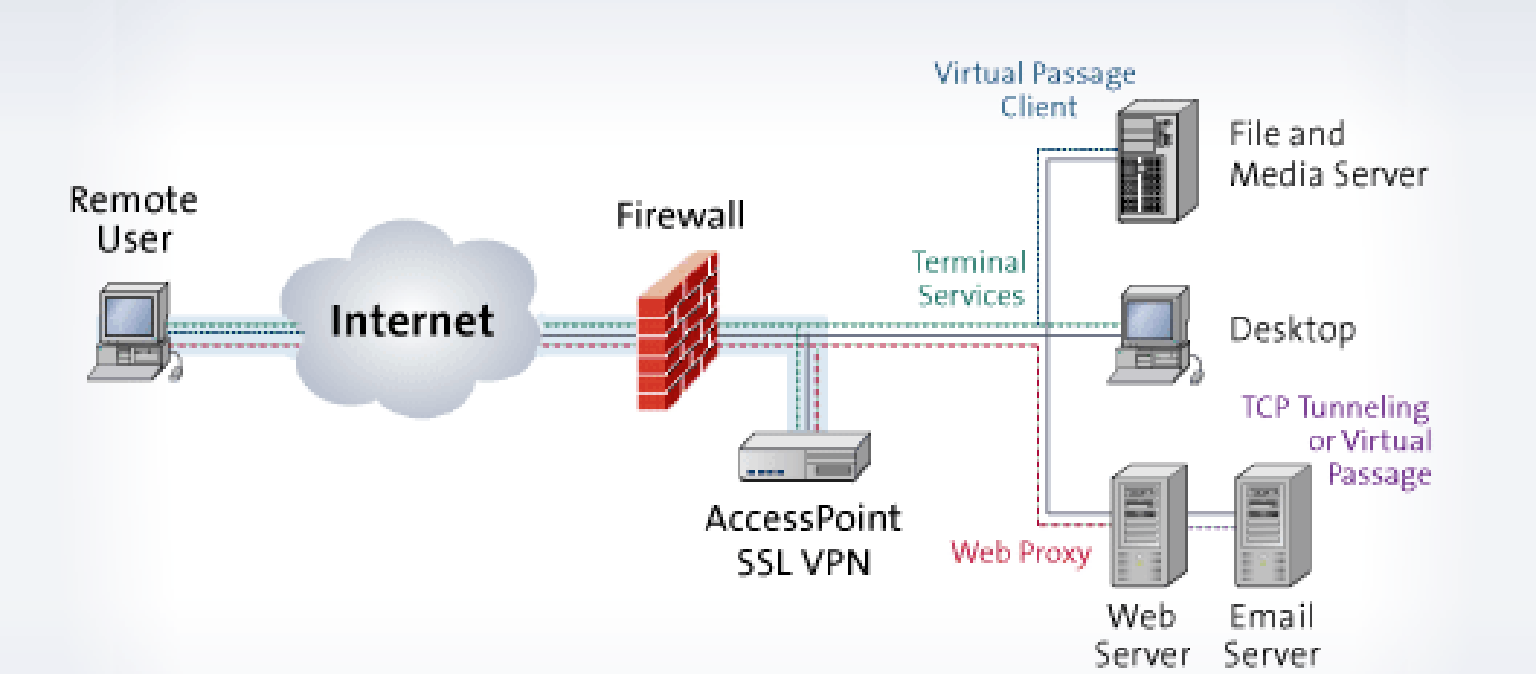
# Inferring the Source of Encrypted HTTP Connections

Marc Liberatore

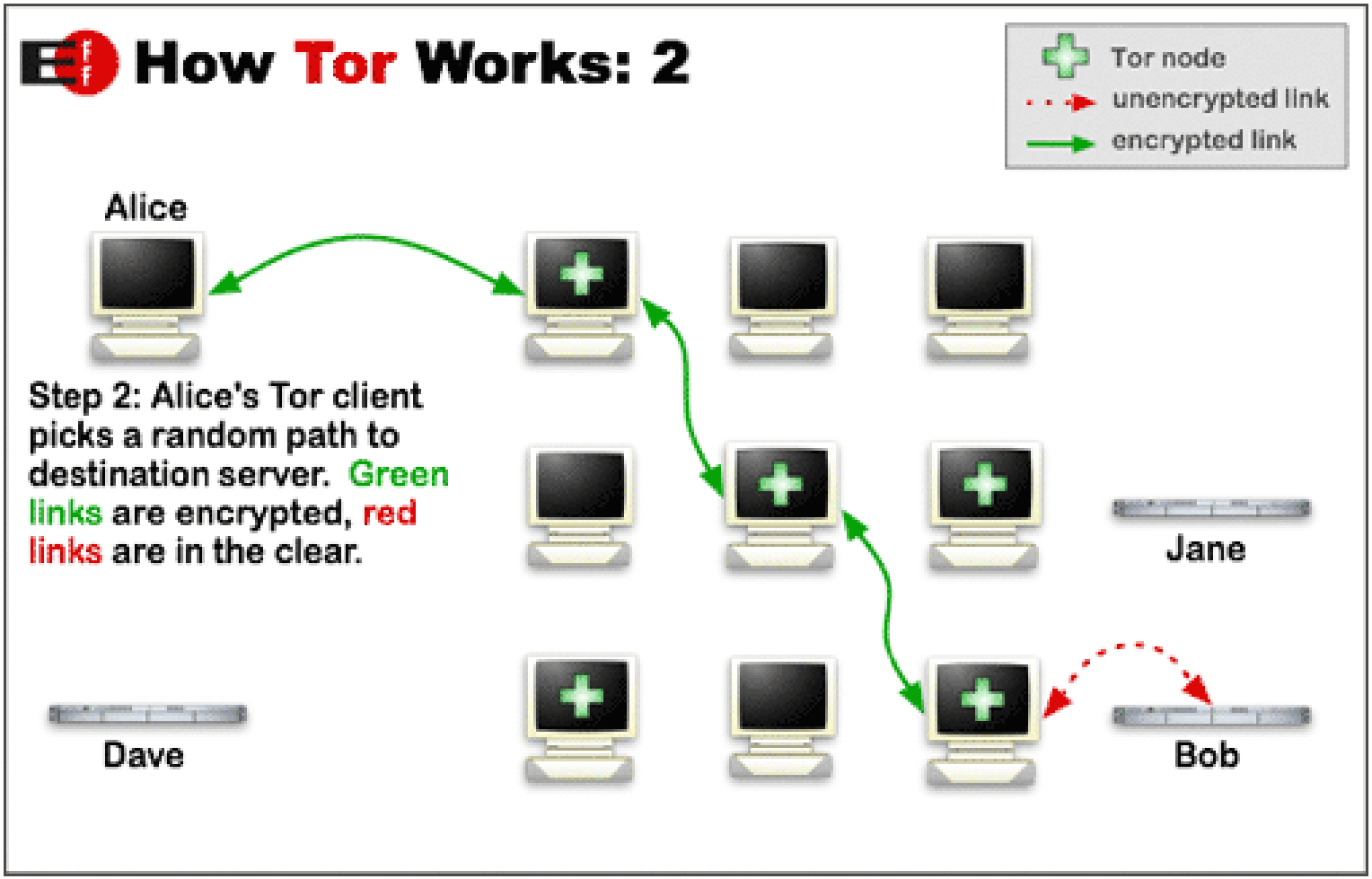
Brian Neil Levine

# Private Communications?




- Does link encryption provide privacy?
- VPNs, SSH tunnels, WEP/WPA, etc.

# Anonymous Communication?



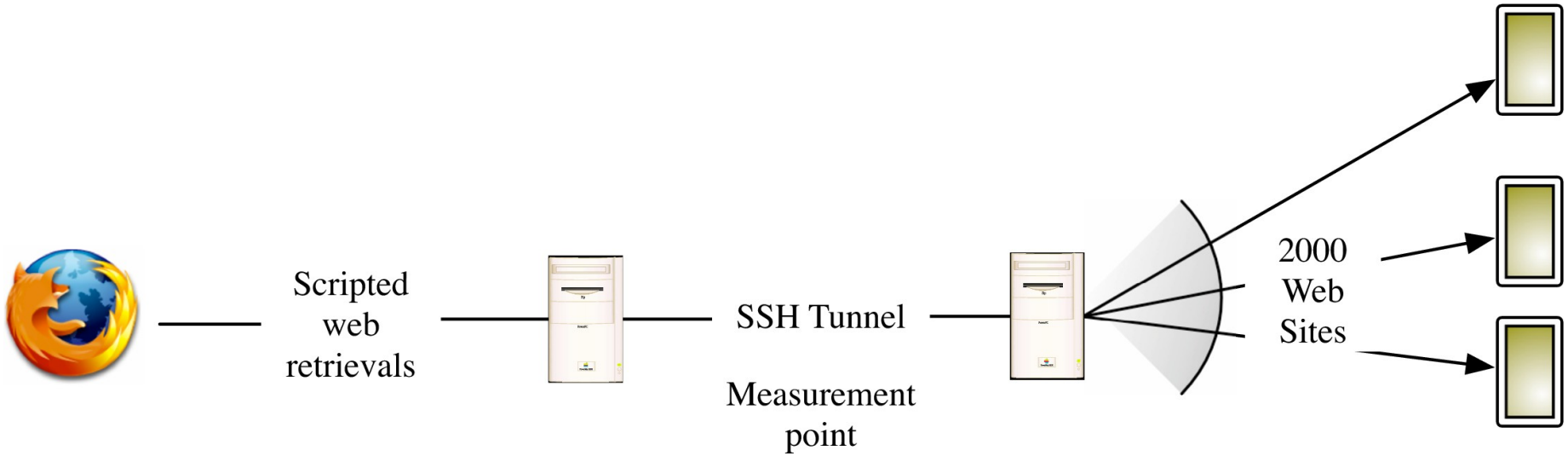
# Not so Private (or Anonymous)!

- We identify responder >75% of the time – no cryptanalysis
- Identification based on stream characteristics
  - Packet size, direction, count
  - No timing information
- Profiles constructed off-line

# Outline

- Introduction
- **Outline**
- Observer Model
- Data Collection
- Classifiers
- Evaluation
- Implications
- Related Work
- Conclusion

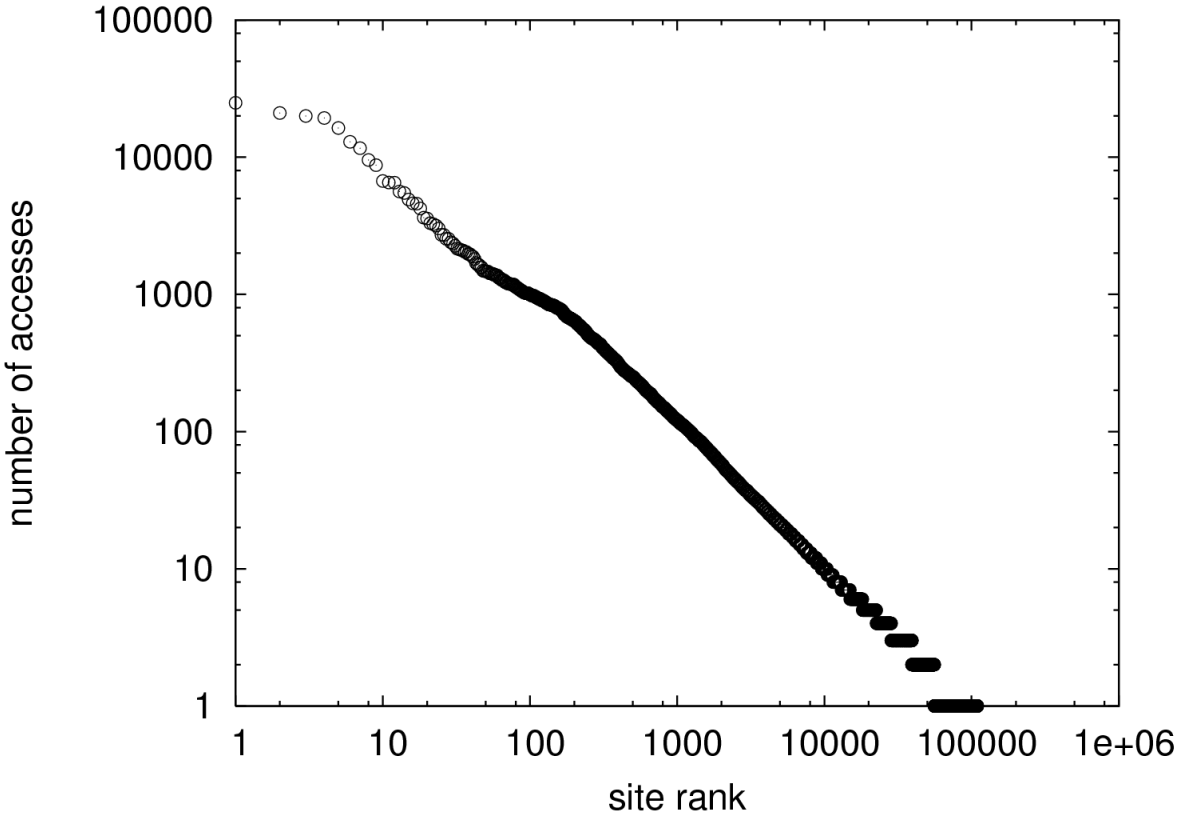
# Observer Model



# Data Collection

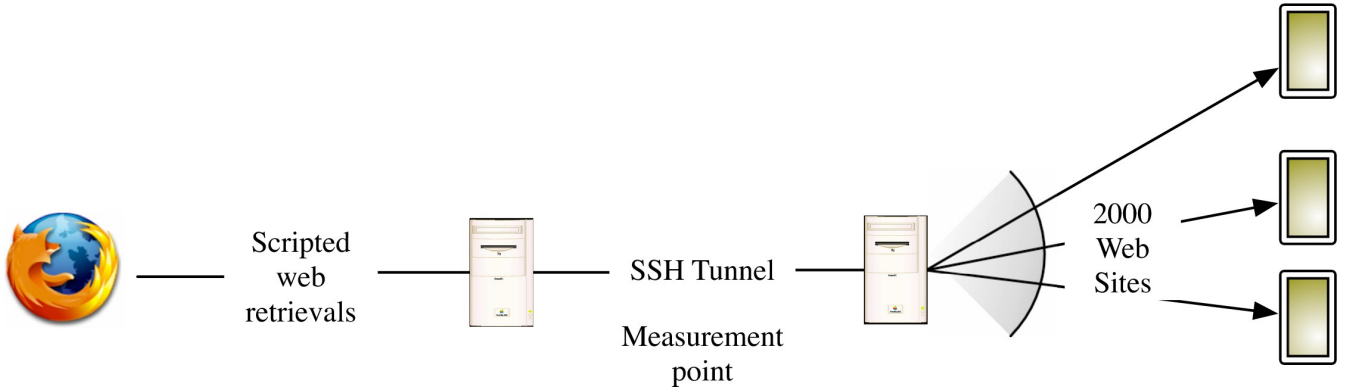
- Logged requests to department DNS server for one month
- Heuristically refined the logs
- Connected to each site on port 80, resolved redirects

# Data Collection 2




- We utilized the top 2000 (by rank); top 64% (by accesses)

# Data Collection 3




- 2000 sites retrieved every six hours for two months
- Firefox fetches over ssh tunnel to proxy
- Collect logs via tcpdump

- Logs available at http://traces.cs.umass.edu/

# Detour: Classifieres

- Goal: build profiles of sites, and match unlabeled traces against them
- This is analogous to the supervised learning problem:

| | |
|---|---|
| Class: | Remote site name |
| Instance: | Log of site retrieval |
| Attribute: | Packet size/direction |
| Value: | Number of packets |
| Training set: | Observer-initiated set |
| Test set: | User-initiated set |

# Classifiers

- Classifiers construct a model based on labeled training data
- The classifier then uses the model to output class membership probabilities of unlabeled test data, e.g.:

| | |
|---|---|
| 0.43 | nytimes.com |
| 0.11 | amazon.com |
| 0.02 | google.com |
| . | . |
| . | . |
| . | . |

# Classifiers: Jaccard's Coefficient

- An instance X is modeled as a set
  - Attributes are direction and size; value is presence
  - Elements in the set are of the form {direction, length}
  - Multiple training instances form a single set by majority

- Similarity of two sets:

$$S(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

- Normalized to give membership probability estimates

# Classifiers: Naïve Bayes

- Assume independence between attributes
- Estimate class probabilities:

$$p(C_j|\mathbf{A}) \propto p(C_j) \prod_{i=1}^{n} p(C_j|A_i)$$

- $p(C_j|A_i)$ can be observed from data directly
  - For a given retrieval of a site, how many packets of a given {size, direction} were observed?

# Evaluation

- Experiment Setup
- Metric
- Results
- Countermeasures

# Evaluation

- How much training data is needed?
- How good are is the classification?
- How long is training data useful?
- Can classification scale with the number of sites?

# Experiment Setup

- sample: log of retrieval instances of top N web sites
- i: initial sample
- t: size of training set
- s: size of test set
- Δ:samples between test and train

# Experiment Setup 2

- An experiment attempts to classify each retrieval instance in the test set
- k-identifiability: 1 if correct class in top k, 0 otherwise
- k-accuracy: mean k-identifiability across all instances in test set

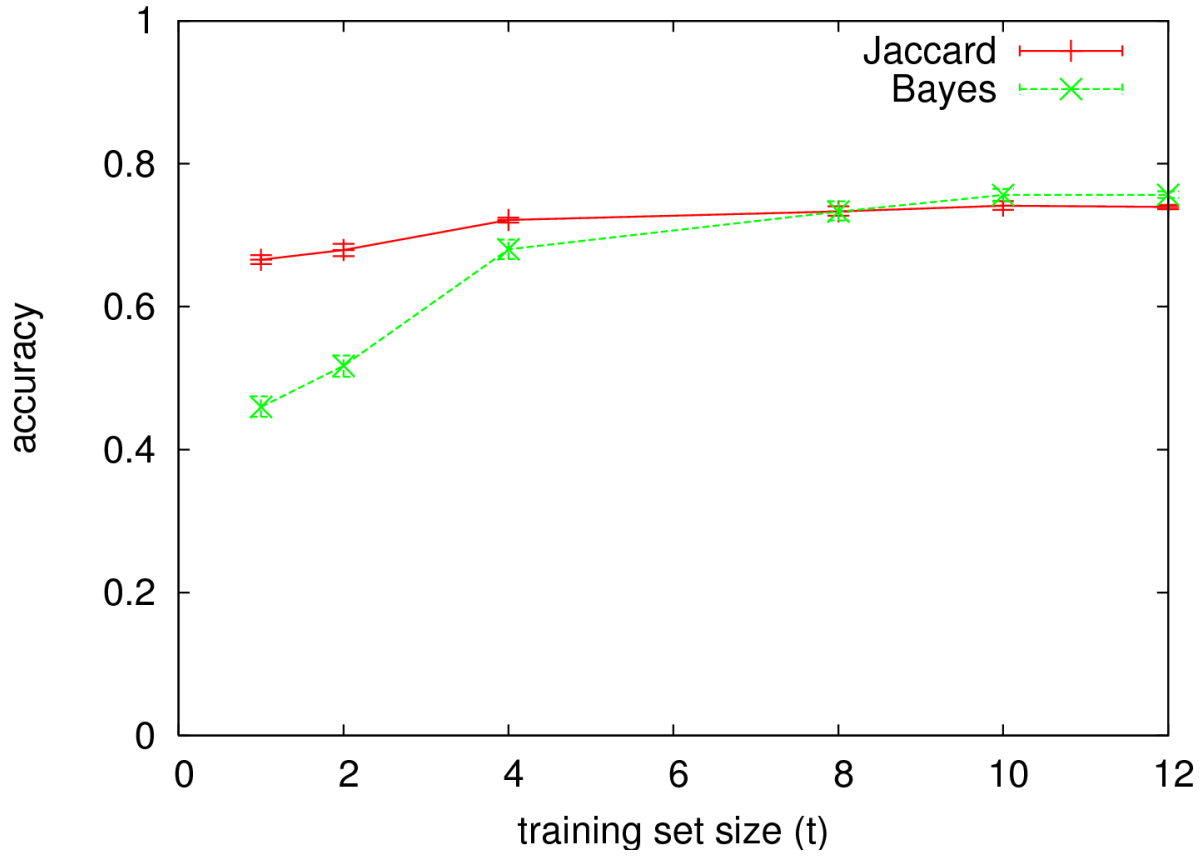
| | |
|---|---|
| 0.43 | nytimes.com |
| 0.11 | **amazon.com** |
| 0.02 | google.com |
| . | . |
| . | . |
| . | . |

# Results

- Variables:
  - k=1
  - s=one day (four retrievals)
  - t = one day (four retrievals)
  - Δ=3 (one day)
- All confidence intervals are 95%, from 10 experiments run with randomly chosen initial sample i

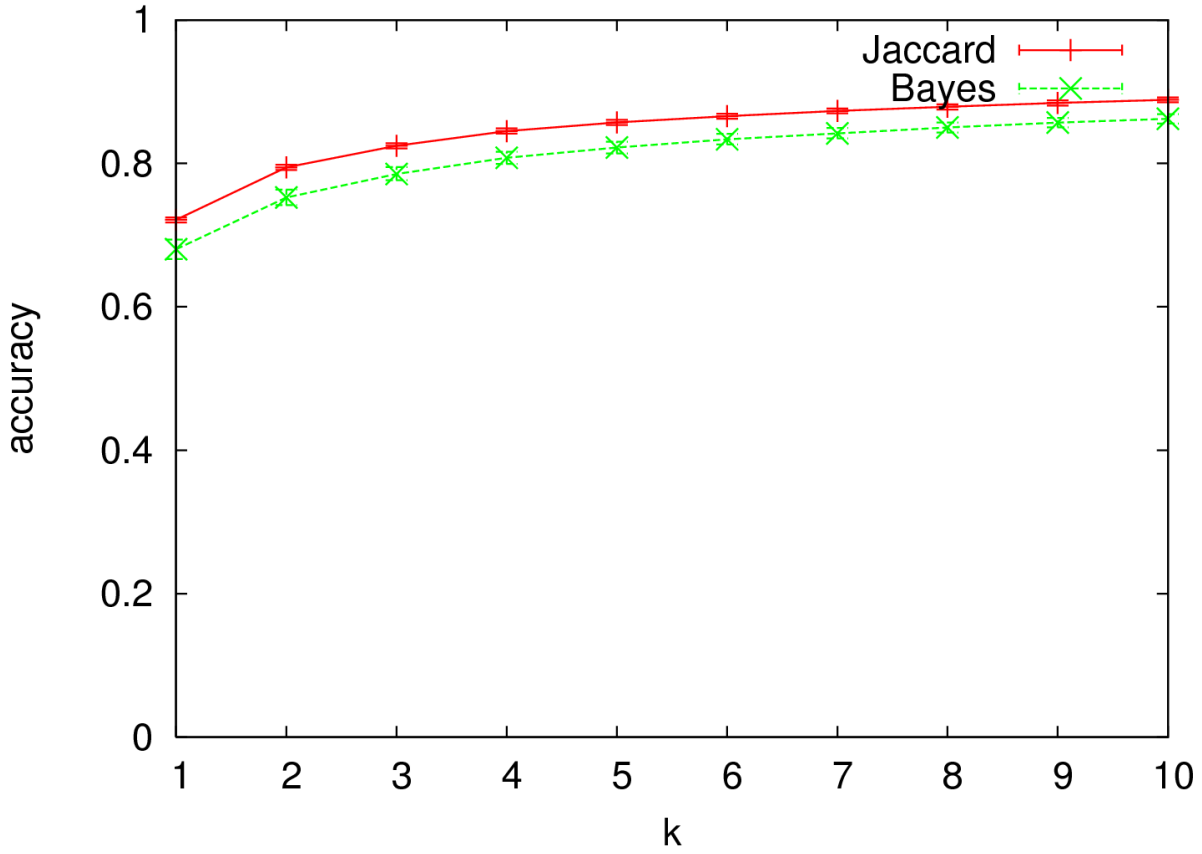
# Results 2

- More training data improves performance



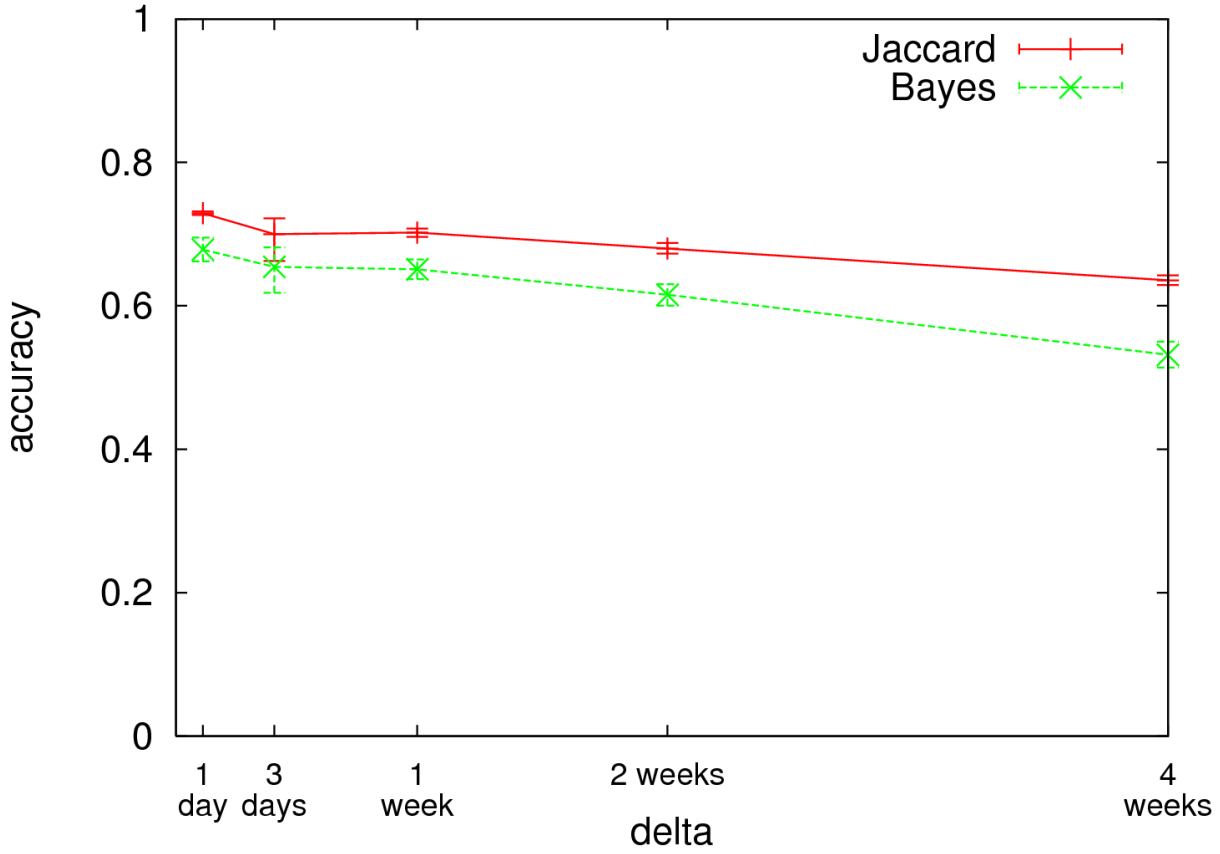
# Results 3

- Give us a few guesses and we can do better



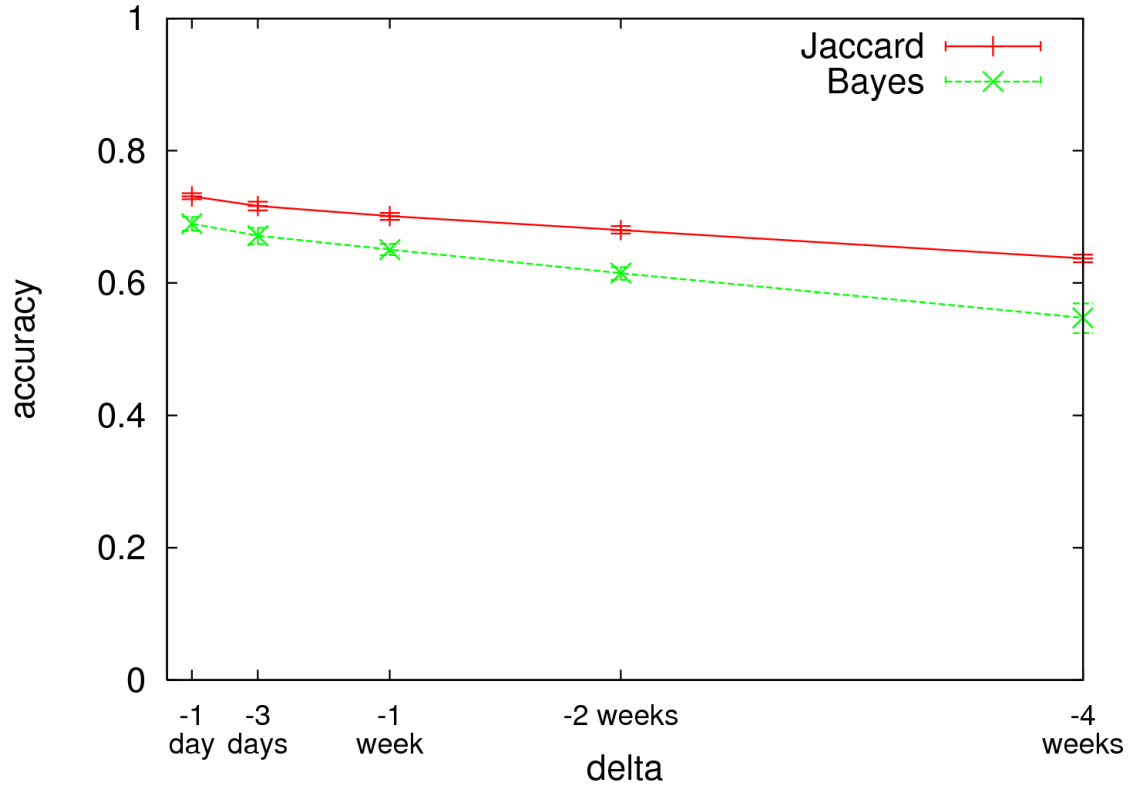
# Results 4

- Old training data is useful for long periods



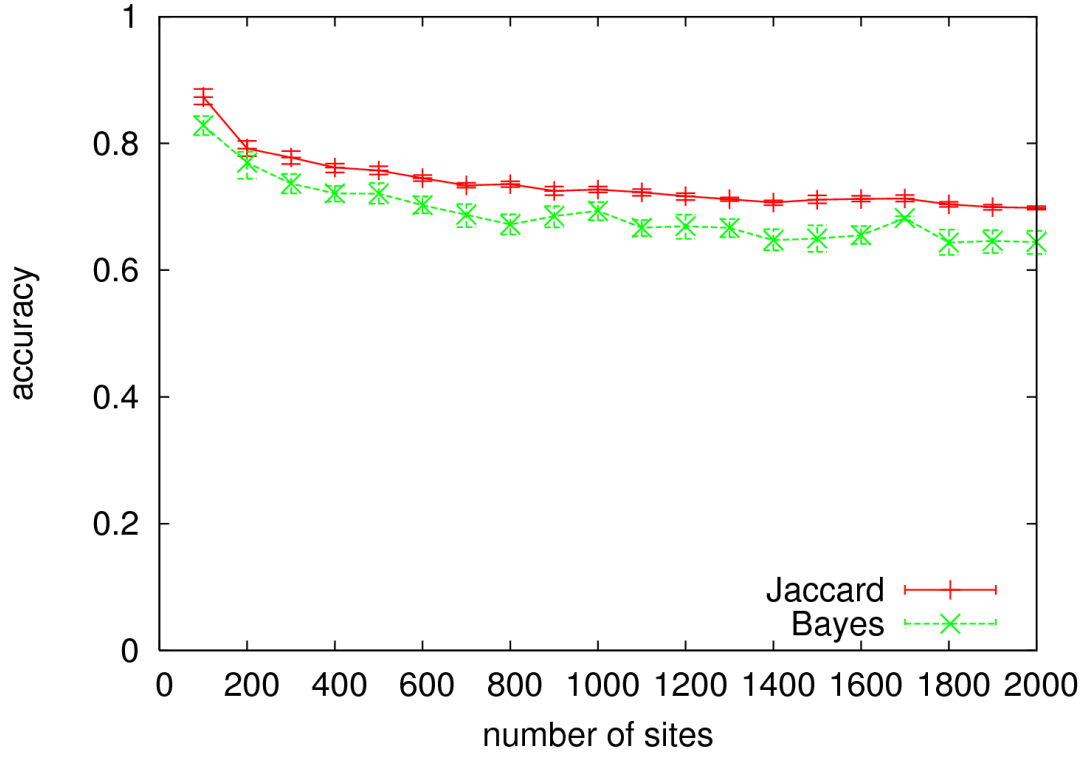
# Results 5

- Traces can be identified with training data gather a posteriori



# Results 6

- More candidate sites decrease accuracy
- Fits acc = A log N + B



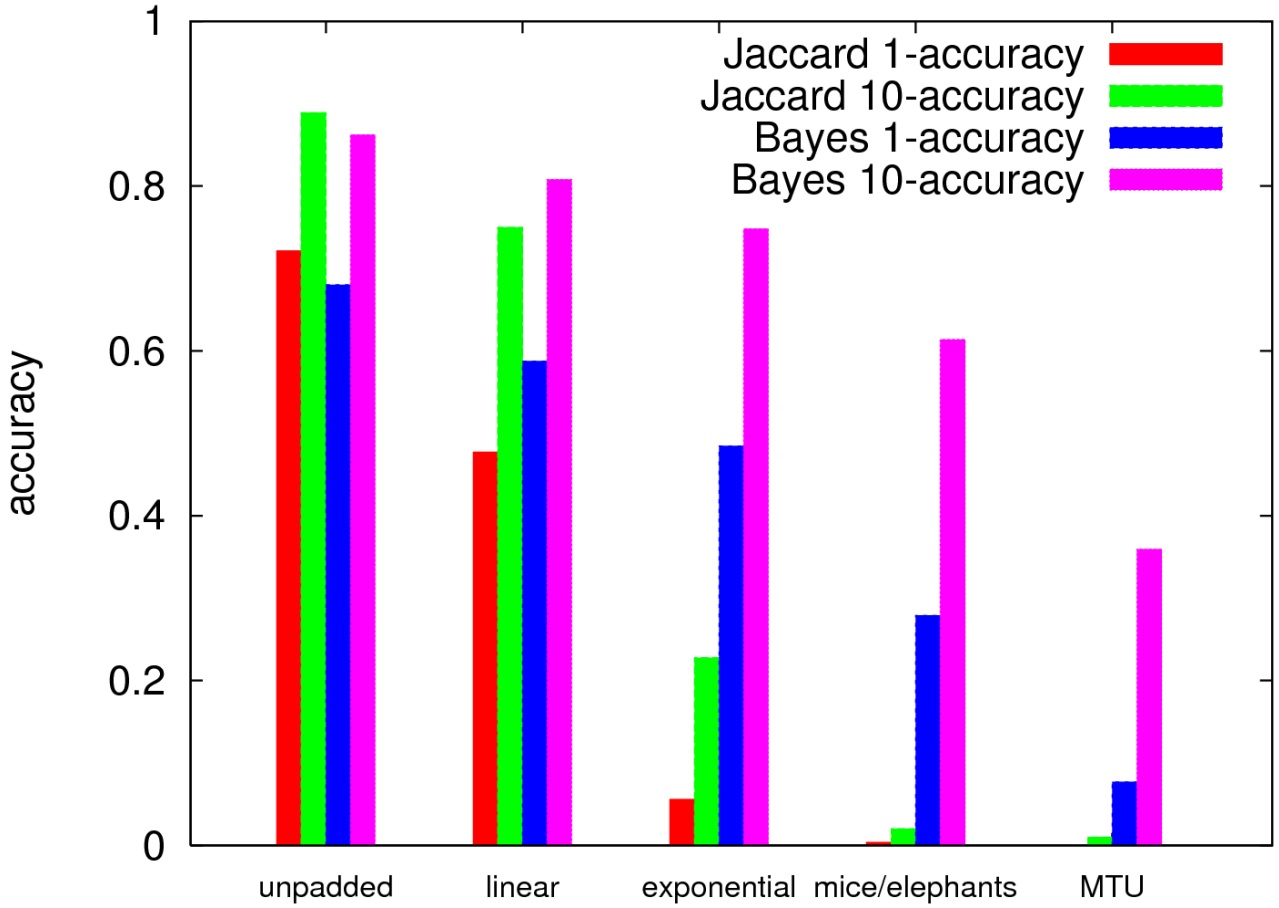
# Forensics

- Properties of interest to forensic investigators:
  - Accuracy decreases slowly over time
  - Classifiers seem to scale well with number of sites
- A distributed effort could build a complete library of profiles
  - 49M active sites (according to Netcraft, October 2006) can be retrieved weekly by 500 volunteers each collecting 600 sites/hour
  - Profiles are based on packet sizes, thus location independent
  - Profiles are small: ~350 bytes  for Bayes, less for Jaccard
  - Less than 17GB for whole Internet!

# Countermeasures

- Pad packets according to some scheme, assume the observer knows the scheme
- **linear** – pad packets to the next multiple of 50 bytes
- **exponential** – pad packets to the next power of two bytes (or MTU)
- **mice/elephants** – pad packets to either 100 bytes or the MTU
- **MTU** – pad all packets to MTU (1500 bytes)
- **tor-like** – if packet < 512 bytes, pad to 512 bytes; if packet > 512 bytes, fragment into 512 byte chunks and pad last chunk to 512 bytes

# Countermeasure Effectiveness



# Selected Related Work

- Traffic analysis: Raymond2002
- Profiling: Hintz2002, Sun2002, Bissias2005
- Passive logging / intersection attacks: Wright2002, 2003
- Countermeasures
  - Dummy packets: Fu2003
  - Defensive Dropping: Levine2004

# Conclusion / Future Work

- Profiling is easy
  - 75% accuracy with only four retrievals
  - Encryption != Privacy
- Even with the strictest padding regime, 10-acc is 38%!
  - Encryption + Padding != Privacy
- A forensic library is easy to build in a distributed fashion

- Timing information might improve attack
- Better countermeasures might mitigate this attack
  - Defensive adding/dropping
  - At what cost?