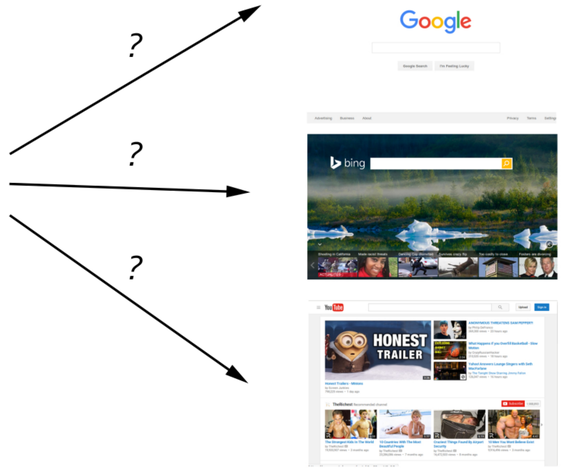
Recognizing web traces of various web services

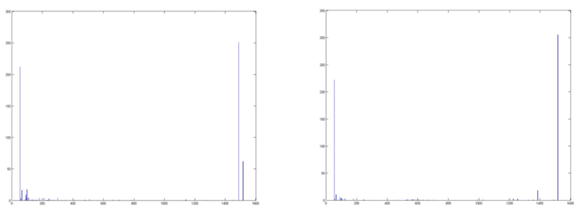
Phase 1
Identifying loaded website

Time	Length
14.255978000	54
14.258290000	252
14.263714000	54
14.266498000	1514
14.266664000	1514
14.266725000	54
14.266835000	568
14.274365000	180
14.274636000	505
14.277985000	54
14.278402000	296
14.315759000	783
14.315866000	54
14.349857000	482
14.357416000	1514
14.357593000	1514
14.357635000	54
14.362813000	1514
14.362955000	1514
14.362990000	54
14.363076000	1514



Features:
Each network capture is represented as a 1514-dimensional vector x :
x_i : number of packets of length i

Distribution of packet length depends on website loaded:



Google Wikipedia

Dataset:
100 training examples for each of the following websites:
Google, Wikipedia, Bing, Youtube, Amazon, Facebook, Yahoo

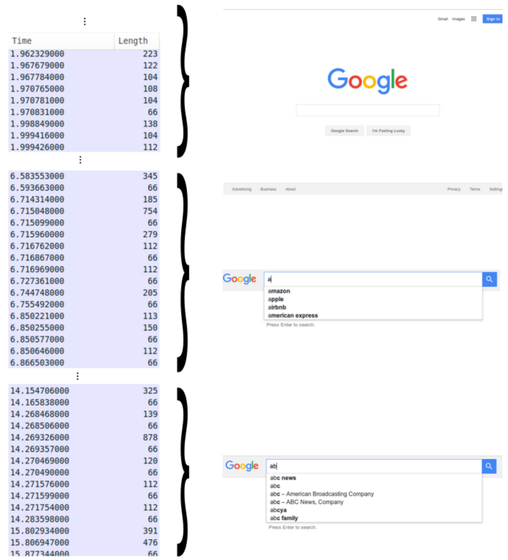
Method:
Naive Bayes multiclass classifier with multinomial event model

Testing:
5-fold Cross Validation on this dataset yields average error:

0.14 %

Conclusion: Excellent performance on differentiating between 7 of the most popular websites

Phase 2
Locating the user action



Goal : finding the traces corresponding to auto-completion usage

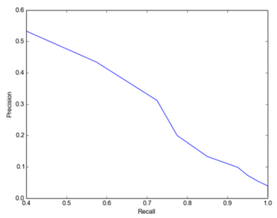
Training set:
- 200 traces of auto-completion usage
- 20,000 packets of standard web traffic (no auto-completion)

Test set :
- 200,000 packets of "mixed" web traffic (random websites + auto-completion)

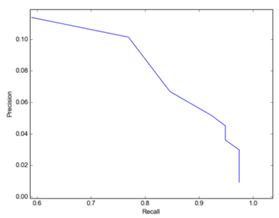
Algorithm : Naive Bayes with multinomial event model.
Sliding window to consider n (=20) packets at a time

Probability of being an auto-completion trace (*p(y)*) can vary in order to change precision and recall.

Precision-Recall graph for Google auto-completion



Precision-Recall graph for Bing auto-completion



Conclusion : low precision rate, especially for Bing. Could be improved by further filtering the result using the classifier derived in step 1.

Together, these two phases would allow us to scan through large webtraces and locate the data containing information about the usage of auto-completion features. Applying previous work involving side-channel attacks, we would then be able to retrieve the words searched for, using only the captured packet sizes.