Robust Keystroke Transcription from the Acoustic Side-Channel

David Slater, Scott Novotney, Jessica Moore, Sean Morgan, Scott Tenaglia

Two Six Labs, LLC
Arlington, Virginia, USA

ACSAC, December 9-13, 2019 San Juan
Keystroke Transcription – Problem Motivation

Primary Channel (Keystrokes)
Keystroke Transcription – Problem Motivation

Primary Channel (Keystrokes)

Side Channel (Acoustic)
Keystroke Transcription – Problem Motivation

Primary Channel (Keystrokes)

Side Channel (Acoustic)

Inferred Keystrokes
Outline

• Breaking Assumptions of Prior Work

• Attack Methodology: Joint Keystroke Detection and Classification

• Results

• Conclusion
Breaking Assumptions

Prior work almost uniformly assumes that the true positive rate (TPR) and true negative rate (TNR) of keystroke detection is 100%.

Prior work reports keystroke classification accuracies as high as 94%.

Data Collection

Prior Datasets: Small and Unavailable

- Keystrokes collected via standard software keylogger
- Audio recording using high-quality microphone
  - Placed < 1 meter to the left of the keyboard
  - Timing synchronized by audio chirp
- 17 users – Two Six Labs employee volunteers
  - IRB prevents sharing the dataset
- 4-5 typing sessions each (76 sessions total)
  - Fixed, variable, and free text formats
  - English and code tasks
  - ~1100 keystrokes per session (86k keystrokes total, 8.2 hours of audio)
Acoustic Waveform of "Ideal" Keystroke
Assumption: Key Detection and Segmentation are Trivial
Acoustic Waveform of “Typical” Keystrokes
Assumption: Key Detection and Segmentation are Trivial
Single User Key Clustering (t-SNE)
Assumption: Key Acoustics Are Consistent Across Users
Multiple User Key Clustering (t-SNE)

Assumption: Key Acoustics Are Consistent Across Users
Attack Methodology

Preprocessing
- Filter Noise & Generate 10 ms Power Levels via FFT
- Filter Noise, Downsample, & Generate 10 ms Spectrograms

Detection
- Detect Keystrokes via Power Threshold

Feature Extraction
- Segment Audio & Generate 40 ms Spectrograms

Jointly Detect and Classify Keystrokes via Recurrent Neural Network

Classification
- Classify Keystrokes via Logistic Regression

Transcript
- "T", "N", "E", ...
- "T", "H", "E", ...

Transcript
Input: Spectrograms

Convolutional Layers

Recurrent Layers

Dense Layer

Output: Key probabilities

\[
\{ p_a^{(1)} p_a^{(2)} \ldots p_a^{(T-1)} p_a^{(T)} \} \\
\ldots \\
\{ p_z^{(1)} p_z^{(2)} \ldots p_z^{(T-1)} p_z^{(T)} \}
\]

\[ \arg\max_{k \in \{a, b, \ldots, z, \}} \frac{p_k^{(i)}}{t} \quad _{t} _{h} _{h} _{e} \]

Final Output: Key Transcript

• Spectrogram sequence of typing audio

• Deep neural network learns relationship between sounds & keystroke
  • [Adapted from Deep Speech 2]

• Probability distributions over keys
  • “_” indicates no keystroke occurred

• Predicted sequence of key classes

• Blanks and duplicates removed
  • [Insert Language Model Here]
Model Training and Evaluation

• 4-Fold Cross Validation
  • Stratification per Session (Known Typist) or User (Unknown Typist)

• **RNN Training**: split sessions into 20-second cuts
  • Paired audio and keylogger transcripts (no detailed timing information)
  • Loss function: connectionist temporal classification (CTC)

• **Baseline Classifier Training**: based on correctly pre-segmented audio
  • Keystroke detection: power threshold set based on global optimum for minimizing CER
  • Model based on Compagno et al. 2017 “Skype & Type” paper

• **Primary Metric**: Character Error Rate (CER)
  \[
  CER = \frac{\#\text{insertions} + \#\text{deletions} + \#\text{substitutions}}{\#\text{reference}}
  \]
<table>
<thead>
<tr>
<th>Key Timings</th>
<th>RNN</th>
<th>Oracle</th>
<th>Baseline</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>CER (Mean)</td>
<td>7.41%</td>
<td>14.3%</td>
<td>36.0%</td>
<td>86.7%</td>
</tr>
</tbody>
</table>

[Session stratification on English task]
Breakdown of Baseline Model [Skype & Type]

• **Keystroke detection:** power threshold – 40 ms segmented windows

• **Keystroke classification:** logistic regression (on true positive detections)

• **End-to-end performance:** detection followed by classification (CER)

<table>
<thead>
<tr>
<th>Task</th>
<th>Stratification</th>
<th>Detection</th>
<th>Classification</th>
<th>End-to-end</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>Accuracy</td>
</tr>
<tr>
<td>English</td>
<td>Session</td>
<td>79.3%</td>
<td>95.6%</td>
<td>86.3%</td>
</tr>
<tr>
<td>English</td>
<td>User</td>
<td>79.3%</td>
<td>95.6%</td>
<td>80.2%</td>
</tr>
<tr>
<td>Code</td>
<td>Session</td>
<td>83.0%</td>
<td>92.9%</td>
<td>91.5%</td>
</tr>
<tr>
<td>Code</td>
<td>User</td>
<td>83.0%</td>
<td>92.9%</td>
<td>86.2%</td>
</tr>
</tbody>
</table>
RNN Keystroke Transcription Performance
Across Stratification [User / Session] and Task [Code / English]

- User-Code: 12.4%
- Session-Code: 10.5%
- User-English: 9.7%
- Session-English: 6.8%

Median
Importance of User Variation on Test Performance

![Graph showing the relationship between minutes of training data and character error rate for different numbers of users trained on. The graph includes three lines representing training on 4, 8, and 16 users.](image-url)
Conclusion

Robust acoustic keystroke transcription remains dangerously feasible
Future Work

• Generalize to different microphone positions and acoustic environments

• Generate *releasable* acoustic keystroke dataset

• Analyze feasibility of far-field transcription, such as with Amazon Echo

• Defenses?
Appendix
Histogram: Times from Key Release to Subsequent Key Press
Primary Metric: Character Error Rate (CER)

\[ CER = \frac{\#insertions + \#deletions + \#substitutions}{\#reference} \]

**Example Reference Text**
hello fred, i regret to inform you that you employment here must be terminated. your recent lack of effort and overall desire to attend work events has left the management - myself included - at a loss. please, don’t come back. wishing you the best, mr. t

**Example Transcribed Text (10% CER)**
‘ello fred, l reret to infor, you that you emsloument here has be termnaten. your recent lack of effort and overall desire to attend uofk etenta has the anahement – msalf included – at a loss. pleasa, donat come back. wishing you the bet, mr. t
Language Model for English Transcription (User Stratification)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CER</th>
</tr>
</thead>
<tbody>
<tr>
<td>English+Code</td>
<td>15.41%</td>
</tr>
<tr>
<td>English only</td>
<td>13.67%</td>
</tr>
<tr>
<td>English+21M Keylogger</td>
<td>10.63%</td>
</tr>
</tbody>
</table>

Table 6: Impact of adding external language model data for English transcription. Adding training data from our coding task hurts performance, while adding 21M keystrokes from an external corpus of general typing tasks helps substantially. Similar relative gains were observed when applying a language model to the baseline model.
Table 5: CER by task. The performance between fixed and variable text tasks is very similar, indicating that the model has not overfit to the keystrokes present in the fixed text, even though the deep learning model has the information capacity to do so. Further, as indicated above, the free coding task resulted in very large outliers that account for the poor performance here.
Keystokes Mis-Transcribed as “K”