Defeating Hidden Audio Channel Attacks on Voice Assistants via Audio-Induced Surface Vibrations

Presenter: Chen Wang

Chen Wang†, Abhishek Anand *, Jian Liu†, Payton Walker*, Yingying Chen†, Nitesh Saxena*

†WINLAB, Rutgers University, NJ, USA
* University of Alabama at Birmingham, AL, USA
Introduction

- Widely deployed voice controllable systems (VCS)
  - Convenient way of interaction
  - Integrated into many platforms

Mobile phones (e.g., Siri and Google Now)

- Fundamental vulnerabilities due to the propagation properties of sound

- Emerging hidden voice commands
  - Recognizable to VCS
  - Incomprehensible to humans
Hidden Voice Command

- Attacks the disparities of voice recognition between human and machine
- Iteratively shaping their audio features to meet the requirements:
  - Understandable to VCSs
  - Hard to be perceived by the users

**Attack model**
- Internal attack – embedded in media and played by the target device
- External attack – played via a loudspeaker in the proximity

![Diagram showing MFCC Feature Extraction, Adjusting MFCC parameters, Inverse MFCC, and speech recognition system.]

- Yes: Recognized by human attacker
- Yes: Recognized by the system
- No: Hidden voice command
- No: Recognized by the system

Examples:
- *browse evil.com*
- *call 911*
Related Work

Defend acoustic attacks based on audio information
  
  Voice authentication models
  
  Speech vocal features (e.g., )

Speaker liveness detection
  
  Restricted application scenarios by either requiring the microphone to be held close to mouth or additional dedicated hardware (e.g., on a wearable)

A multi-modality authentication framework is highly desirable to provide enhanced security:

Audio sending modality + vibration sensing modality

Only relying on speech audio features is vulnerable to hidden voice commands
Basic Idea

Basic Idea: utilizing the vibration signatures of the voice command to detect hidden voice commands

- Many VCS devices (e.g., smartphones and voice assistant systems) are already equipped with motion sensors
- Unique audio-induced surface vibrations captured by the motion sensor are hard to forge
- Two modes for capturing noticeable speech impact on motion sensors based on playback

![Diagram showing mobile device, HomePod, motion sensor, and speaker connections for front-end and back-end playback.]

12/13/2019
Capturing Voice Using Motion Sensors

- Shared surface between loudspeaker and microphone
- Low sampling rate motion sensors (e.g., < 200Hz)
- Nonlinear vibration responses
- Distinct vibration domain

\[ f_{\text{alias}} = |f - Nf_s|, \ N \in \mathbb{Z}. \]

Played Audio  
Vibration Responses

Lead to aliased vibration signals

“show facebook.com”
Why Vibration?

- Existing speech/voice recognition methods are based audio domain voice vocal features
- Hidden voice commands are designed to duplicate these audio domain features by iteratively modify a voice command
- Audio-induced surface vibrations
  - An additional sensing domain, distinct to audio
  - Hard to be forged from audio signals in software
  - Similar audio features result in distinct vibration features
  - Resulting vibration responses are device-dependent (device physical vibrations, motion sensors)

The vibration domain approach can work in conjunction with the audio domain approach to more effectively detect the hidden voice commands.
System Overview

1. Accelerometer Readings
2. Vibration Feature Derivation
   - Time/Frequency Domain
   - Statistical Features
3. Acoustic Features
   - (MFCC, Chroma Vector)
4. Vibration Noise Removal
5. Voice Command Segmentation
6. Data Calibration
7. Feature Normalization
8. Hidden Voice Command Detection
   - Supervised Learning-based Classifier
     - Simple Logistic
     - Random Tree
     - Random Forest
   - Unsupervised Learning-based Classifier
     - K-means
     - K-medoid

Frontend Playback

Backend Playback

Motion Sensor

Speaker

Replay Device in Cloud Service

Mobile Device or HomePod
Vibration Feature Derivation

- Unique and hard to forge vibration features
  - Statistical features in time and frequency domains
  - Deriving Acoustic Features from Motion Sensor Data
    - MFCC
    - Chrome vectors

- Nonlinear relationship between audio features and vibration features
Vibration Feature Derivation

- Unique and hard to forge vibration features
  - Statistical features in time and frequency domains
  - Deriving Acoustic Features from Motion Sensor Data
    - MFCC
    - Chrome vectors

- Nonlinear relationship between audio features and vibration features

- Feature Selection Based on Statistical Analysis

\[
s = \frac{\bar{F}_{\text{hid}} - \bar{F}_{\text{hum}}}{\max \left( \frac{\sqrt{\sum (F_{\text{hid}}(t) - \bar{F}_{\text{hid}})^2}}{n}, \frac{\sqrt{\sum (F_{\text{hum}}(t) - \bar{F}_{\text{hum}})^2}}{n} \right)}
\]
Feature Selection Based on Statistical Analysis
Hidden Voice Command Detection

- **Supervised Learning-based method**
  - Simple Logistic
  - Support Vector Machine
  - Random Forest
  - Random Tree

- **Unsupervised learning-based method**
  - $k$-means/$k$-medoids based methods
  - Calculating the Euclidean distance of the voice command samples to the cluster centroid
  - Not require much training
Experimental Setup

- Front-end playback setup
  - 4 different smartphones
  - On table
  - Held by hand
  - Placed on sofa

- Backend playback setup
  - Imitated cloud service device
  - Prototype on Raspberry Pi

- 10 voice commands, 5 speakers

- 13,000 vibration data traces
  - 6500 benign commands
  - 6500 hidden voice commands
## Performance Evaluation

### Supervised-learning

<table>
<thead>
<tr>
<th></th>
<th>Note 4</th>
<th>G3</th>
<th>Nexus 6</th>
<th>S6</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimpleLogistic</td>
<td>100%</td>
<td>99.8%</td>
<td>100%</td>
<td>88.3%</td>
</tr>
<tr>
<td>SMO</td>
<td>100%</td>
<td>99.9%</td>
<td>99.9%</td>
<td>85.4%</td>
</tr>
<tr>
<td>Random Forest</td>
<td>100%</td>
<td>99.5%</td>
<td>100%</td>
<td>93.1%</td>
</tr>
<tr>
<td>Random Tree</td>
<td>99.9%</td>
<td>98.1%</td>
<td>100%</td>
<td>87.4%</td>
</tr>
</tbody>
</table>

### Front-end playback setup

### Back-end playback setup

<table>
<thead>
<tr>
<th></th>
<th>Note 4</th>
<th>G3</th>
<th>Nexus 6</th>
<th>S6</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimpleLogistic</td>
<td>99.9%</td>
<td>99.8%</td>
<td>99.8%</td>
<td>95.3%</td>
</tr>
<tr>
<td>SMO</td>
<td>99.9%</td>
<td>99.9%</td>
<td>99.3%</td>
<td>95.0%</td>
</tr>
<tr>
<td>Random Forest</td>
<td>99.9%</td>
<td>100%</td>
<td>99.8%</td>
<td>95.3%</td>
</tr>
<tr>
<td>Random Tree</td>
<td>99.9%</td>
<td>98.9%</td>
<td>97.9%</td>
<td>89.7%</td>
</tr>
</tbody>
</table>
Performance Evaluation
Unsupervised-learning

Up to 99% accuracy for both frontend and backend setups to differentiate normal commands from hidden voice commands
Performance Evaluation

- Partial playback to reduce delay

### Front-end playback setup

<table>
<thead>
<tr>
<th></th>
<th>Note 4</th>
<th>G3</th>
<th>Nexus 6</th>
<th>S6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replay all</td>
<td>100%</td>
<td>99.10%</td>
<td>100%</td>
<td>85.70%</td>
</tr>
<tr>
<td>Replay 1s</td>
<td>100%</td>
<td>89.10%</td>
<td>99.90%</td>
<td>85.60%</td>
</tr>
<tr>
<td>Replay 0.5s</td>
<td>99.90%</td>
<td>85.20%</td>
<td>95.90%</td>
<td>85%</td>
</tr>
</tbody>
</table>

### Back-end playback setup

<table>
<thead>
<tr>
<th></th>
<th>Note 4</th>
<th>G3</th>
<th>Nexus 6</th>
<th>S6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replay all</td>
<td>99.90%</td>
<td>97.90%</td>
<td>93.40%</td>
<td>76%</td>
</tr>
<tr>
<td>Replay 1s</td>
<td>92.9</td>
<td>99.10%</td>
<td>92.40%</td>
<td>75.90%</td>
</tr>
<tr>
<td>Replay 0.5s</td>
<td>88.5</td>
<td>90.20%</td>
<td>90.50%</td>
<td>73.80%</td>
</tr>
</tbody>
</table>

- Various mobile device usage scenarios of frontend playback setup

<table>
<thead>
<tr>
<th></th>
<th>Table</th>
<th>Held in hand</th>
<th>Placed on sofa</th>
<th>80%vol. on table</th>
<th>2x speed on table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kmed</td>
<td>100%</td>
<td>87.30%</td>
<td>100%</td>
<td>100%</td>
<td>88.30%</td>
</tr>
<tr>
<td>Kmea</td>
<td>100%</td>
<td>87.30%</td>
<td>100%</td>
<td>100%</td>
<td>85.20%</td>
</tr>
</tbody>
</table>
Conclusion

- Show that hidden voice commands can be detected by their speech features in the vibration domain.
- Derive the unique vibration features (statistical features in the time and frequency domains and speech features to distinguish hidden voice commands from normal commands).
- Develop both supervised and unsupervised learning-based systems to detect hidden voice commands.
- Implemented the proposed system in two modes (i.e., frontend playback and backend playback).
- Extensive experiments show that the hidden voice commands can be detected based on their speech features in the vibration domain with high accuracy.