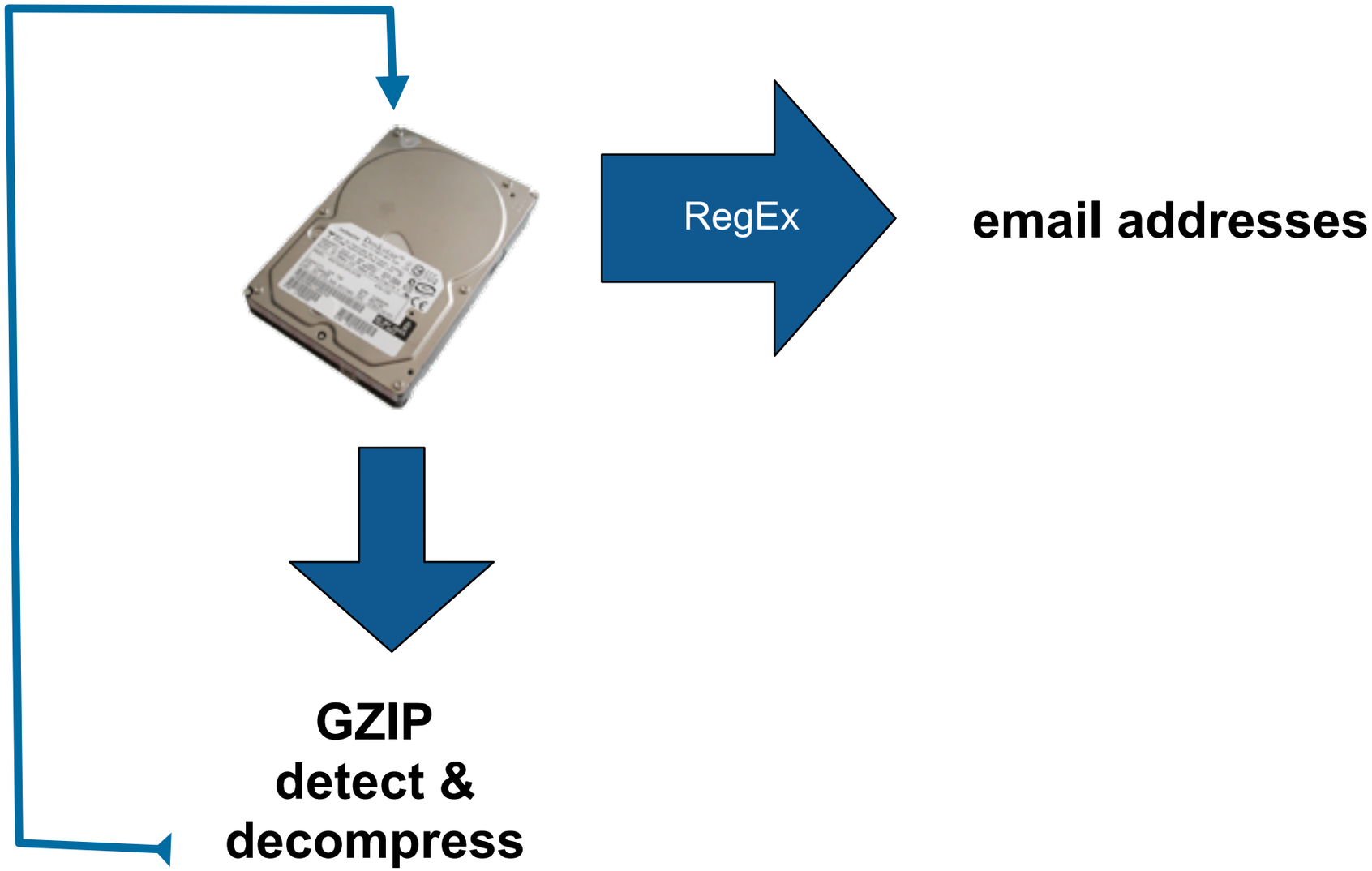


...those that remain are the “invisible” email addresses.



bulk_extractor is an experimental email extraction tool.

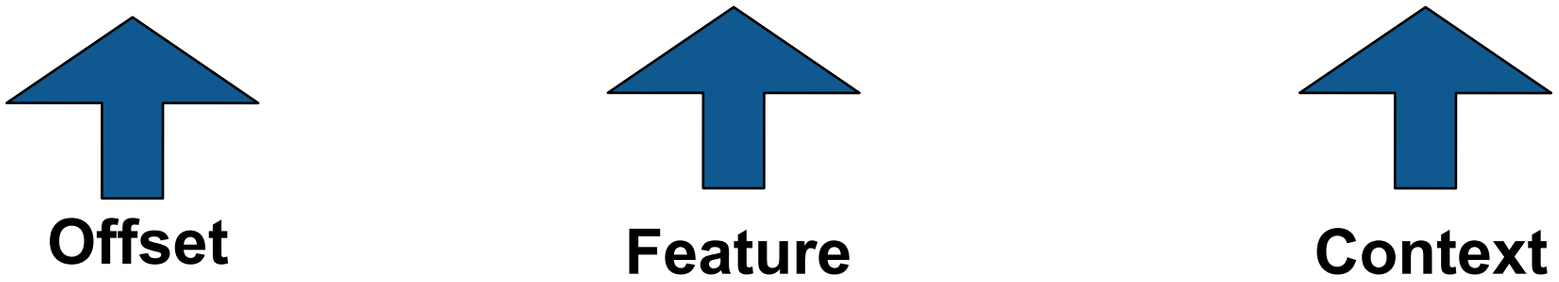
"Digital media triage with bulk data analysis and bulk_extractor,"
Simson L. Garfinkel, *Computers and Security 32 (2013) 56-72*



bulk_extractor can find both plain and compressed text.

“Feature files” contain the extracted email addresses.

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



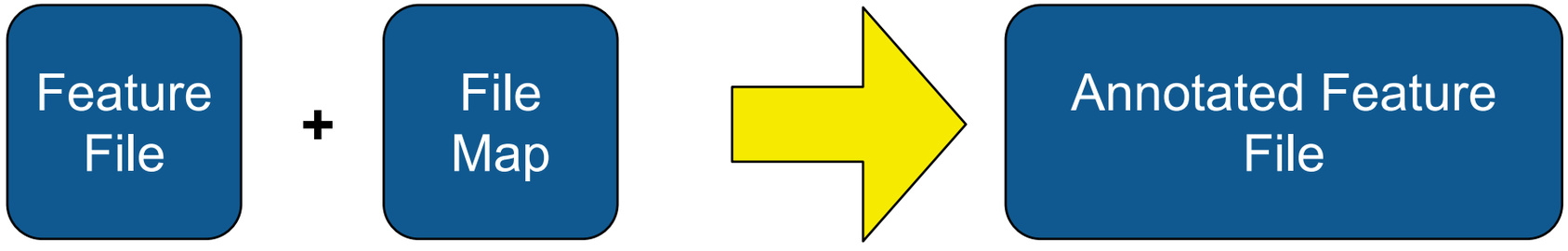
Plain text features have numeric offsets:

392175418

Compressed features will indicate the algorithm:

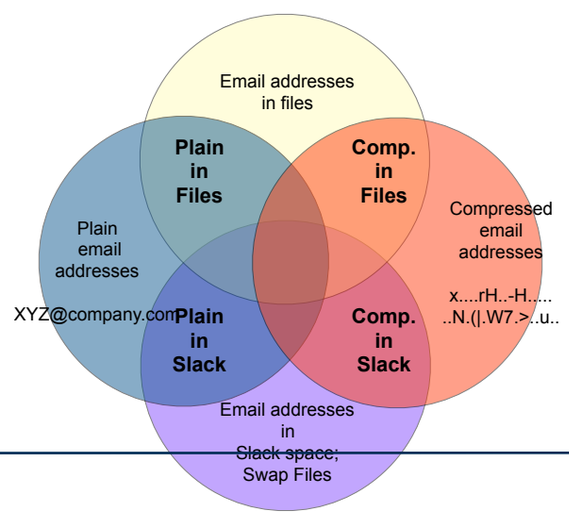
3772517888-GZIP-28322

Post-processing with identify_files.py reveals file names




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

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



bulk_extractor 1.4 recognizes a wide variety of features and encoding types:

Feature types:

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

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

Encoding Types:

- ZIP; GZIP; RAR; Windows Hibernation
- BASE16, BASE64

Some drives have a lot of compressed data

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

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

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

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

Emails seen	count	1) Plain in Files	2) Comp. in Files	3) Plain in non-files	4) Comp in non-files
Cleartext		358	--	5341	--
All Comp		--	9	--	135
GZIP	50		14		36
HIBER	39		7		32
HIBER-GZIP	23				23
PDF	88		1		87
ZIP	28		7		21
ZIP-PDF	18				18

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

Many of these email addresses are significant

Example email addresses (sanitized)

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

Questions:

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

We do science with “real data.”

The Real Data Corpus (60TB)

- 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.

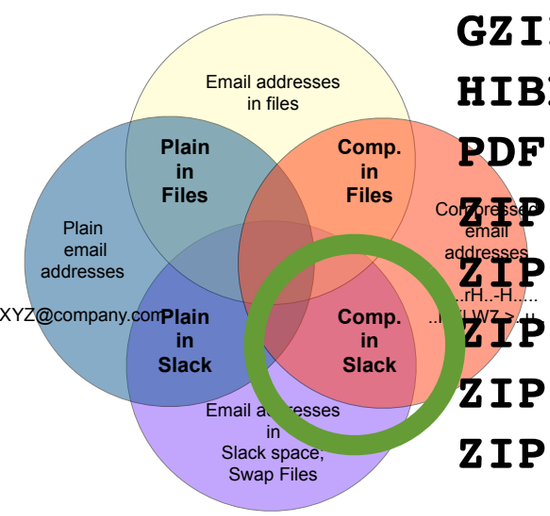
The problems we encounter obtaining, curating and exploiting this data mirror those of national organizations

—*Garfinkel, Farrell, Roussev and Dinolt, Bringing Science to Digital Forensics with Standardized Forensic Corpora, DFRWS 2009. BEST PAPER AWARD.*

—*http://digitalcorpora.org/*

We analysis 1,646 disk images that had intact file systems. Many email addresses existed only encoded, in non-files.

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



Some drives had more than 10,000 compressed email addrs.

Remember — compressed email addresses in non-files are ignored by today's forensic tools.

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

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

Folders.pst

Mother.JPG

Presentation.pptx

Sequestration.docx

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

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

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

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

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

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

Documents:

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

Archives:

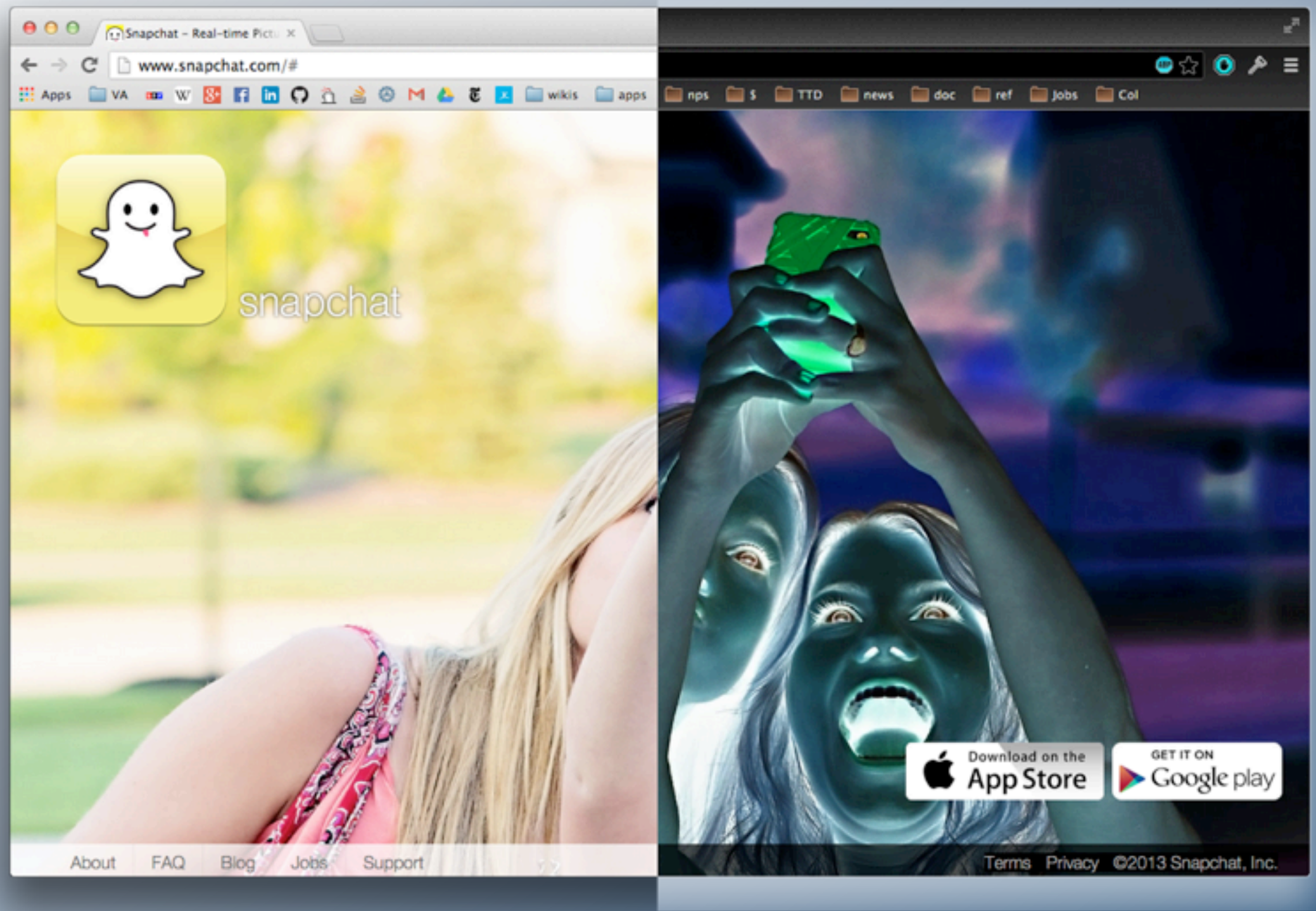
- ZIP files; GZIP (GZ) files

System Resources:

- Hibernation files & file fragments

If forensic examiners miss an email address:

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



XOR

Summer 2013 Research Project

Each summer for the past three years, NPS NCR has hosted interns to research digital forensics.

Previous projects:

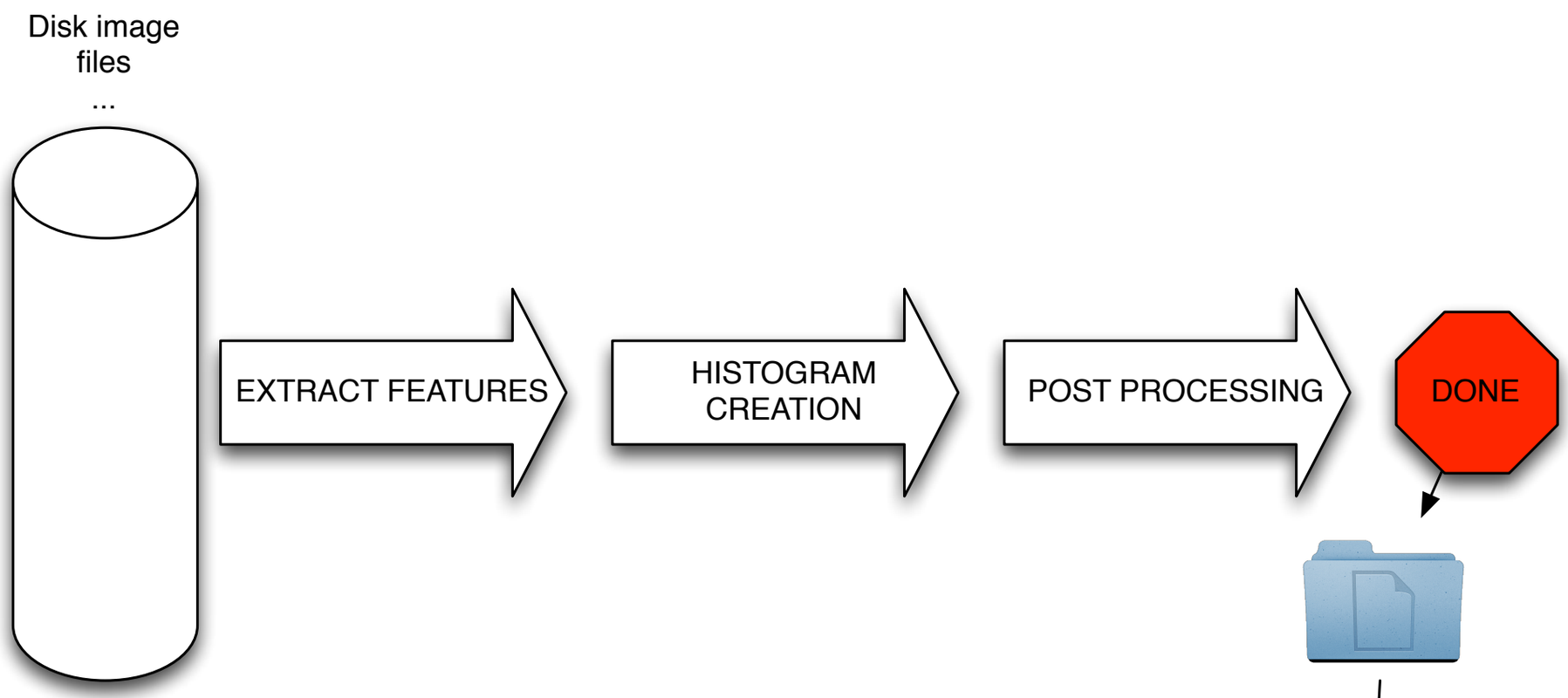
- Summer 2011 — bulk_extractor enhancements
- Summer 2012 — National Gallery DC Scenario

Summer 2013 — XOR usage in the Real Data Corpus

- CDT Aubin Heffernan, USMA
- CDT Scott Horras, USMA
- CDT Kyle Gorak, USMA
- Ms. Carolina Zarate, Poolesville High School

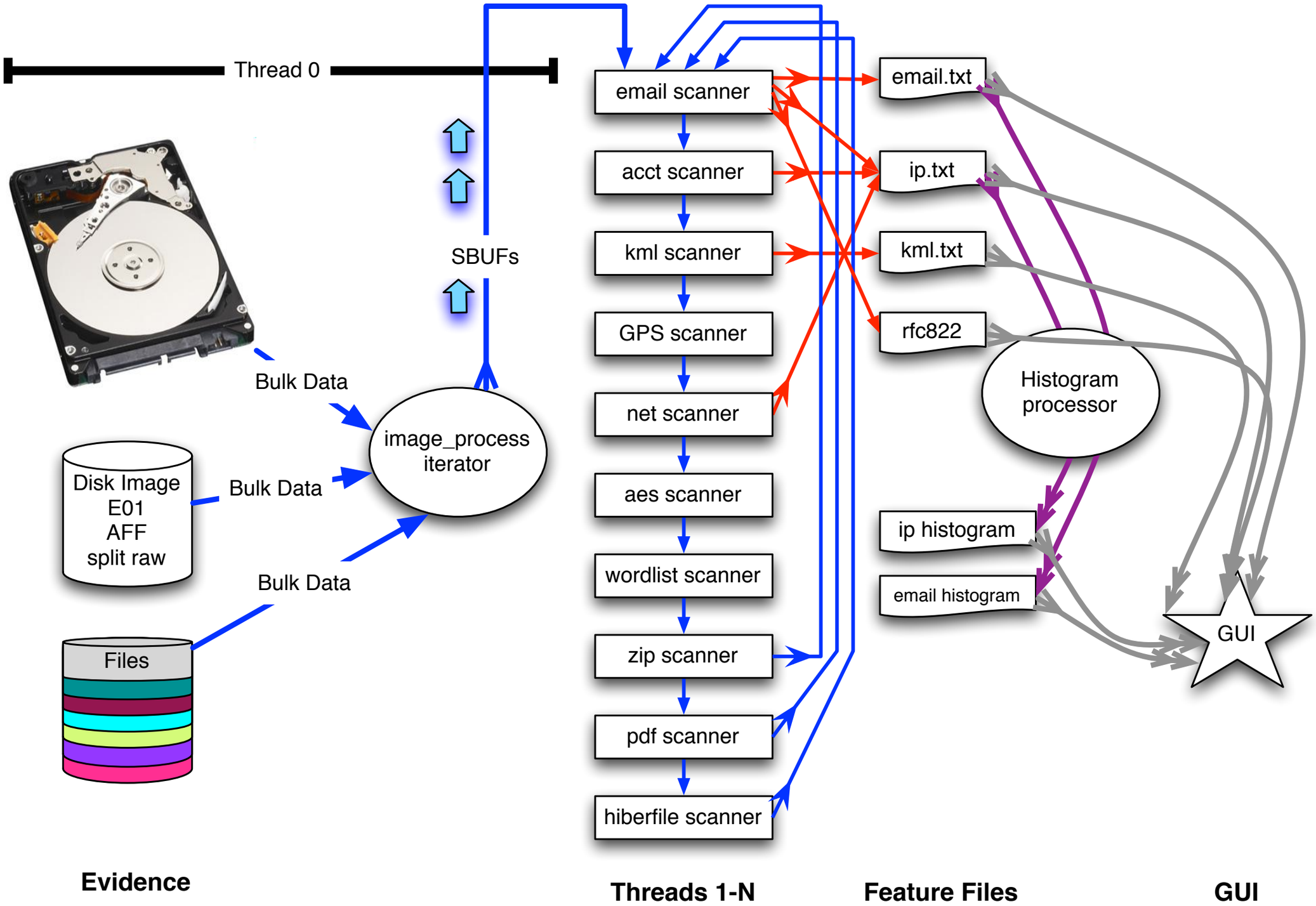
The students analyzed bulk_extractor output.

Recall that bulk_extractor processes data and outputs feature files:

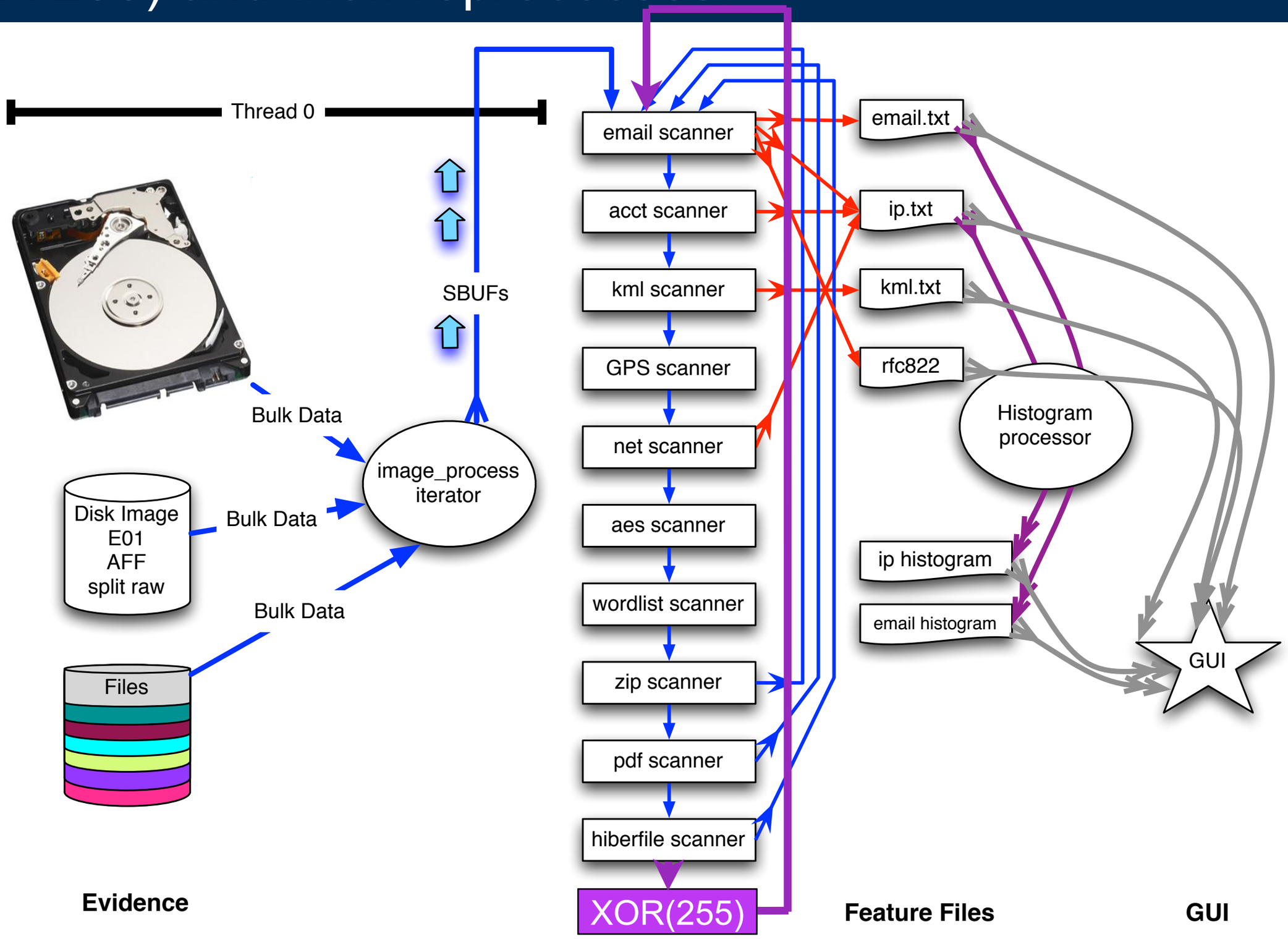


report.xml — log file
telephone.txt — list of phone numbers with context
telephone_histogram.txt — histogram of phone numbers
vcard/ — directory of VCARDs
...

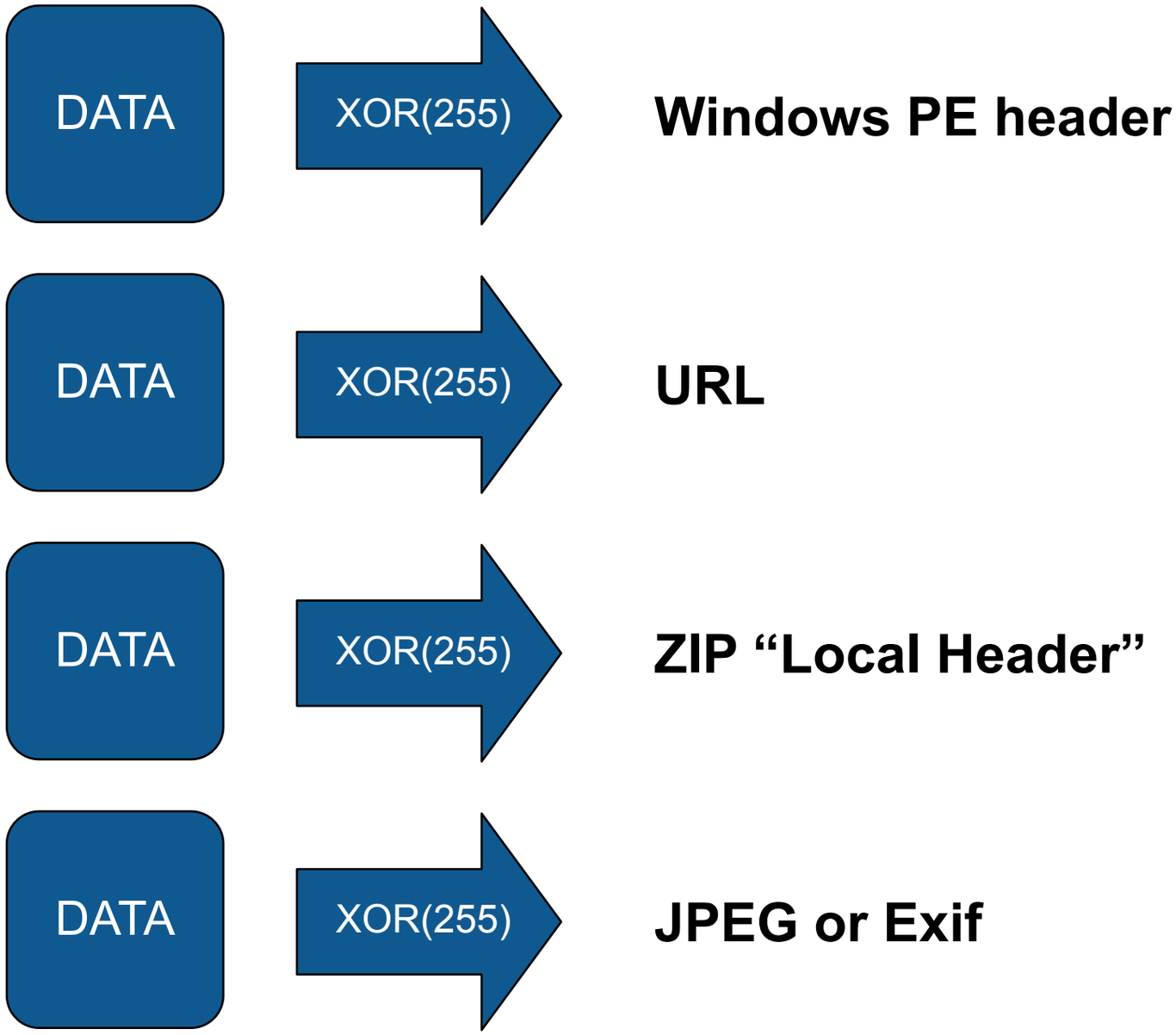
bulk_extractor's internal design:



We created another “scanner” that inverts the SBuf (XOR 255) and then reprocesses.

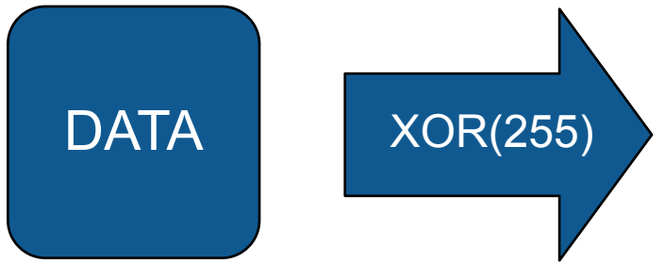


We searched for valid data that had been XOR'ed.

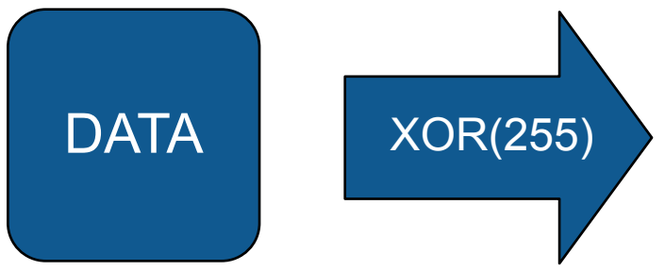


These kinds of data can be recognized with high reliability.

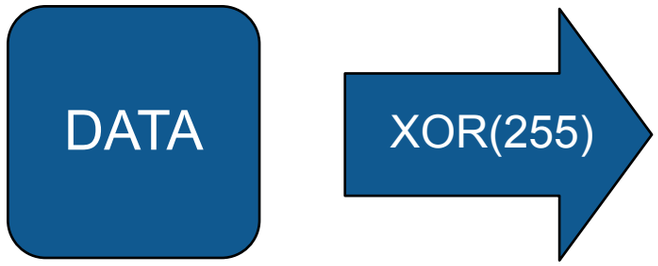
We found substantial XOR'ed data.



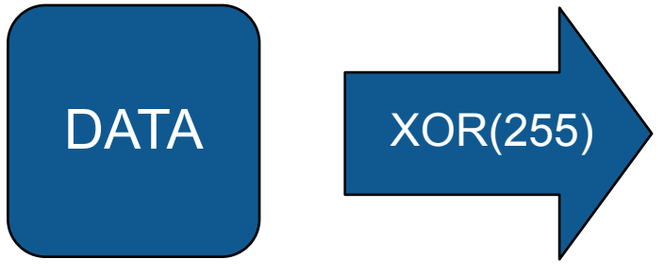
Windows PE header ***Malware***



URL ***Malware URLs***



ZIP "Local Header" ***Malware data exfil***



JPEG or Exif ***Game components***

These kinds of data can be recognized with high reliability.

Both legitimate and illicit use of XOR(255) to hide data.

We examined anti-virus systems and found:

- Malware used XOR(255) to hide download URLs
- AV XOR'ed Malware that was put into QUARANTINE
- VirusTotal did not recognize uploaded malware that had been XORed.

XOR(255) in commercial software:

- Real Audio — to obscure Dr. Yuriy Reznik's email address.
- Nero 7 — to hide a watermark (http://www.nero.com)

XOR(255) in Exfil'ed data:

- Fragments of a ZIP file that had been XOR'ed.
- Contents were Excel spreadsheets with names & salary data.

XOR(255) is throughout our corpus.

Year	# URL	# WinPE	# ZIP	#Exif
1980	4	7	0	0
1981	6	0	0	0
1985	15	0	0	0
1990	0	20	0	0
1996	2	11	0	0
1997	185	15	0	0
1998	443	126	3	0
1999	261	526	44	0
2000	252	526	12	0
2001	593	238	1	0
2002	734	234	1	0
2003	224	87	0	0
2004	1,359	427	34	0
2005	2,640	184	0	0
2006	1,934	3,840	6	0
2007	315	16,782	0	0
2008	1,376	1,973	0	0
2009	1,722	489	0	0
2010	802	468	0	0
2011	14,594	8	74	0
2013	11	1	0	0
2014	49	1	0	0
2016	10	1	0	0
2018	818	0	0	0
2019	3	0	0	0
2023	2	0	0	0
2027	346	0	0	0
2029	4	1	0	0
2030	4	2	0	0
2033	218	0	0	0
2037	14	0	0	0
2080	20	14	0	0
2081	10	2	0	0
no file	252,742	11,550	4,594	130
Total	281,712	37,533	4,769	130

Table 1: Validated XOR features by year for the analyzed drives, where the "year" is corresponds to the modification time of the file within which each XOR-encoded feature was found. "no file" indicates that the XOR-encoded features could not be located to a specific file. Timestamps prior to 1996 and after 2011 are likely the result of an improperly set system clock or on-disk corruption and are reported here for completeness.

XOR(255) was found in drives from (all) 21 countries .

country	total drives	drives with XOR WinPE	drives with XOR URL	drives with XOR ZIP	drives with XOR exif
BANGLADESH	57	15	5	0	0
BOSNIA AND HERZEGOVINA	7	0	0	0	0
CANADA	48	8	1	0	0
CHINA	807	25	1	0	0
EGYPT	7	2	2	0	0
GERMANY	37	22	6	1	0
GHANA	19	8	1	0	0
GREECE	10	2	0	0	0
INDIA	603	185	77	13	4
ISRAEL	260	84	39	9	0
MEXICO	173	73	16	3	1
MONACO	11	6	2	0	1
NEW ZEALAND	1	0	0	0	0
PAKISTAN	81	31	2	0	0
PALESTINE, STATE OF	140	39	8	3	0
SINGAPORE	34	4	1	0	0
SWITZERLAND	2	0	0	0	0
THAILAND	17	9	1	2	1
TURKEY	10	6	2	0	0
UNITED ARAB EMIRATES	87	62	7	19	0
Total	2,411	581	171	50	7

Table 2: Incidence of drives with Validated XOR features, by country

Unfortunately, our current XOR implementation significantly increases processing time.

test image	Size	without XOR	with XOR	Δ
nps-2009-domexusers	40GB	522 sec	799 sec	+53%
nps-2011-2tb	2TB	34,140 sec	58,147 sec	+70%

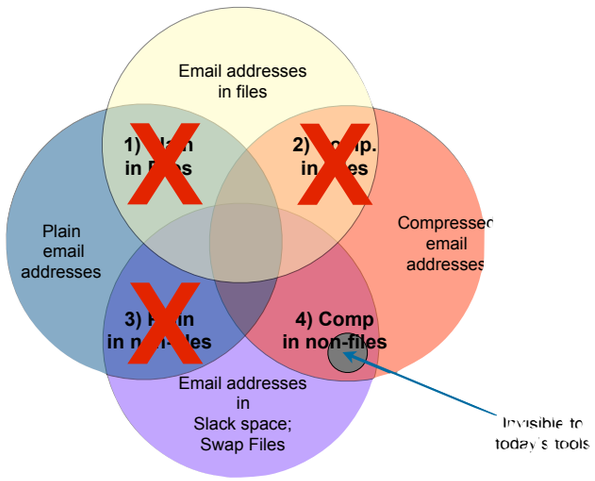
Table 4: Observed processing times for *bulk_extractor* with and without XOR scanner.

Solution 1 — Only use when “necessary.”

Solution 2 — Examine data *before* XORing

In conclusion:
Encoded data in non-file space is systematically ignored.

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



We demonstrated the extent of the problem with:

- bulk_extractor, a high-performance stream-based feature extractor
  - —*https://github.com/simsong/bulk_extractor* *(dev tree)*
  - —*http://digitalcorpora.org/downloads/bulk_extractor* *(downloads)*
  - —*http://www.sciencedirect.com/science/article/pii/S0167404812001472 (paper)*
  - —*http://simson.net/clips/academic/2013.COSE.bulk_extractor.pdf*
- Real Data Corpus:
  - —*http://digitalcorpora.org/*

Contact Information:
Simson L. Garfinkel
slgarfin@nps.edu
http://simson.net/