Side Channel Attack on Encrypted Traffic

Rahul Kejriwal, CS14B023

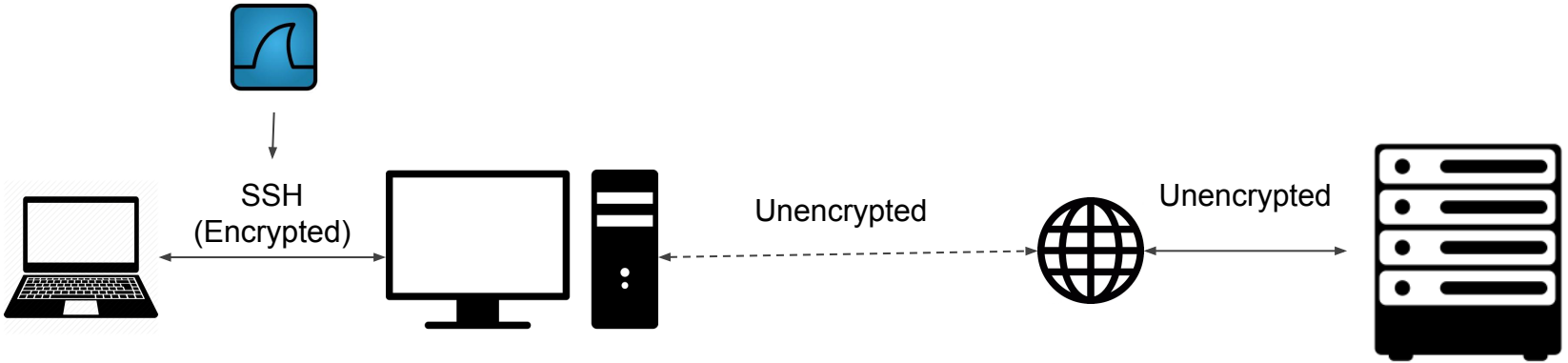
Code at: *https://github.com/kejriwalrahul/TrafficLeak*

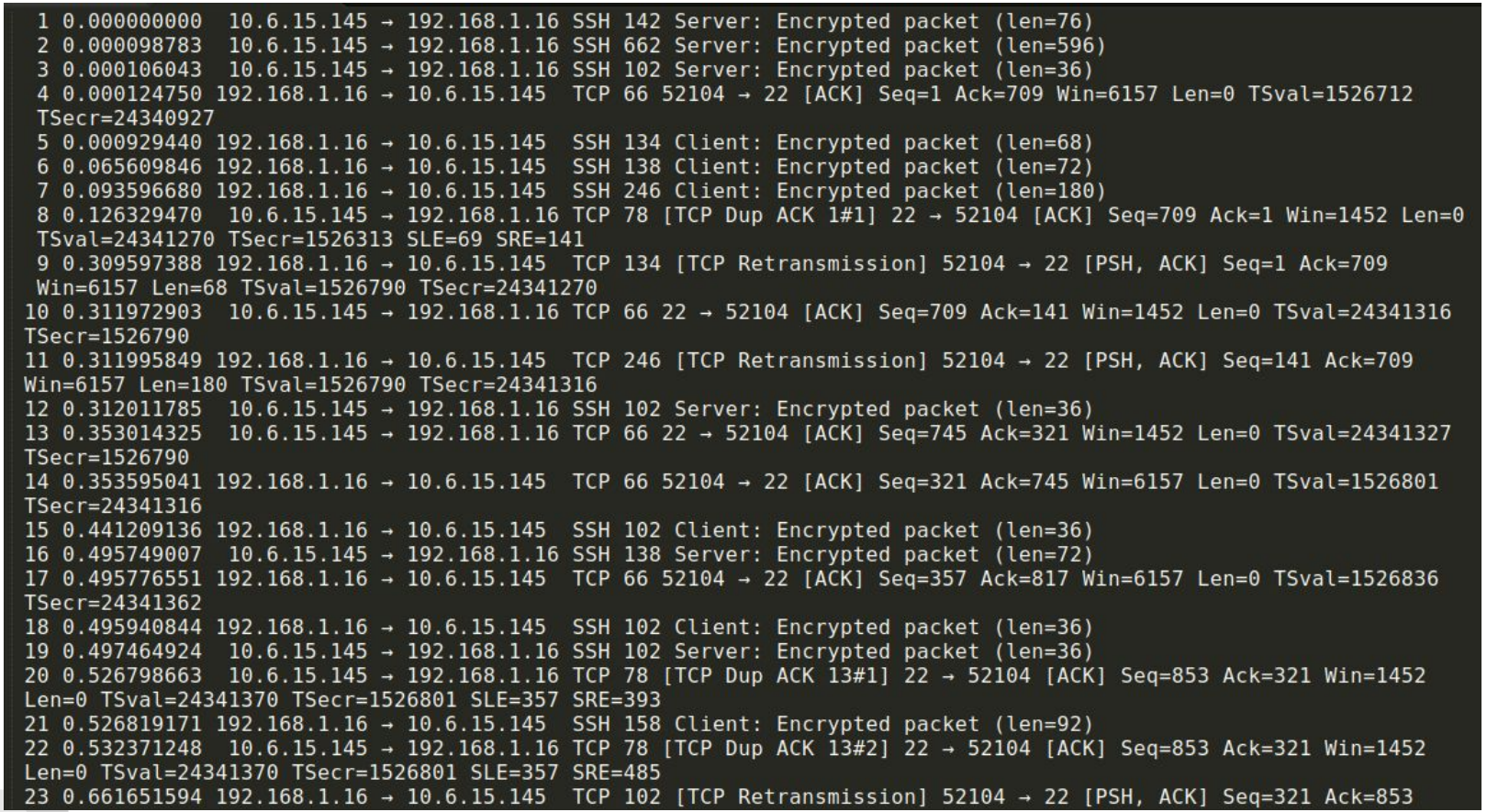
Assumptions

- Closed World Assumption
- No interleaved browsing
- Well demarcated request boundaries
- Webpage-level fingerprinting
- Caching Effects (assume hot pages)

Data Collection Methodology

- Chose top ~79 popular webpages (Source: Wikipedia)
- No highly similar webpages (Ex: google.com and google.co.in)
- Caching effects (Collected hot traces)





Sample Trace

Model 1: Ad-Hoc Model

Feature Extraction

- Three major sources of information:
  a. *Packet size distribution* (and direction)
  b. *Timing distribution* (rate of transmission)
  c. *Ordering of packets*

- Features used:
  a. *Packet size distribution*
     i. # of packets (from client and server)
     ii. Avg Pkt Length (from client and server)
     iii. Std. Dev. of Pkt Lengths (from client and server)
     iv. # of full pkts (from client and server)
  b. *Timing distribution*
     i. Total Time
     ii. # of lag periods

Models

- 4-fold cross-validation on dataset
- 79 websites with 4 traces each = 316 training instances
- Best accuracy of ~67% using ExtraTrees Classifier

Sr. No.	Model (sklearn models)	Avg. Accuracy (after 4-fold cross validation)
1	k-NN	55.24%
2	LDA	60.02%
3	Logistic Regression	52.68%
4	SVM (Linear Kernel)	58.40%
5	SVM (Poly Kernel)	59.36%
6	SVM (RBF Kernel)	56.52%
7	Decision Trees	51.11%
8	Random Forests	66.05%
9	**ExtraTrees**	**66.99%**
10	MLP	40.01%

Model 2: SVMs using Damerau-Levenshtein distance Model (Discretized Packet Size)

Src: Xiang et al. [2]

Strategy

1. Request packets are likely to get reordered, broken up and/or merged at different times.
   a. Use *Damerau-Levenshtein distance* to measure edit-distance (similarity) between multiple traces. This captures insertion, deletion, substitution and transposition of requests. (Cost schemes are hyperparameters.)
   b. Discretized packet size into 24 levels (hyperparameter) for computing DL distance.

2. Normalize DL distance:

$$L(t, t') = \frac{d(t, t')}{\min(|t|, |t'|)}$$

3. Build kernel using DL distance for SVMs (γ is hyperparameter):

$$K(t, t') = \exp(-\gamma L(t, t')^2)$$

Results & Discussion

- Xiang et al. [2] unfortunately do not give us the complete cost schemes used for computing DL distance.

- Avg. accuracy (after 4-fold cross-validation) was **75.26%**.
  - Easily outperforms the ad-hoc models considered earlier.

- Xiang et al. trained the model using **20-40 cold traces** and got **80-85%** accuracy.
  - We used only **4 hot traces/webpage**.
  - Possible to improve our accuracy by increasing # of traces.

- **Advantages:**
  - Unlike ad-hoc models, this one scales well with increasing # of websites.
  - Robust against many proposed traffic obfuscation based security measures.

Model 3: DLSVM Model (No packet size information)

Src: Xiang et al. [2]

Strategy & Results

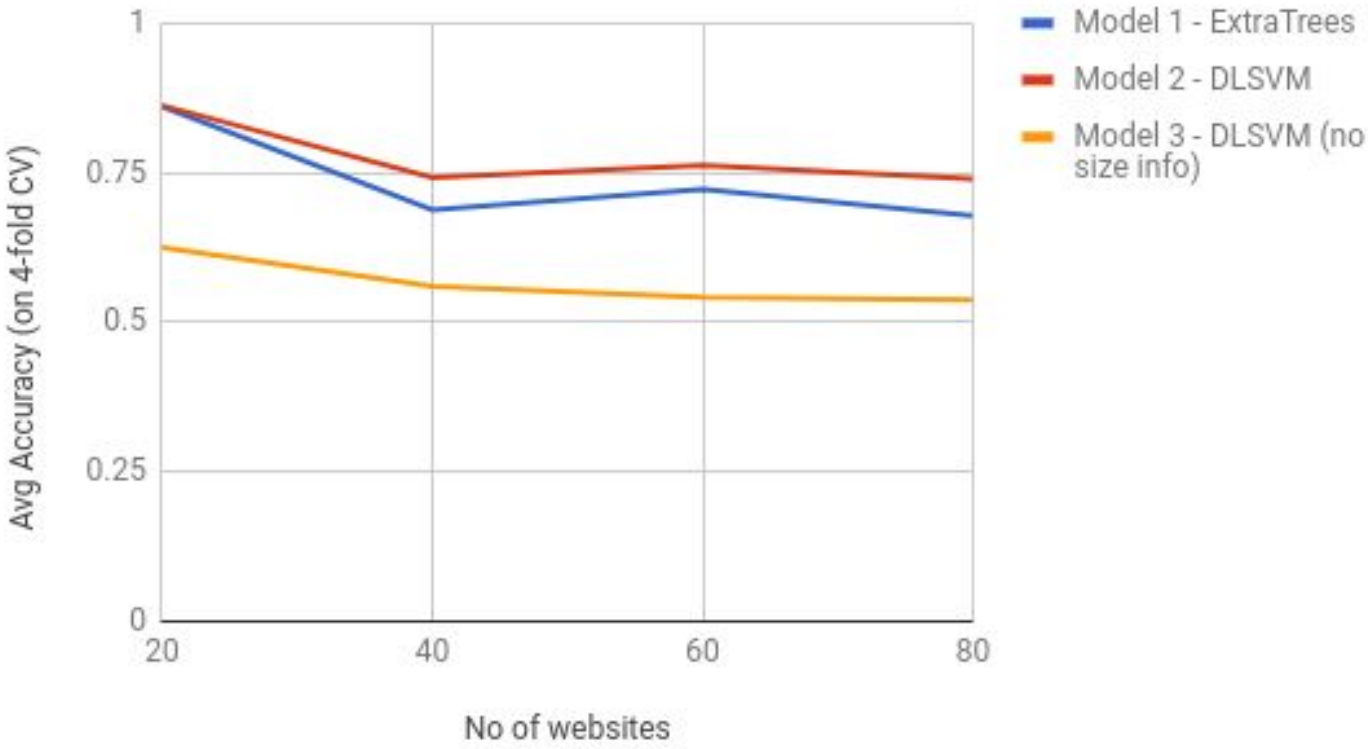
- Almost same model as previous one.
  - Only 1 level of packet size, sign gives direction.

- This model uses **only** ordering information and ignores timing and size distribution present in the trace.

- Avg. accuracy (after 4-fold cross-validation) was **53.96%**
  - On-par with most ad-hoc models considered earlier.

- Still performs quite well (as good as many ad-hoc models)
  - Order of requests contains lot of information ignored by previous work

Scalability



Differences wrt Xiang et al.

Procedural Differences

- Worked on **hot traces** rather than **cold traces** which is more difficult to distinguish.
- Used only SSH packets for experiments. Xiang et al. also use TCP level packets.
- Used much lesser training instances (4 per website as compared to 20-40 per website).
- Used different cost schemes for DL distance and different hyperparameter configurations.

Possible Further Work

Relook at some Assumptions

- Closed World Assumption:
  Larger # of negative instances needed for training the model.

- Webpage-level fingerprinting:
  Xiang et al.[2] have given a strategy to extend this webpage fingerprinting to create fingerprints for websites using HMM models

- Caching Effects (assumed hot pages):
  Cold pages are actually easier to distinguish due to the larger # of differing requests.

Security Counter-Measures

Some Counter-Measures

- Obfuscation based techniques to prevent side-channel attacks:
  - Reorder, merge, split requests
  - Make dummy requests
  - Add dummy data in headers

- Proposed & Used Techniques:
  - Randomized Pipelining over Tor
  - HTTPOS
  - Cover Traffic
  - Traffic morphing

- Xiang et al. showed that such schemes are not very secure against DLSVM based attacks.

References

1. Cai, Xiang, et al. "Touching from a distance: Website fingerprinting attacks and defenses." Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012.

2. Herrmann, Dominik, Rolf Wendolsky, and Hannes Federrath. "Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier." Proceedings of the 2009 ACM workshop on Cloud computing security. ACM, 2009.

3. Perry, Mike. “A Critique of Website Traffic Fingerprinting Attacks.” Tor Blog, Tor Blog, 7 Nov. 2013, blog.torproject.org/critique-website-traffic-fingerprinting-attacks.