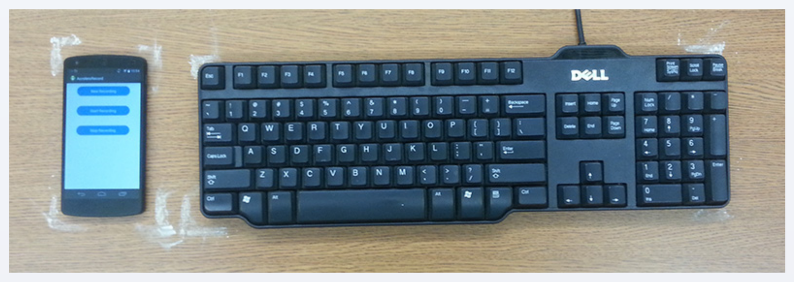
EavesDroid: Keystroke Recovery using Mobile Phone Accelerometers

Jennifer Guo   Yi-Hsien Lin   Akshay Mittal   Wathsala Vithanage

Department of Computer Science

Threat Model

- Smartphone accelerometers do not require explicit user permissions
- Smartphones are often placed next to the user's laptop or keyboard
- Malicious applications can recover text by identifying signals from different keystrokes

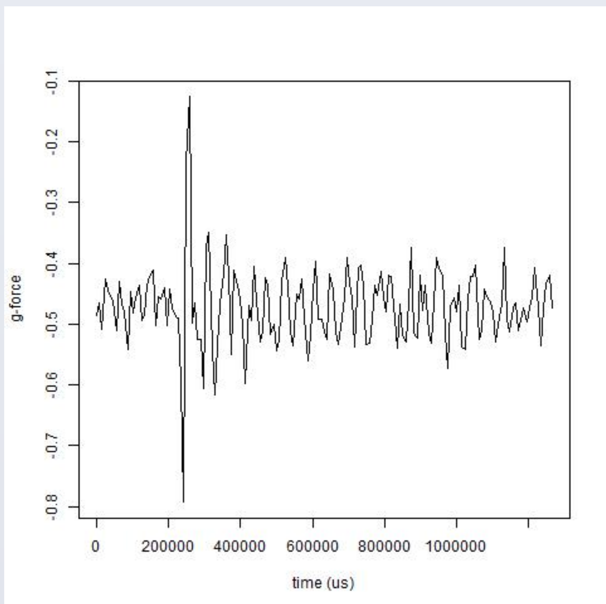


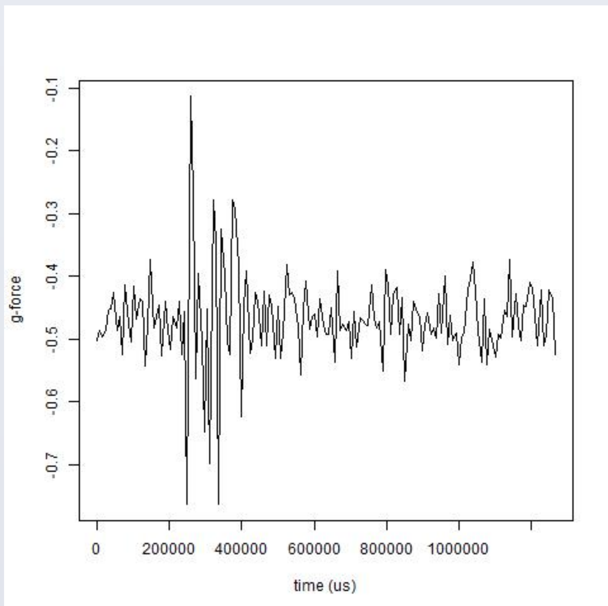
Contributions

- Develop an infrastructure for characterizing keypress vibrations
  - Captured, analyzed and built profiles of keypresses on a nearby keyboard based on the generated vibrations
  - Successfully recovered words using Boosted Decision Stumps
- Dataset made publicly available
  - Provided noise-free signal data for each letter of the English alphabet
  - Developed an infrastructure for analysis and extraction of features

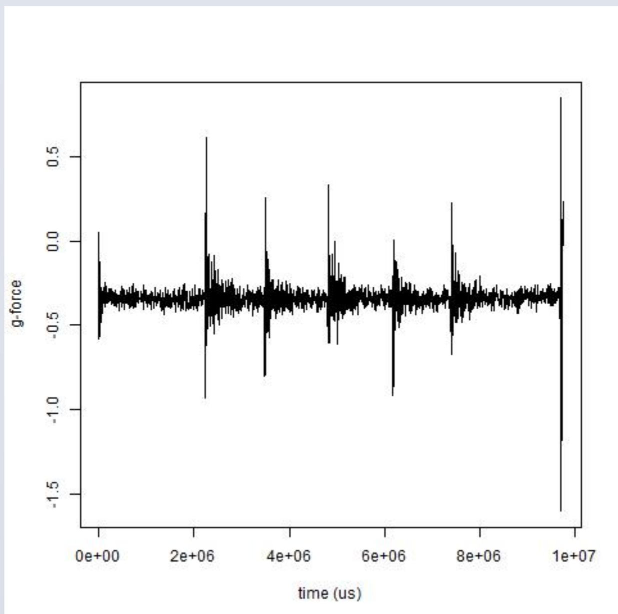
Dataset Overview

- Recorded 1000 data points in 40 sessions with 25 letters in each sessions.
- Sampled vibration signals for letters 'a' and 'b' are quite distinct
  - Letter 'a' consists of one peak while letter 'b' is consist of three





- Sample vibration signal for the word "juice"
  - Shows distinct peaks for each letter
  - Clipper module removes the beginning and ending noise

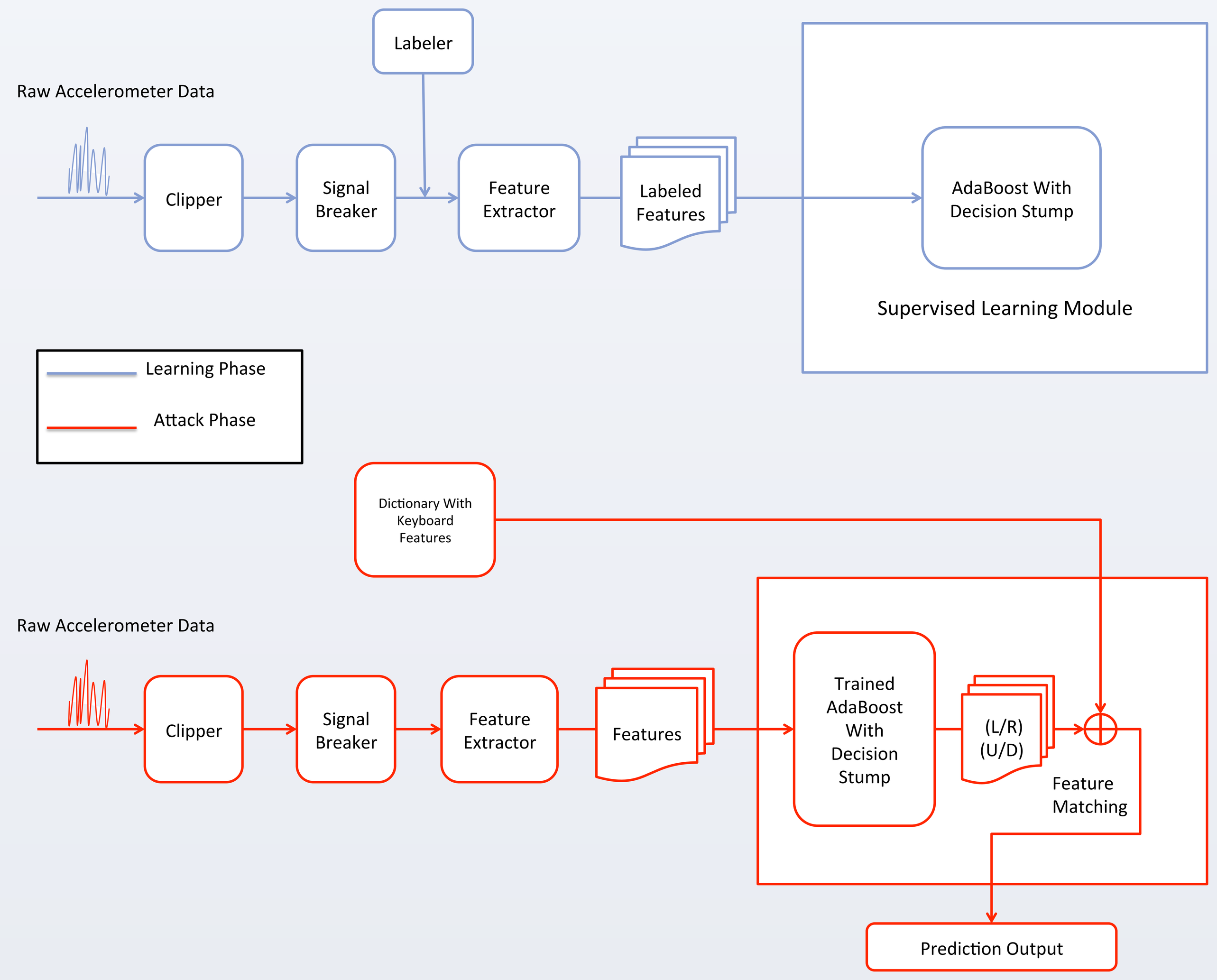


Left(L)/Right(R), Up(U)/Down(D) and Triads

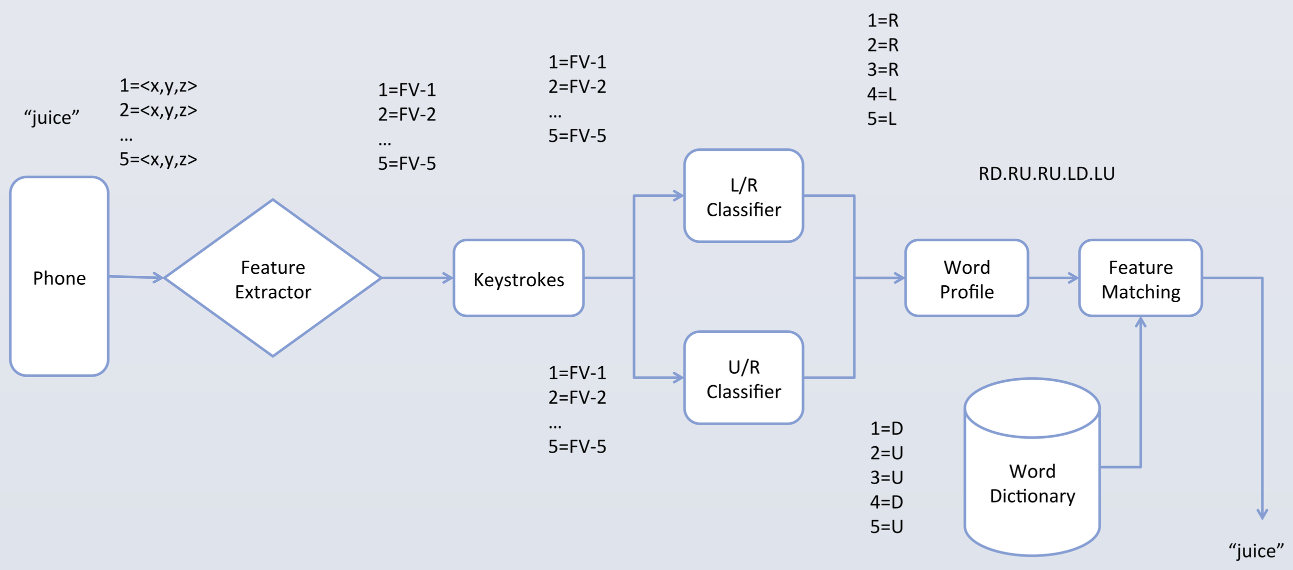
- Provides coarse grained labeling compared to exact alphabet labeling
  - Letters on the left side of and including T, G, B – Left (L)
  - Letter in the top row – Up (U)
  - Adjacent keys grouped into triples

Data Processing Architecture

- Features extracted
  - mean, rms, skewness, kurtosis, variance, min, max
  - Fast Fourier Transformation coefficients (30)
- Classifier used: AdaBoost with Decision Stump as weak learner



- SignalBreaker module breaks the word signal into individual letters
- Predictions made by the classifier are matched against a dictionary
  - Used Harvard sentences as the candidate dictionary
  - Labeled dictionary using L/R and U/D labels
  - Each n-letter word gets 2n-labels
  - Hamming distance with each dictionary word is computed
  - Output dictionary words with lowest Hamming distance
- Example



- Input: A sentence from the Harvard dictionary
- Output: All candidate words that constitute this sentence
  - Exact matches are shown without other candidates
  - Partial matches shown with lowest Hamming distanced words
- EavesDroid is able to recover 1-3 lettered words (such as 'box', 'key', 'in') while Marquardt *et. al* work relies on human identification

```
    Typed Text: Glue the sheet to the dark blue background.
Recovered Text: Glue the sheet to the dark blue background.
                blue     juice        days glue
                corn     depth        work size
                hard     canoe        lack four
                 .        .            .    .
                 .        .            .    .
                 .        .            .    .
```

Experimentation

- AdaBoost + Decision Stump achieves comparable accuracy with AdaBoost + Random Forests and Neural Networks.
- By the principle of Occam's Razor, we chose AdaBoost + Decision Stump as the classifier of **EavesDroid**

Labeled Dataset	Algorithm	Test Accuracy (%)
L/R	AdaBoost (RandomForests)	68.67
	AdaBoost (DecisionStumps)	69.82
	Neural Networks	68.10
U/D	AdaBoost (RandomForests)	56.60
	AdaBoost (DecisionStumps)	58.68
	Neural Networks	58.62
Triads	AdaBoost (RandomForests)	16.37
	AdaBoost (DecisionStumps)	13.21
	Neural Networks	14.65

- Data set = Training set (66%) TR + Test set (33%) TE
  - Prediction model built using TR
- Experiment 1: 72 sets of Harvard sentences (4490 words)
  - Represents words using individual letter signals from TE
  - Among 4490 words
    - 85.67% have less than 5 labeling errors

# labeling errors (L/R, U/D)	Recovered Words Accuracy (%)
0	5.46
1	23.41
2	41.64
3	70.31
4	85.67
5	94.54
6	97.27

- Experiment 2: New York Times article (138 dictionary words + 257 new words)
  - 121 out of 138 dictionary words have less than 5 labeling errors
  - This demonstrates practical applications of EavesDroid

Challenges and Future Work

- Number of candidate words increases along with dictionary size
  - Leads to additional work in contextual identification
  - Reducing search space is a possible solution (grouping letters into triads)
- Accelerometer is extremely sensitive to surrounding noises
  - Better signal filtering techniques required
  - With advanced signal filtering techniques profiling of consecutive key presses into the model
- Clustering on the letters' signals and labeling them based on the frequency of each letter's average appearance
- Building the model on a word's signal instead of the letter's signal might identify word signatures with greater accuracy

Conclusion

- Provide a simple model based on AdaBoost + Decision Stump and achieve comparable accuracies to a Neural Network based model used by Marquardt *et. al*