No Training Hurdles: Fast Training-Agnostic Attacks to Infer Your Typing

Song Fang*, Ian Markwood†, Yao Liu†, Shangqing Zhao†, Zhuo Lu†, Haojin Zhu‡

*University of Oklahoma
†University of South Florida
‡Shanghai Jiaotong University
Background

• Typing via a keyboard plays a very important role in our daily life.
Existing Non-invasive Attacks

Software or hardware based keylogger

General principle: pressing *a key* causes subtle *environmental impacts* unique to that key
Example Attacks

Environmental change

Vibration pattern
Acoustic feature
Wireless distortion

Training Phase
Trained model
Attack Phase

Unknown disturbances
Checking training data
Keystrokes
Why Is Training A Hurdle

- Require pressed key knowledge
- No physical control of keyboard
- A user may change typing behaviors
Statistical Methods

- Frequency analysis: analyzing the frequencies of observed disturbances

A large amount of text

Letter frequency distribution in English
Question: Is it possible to develop a non-invasive keystroke eavesdropping within a shorter time?
Wireless Signal Based Attacks

- Advantages:
  - Ubiquitous deployment of wireless infrastructures
  - Radio signal nature of invisibility
  - Elimination of the line-of-sight requirement

- CSI (channel state information) quantifies the disturbances

\[ H(f,t) = \frac{Y(f,t)}{X(f,t)} \]
Outline

• Motivation
• Attack Design
• Experiment Results
• Conclusion
A CSI sample refers to an individual segment corresponding to the action of pressing a key.
CSI Word Group Generation

A **CSI word group** refers to the a group of CSI samples comprising each typed word.
Sort based on the size
Classification → Sorting → Word segmentation

CSI word group

Space-associated
Non-space-associated

Dictionary demodulation
Dictionary Demodulation (DD)

1. Feature Extraction
2. Joint Demodulation
3. Error Tolerance
4. Non-Alphabetical Impact

CSI word groups (Eg., 🌟🌟🌟🌟)

English words

dictionary demodulation

CSI word group 1 🌟🌟🌟🌟 apple

space

CSI word group 2 🌟🌟🌟🌟 hat

space

CSI word group 3 🌟🌟🌟🌟 old
Feature Extraction

- Length $L$: number of constituent letters
- Repetition $\{L, (t_1, \ldots, t_r)\}$:
  - $r$ is the number of distinct letters that repeat,
  - $t_i$ denotes how many times the corresponding letter repeats
- Inter-Element Relationship Matrix $M$

\[
M : [x_1, \ldots, x_n] \mapsto \begin{bmatrix}
  r_{1,1} & r_{1,2} & r_{1,3} & \cdots & r_{1,n} \\
  r_{2,1} & r_{2,2} & r_{2,3} & \cdots & r_{2,n} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  r_{n,1} & r_{n,2} & r_{n,3} & \cdots & r_{n,n}
\end{bmatrix}
\]

$r_{i,j} = 1$ if $x_i$ and $x_j$ are same or similar
Feature Extraction

- Dictionary: Top 1,500 most frequently used word list[1]

Uniqueness rate \( = \frac{T^p}{T} \) -- number of sets obtained

Better partitioning (distinguishability)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Uniqueness rate</th>
<th>Average set cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>0.009</td>
<td>107</td>
</tr>
<tr>
<td>Repetition</td>
<td>0.042</td>
<td>24</td>
</tr>
<tr>
<td>Relationship matrix</td>
<td>0.225</td>
<td>4</td>
</tr>
</tbody>
</table>
Joint Demodulation

- Example:
  o Type in two words: “apple” and “pen”

1) \( c_1 \| c_2 \| c_3 \| c_4 \| c_5 \) \rightarrow \( R_1 \):

\[
\begin{array}{cccccc}
& c_1 & c_2 & c_3 & c_4 & c_5 \\
c_1 & 1 & 0 & 0 & 0 & 0 \\
c_2 & 0 & 1 & 1 & 0 & 0 \\
c_3 & 0 & 1 & 1 & 0 & 0 \\
c_4 & 0 & 0 & 0 & 1 & 0 \\
c_5 & 0 & 0 & 0 & 0 & 1
\end{array}
\]

2) compute the relationship matrix for each word in \( W \), and compare each with \( R_1 \) \( \rightarrow \) Candidates: “apple” and “offer”
3) $c_6 \| c_7 \| c_8 \rightarrow \text{Candidates: \{“hat”, “old”, “are”, “pen”\}}$

4) $c_1 \| c_2 \| c_3 \| c_4 \| c_5 \| c_6 \| c_7 \| c_8 \rightarrow R_{new}$

5) Candidates $T$ of the two-word sequence,
   \{“apple||hat”, “apple||old”, “apple||are”, “apple||pen”,
   “offer||hat”, “offer||old”, “offer||are”, “offer||pen”\}

6) Generate the relationship matrix for each new candidate in $T$ and compare it with $R_{new}$

   Final result: “apple||pen”
• Input:
  - $m$ CSI word groups $\mathbf{S} = \{S_1, S_2, \ldots, S_m\}$;
  - dictionary with $q$ words $\mathbf{W} = \{W_1, W_2, \ldots, W_q\}$

• Output:
  - a corresponding phrase of $m$ words

• Observation:
  - each CSI word group $\Rightarrow$ multiple candidate words
  - each candidate $\Rightarrow$ $<\text{CSI sample, letter}>$ mapping info
Step 1: find initial candidate words for each CSI word group

$R_{\text{CSI word group}} \xleftarrow{\text{Compare}} R_{\text{each word}}$

=> match, add the word as a candidate; no match, add the CSI word group to the “undemodulated set” $U$
Joint Demodulation

Step 2 (iteratively):

(a) $T_i$: concatenation of the first $i-1$ demodulated CSI word groups; candidates for $T_i$ are \{$T_{i1}, T_{i2}, \ldots, T_{ip}$\}

(b) $S_i$: the $i$-th CSI word group;

candidates for $S_i$ are \{$S_{i1}, S_{i2}, \ldots, S_{iq}$\} (by step 1)

(c) Find new candidates for concatenated CSI word groups

\[ R_{T_i \| S_i} \leftarrow \text{Compare} \rightarrow R_{T_{ij} \| S_{ik}} \] \hspace{1cm} (1 \leq j \leq p, \hspace{0.5cm} 1 \leq k \leq q)

$\Rightarrow$ match, add $T_{ij} \| S_{ik}$ as a candidate for $T_{i+1}$;

no match, add $S_i$ to $U$ and skip to $S_{i+1}$
Joint Demodulation

- Alphabet matching: the mapping can be applied to the remaining CSI word groups and those in $\mathbf{U}$
  
  Example: the user types “deed” || “would” after the mapping is established;
Error/Non-Alphabetical Characters Tolerance

- Abnormal situations:
  - CSI classification errors
  - Typos/Non-Alphabetical Characters

A CSI sample for the letter X

Set of CSI samples for the letter

Have no candidates
Add the CSI word group to the set U

Match with invalid words
Cascading discovery failures

25 of 37
Outline

• Motivation
• Attack Design
• Experiment Results
• Conclusion
Experiment Results

• Attack system:
  - a wireless transmitter + a receiver
    (each is a USRP connected with a PC)
  - the channel estimation algorithm runs at the receiver to extract the CSI for key inference.
  - dictionary: Top 1,500 most frequently used word list

• Target user:
  - a desktop computer with a Dell SK-8115 USB wired standard keyboard
Example Recovery Process

- Randomly select 5 sentences from the representative English sentences in the Harvard sentences\(^2\).

Input paragraph: The boy was there when the sun rose. A rod is used to catch pink salmon. The source of the huge river is the clear spring. Kick the ball straight and follow through. Help the woman get back to her feet.

Step 1 Searching results:

The boy/box was there when the sun rose. A *** is used to catch **** *****. The source of the huge river is the clear spring. **** the ball straight and follow through. Help the woman get back to her ****.

Step 2 Recovering words not in the dictionary:

(1) rod; (2) pink; (3) salmon; (4) Kick; (5) feet.

Eavesdropping Accuracy

Word recovery ratio = \( \frac{\text{\# of successfully recovered words}}{\text{total \# of input words}} \)

- Single article recovery (Type a piece of CNN news)
Impact of CSI Sample Classification Errors

- We artificially introduce errors into the groupings.

![Graph showing word recovery ratio against success rate of classification for 1500-word, 1000-word, and 500-word dictionaries.]
Overall Recovery Accuracy

- $L_{WRR>x}$ denotes the required number of typed words from each article to satisfy the ratio $x$. 

\begin{center}
\includegraphics[width=\textwidth]{chart.png}
\end{center}
Time Complexity Analysis

- The **comparison of relationship matrices** is the dominant part of the demodulation phase.
Password Entropy Reduction

• The higher the entropy, the more the randomness

• 2012 Yahoo! Voices hack[^3]: 342,508 passwords: 98.42% of passwords are 12 characters or fewer

Password Entropy Reduction (Cont’d)

• Breaking a 9-character password is reduced to guessing 1-5 non-letter characters.
Outline

• Motivation
• Attack Design
• Experiment Results
• Conclusion
Conclusion

- Identify a new type of keystroke eavesdropping attack bypassing the training requirement

- Create a joint demodulation algorithm to establish the mapping between a letter and a CSI sample

- Implement this attack on software-defined radio platforms and conduct a suite of experiments to validate its impact
Thank you!
Any questions?