



Auditory Eyesight: Demystifying μ s-Precision Keystroke Tracking Attacks on Unconstrained Keyboard Inputs

Yazhou Tu, Liqun Shan, and Md Imran Hossen, *University of Louisiana at Lafayette*;
Sara Rampazzi and Kevin Butler, *University of Florida*;
Xiali Hei, *University of Louisiana at Lafayette*

<https://www.usenix.org/conference/usenixsecurity23/presentation/tu>

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 32nd USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 32nd USENIX Security Symposium.

August 9–11, 2023 • Anaheim, CA, USA

978-1-939133-37-3

Open access to the Artifact Appendices to the Proceedings of the 32nd USENIX Security Symposium is sponsored by USENIX.



USENIX'23 Artifact Appendix

Auditory Eyesight: Demystifying μ s-Precision Keystroke Tracking Attacks on Unconstrained Keyboard Inputs

Yazhou Tu, Liqun Shan, Md Imran Hossen, Sara Rampazzi[†], Kevin Butler[†], Xiali Hei
University of Louisiana at Lafayette
[†]University of Florida

A Artifact Appendix

A.1 Abstract

The artifact contains the dataset and benchmark results of tracking users' keystroke sounds to recover unconstrained keyboard inputs. It contains test cases with code to reproduce the attack results in different scenarios.

A.2 Description & Requirements

We run experiments on a computer with dual Xeon v3 E2683 processors and 32-GB RAM.

A.2.1 Security, privacy, and ethical concerns

None.

A.2.2 How to access

The artifact can be accessed via the following GitHub link
https://github.com/auditoryeye/auditoryeye_artifact/releases/tag/20230809

A.2.3 Hardware dependencies

We recommend using computers with at least 12-core processors and 32-GB RAM.

A.2.4 Software dependencies

The code was written and tested in Matlab R2020b. We recommend using R2020b or newer versions of Matlab with the parallel computing toolbox.

A.2.5 Benchmarks

None.

A.3 Set-up

We organize the contents in test cases. There are 11 folders in the repository. Each folder contains the data, benchmark results, and code. In each folder, there is a main.m file. The artifact can be evaluated after navigating to one of the folders and opening the main.m file in Matlab.

A.3.1 Installation

The artifact is ready to run after downloading the contents from its GitHub repository:

https://github.com/auditoryeye/auditoryeye_artifact/tree/20230809

A.3.2 Basic Test

Navigate to the 01_proofofconcept_multiround_apple_keys folder. Open main.m file in Matlab and run each line in order. The results will be saved to the 01_proofofconcept_multiround_apple_keys/recording01_keys_interpolated folder. The results will be illustrated in the prompted windows. Step-by-step instructions to start the parallel pool and run the basic test are available at the artifact's GitHub repository.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): *Auditory Eyesight can distinguish compactly spaced keys by localizing keystroke sound signals in the differential range of microseconds.*

This is proven by: 1) experiments (E1, E2) described in paper's Section 4 whose results are illustrated in the paper's Figure 6, Figure 7, Figure 8, Figure 10, Figure 12, Figure 26, Figure 27, Figure 28, Figure 29, Figure 30, Figure 31, Table 1, and Table 2;

2) experiments (E9, E10) described in paper's Section 7 whose results are illustrated in the paper's Figure 22 and Figure 23.

(C2): Auditory Eyesight can reveal unconstrained user inputs. This is proven by the experiment (E3) described in the paper’s Section 5, whose results are summarized in the paper’s Table 3.

(C3): Auditory Eyesight works with different angles, distances, and certain non-line-of-sight scenarios. This is proven by: 1) experiments (E4, E5, E6, E7, E8) described in paper’s Section 7 whose results are summarized in the paper’s Table 6.

2) experiments (E9, E10) described in paper’s Section 7 whose results are illustrated in the paper’s Figure 22 and Figure 23.

3) experiment (E11) described in paper’s Section 6.2 whose results are summarized in the paper’s Figure 21 and Table 5.

A.4.2 Experiments

(E1): [30 human-minutes + 10 compute-minutes + 2GB disk]: Proof-of-concept attacks to localize 598 keystrokes on the Apple Magic keyboard from a 0.5-m attack distance.

How to:

1. Navigate to the `01_proofofconcept_multiround_apple_keys` folder.
2. Read the `01_readme.txt` file.
3. Open `main.m` file in Matlab and run each line in order to calculate the results. The results will be saved into the `01_proofofconcept_multiround_apple_keys/recording01_keys_interpolated` folder.
4. The results will be illustrated in the prompted windows and the console.

Preparation:

Start the Parallel Pools in Matlab.

Execution:

Run each line in the `main.m` file.

Results:

We describe the results and the corresponding contents as follows:

- **E1.01**

Command:

```
run('YZProcessing05_statistics_remoutlier.m')
```

Reproduced Result:

Paper’s Figure 6 top

- **E1.02**

Command:

```
run('Statistics101_round1.m');
```

Reproduced Result:

Paper’s Figure 7

- **E1.03**

Command:

```
run('Statistics201_figure_2d_1round.m');
```

Reproduced Result:

Paper’s Figure 26

- **E1.04**

Command:

```
run('YZProcessing07_2ndroundstatistics.m')
```

Reproduced Result:

Paper’s Figure 6 middle

- **E1.05**

Command:

```
run('Statistics101_round2.m');
```

Reproduced Result:

Paper’s Figure 8

- **E1.06**

Command:

```
run('Statistics201_figure_2d_2round.m');
```

Reproduced Result:

Paper’s Figure 27

- **E1.07**

Command:

```
run('YZProcessing09_3rdroundstatistics')
```

Reproduced Result:

Paper’s Figure 6 bottom

- **E1.08**

Command:

```
run('Statistics101_round3.m');
```

Reproduced Result:

Paper’s Figure 10

- **E1.09**

Command:

```
run('Statistics101_round5');
```

Reproduced Result:

Paper’s Table 2

- **E1.10**

Command:

```
run('Statistics201_figure_2d_5round');
```

Reproduced Result:

Paper’s Figure 28

- **E1.11**

Command:

```
run('Statistics_accuracy_calculation');
```

Reproduced Result:

Paper’s Table 1

(E2): [30 human-minutes + 10 compute-minutes + 2GB

disk]: Proof-of-concept attacks to localize 595 keystrokes on the Razor keyboard from a 0.5-m attack distance.

How to:

1. Navigate to the *02_proofofconcept_multiround_razor_keys* folder.
2. Read the *01_readme.txt* file.
3. Open *main.m* file in Matlab and run each line in order to calculate the results. The results will be saved into the *02_proofofconcept_multiround_razor_keys/recording01_keys_interpolated* folder.
4. The results will be illustrated in the prompted windows and the console.

Preparation:

Start the Parallel Pools in Matlab.

Execution:

Run each line in the main.m file.

Results:

We describe the results and the corresponding contents as follows:

• **E2.01**

Command:

`run('YZProcessing05_statistics_reoutlier.m')`

Reproduced Result:

Paper's Figure 12 top

• **E2.02**

Command:

`run('Statistics201_figure_2d_1round.m');`

Reproduced Result:

Paper's Figure 29

• **E2.03**

Command:

`run('YZProcessing07_2ndroundstatistics.m')`

Reproduced Result:

Paper's Figure 12 middle

• **E2.04**

Command:

`run('Statistics201_figure_2d_2round.m');`

Reproduced Result:

Paper's Figure 30

• **E2.05**

Command:

`run('YZProcessing09_3rdroundstatistics')`

Reproduced Result:

Paper's Figure 12 bottom

• **E2.06**

Command:

`run('Statistics201_figure_2d_5round');`

Reproduced Result:

Paper's Figure 31

• **E2.07**

Command:

`run('Statistics101_round5');`

Reproduced Result:

Paper's Table 2

• **E2.08**

Command:

`run('Statistics_accuracy_calculation');`

Reproduced Result:

Paper's Table 1

(E3): [30 human-minutes + 20 compute-minutes + 10GB disk]: Attacks on unconstrained user inputs

How to:

1. Navigate to the *03_userstudy_1* folder.
2. Read the *01_readme.txt* file.
3. Open *main.m* file in Matlab and run each line in order to calculate the results. The results will be saved into the *02_proofofconcept_multiround_razor_keys/recording01_keys_interpolated* folder.
4. The results will be illustrated in the prompted windows and the console.

Preparation:

Start the Parallel Pools in Matlab.

Execution:

Run main.m file.

Results:

We describe the results and the corresponding contents as follows:

• **E3.1**

Command:

`run('Statistics_accuracy_calculation');`

Reproduced Result:

Paper's Table 3

(E4): [15 human-minutes + 5 compute-minutes + 2GB disk]: Test case with different angle

How to:

1. Navigate to the *04_additiontestcase_angle01* folder.
2. Read the *01_readme.txt* file.
3. Open *main.m* file in Matlab and run each line in order to calculate the results. The results will be saved into the *04_additiontestcase_angle01/recording01_keys_interpolated* folder.
4. The results will be illustrated in the prompted windows and the console.

Preparation:

Start the Parallel Pools in Matlab.

Execution:

Run *main.m* file.

Results:

We describe the results and the corresponding contents as follows:

- **E4.1**

Command:

```
run('Statistics_accuracy_calculation');
```

Reproduced Result:

Paper's Table 6, Test Case 1

(E5): [15 human-minutes + 5 compute-minutes + 2GB disk]:
2nd test case with different angle

How to:

1. Navigate to the *04_additiontestcase_angle02* folder.
2. Read the *01_readme.txt* file.
3. Open *main.m* file in Matlab and run each line in order to calculate the results. The results will be saved into the *04_additiontestcase_angle02/recording01_keys_interpolated* folder.
4. The results will be illustrated in the prompted windows and the console.

Preparation:

Start the Parallel Pools in Matlab.

Execution:

Run *main.m* file.

Results:

We describe the results and the corresponding contents as follows:

- **E5.1**

Command:

```
run('Statistics_accuracy_calculation');
```

Reproduced Result:

Paper's Table 6, Test Case 2

(E6): [15 human-minutes + 5 compute-minutes + 2GB disk]:
3rd test case with different angle

How to:

1. Navigate to the *04_additiontestcase_angle03* folder.
2. Read the *01_readme.txt* file.
3. Open *main.m* file in Matlab and run each line in order to calculate the results. The results will be saved into the *04_additiontestcase_angle03/recording01_keys_interpolated* folder.
4. The results will be illustrated in the prompted windows and the console.

Preparation:

Start the Parallel Pools in Matlab.

Execution:

Run *main.m* file.

Results:

We describe the results and the corresponding contents as follows:

- **E6.1**

Command:

```
run('Statistics_accuracy_calculation');
```

Reproduced Result:

Paper's Table 6, Test Case 3

(E7): [15 human-minutes + 5 compute-minutes + 2GB disk]:
4th test case with different angle and 3 microphones

How to:

1. Navigate to the *04_additiontestcase_angle04_3mics* folder.
2. Read the *01_readme.txt* file.
3. Open *main.m* file in Matlab and run each line in order to calculate the results. The results will be saved into the *04_additiontestcase_angle04_3mics/recording01_keys_interpolated* folder.
4. The results will be illustrated in the prompted windows and the console.

Preparation:

Start the Parallel Pools in Matlab.

Execution:

Run *main.m* file.

Results:

We describe the results and the corresponding contents as follows:

- **E7.1**

Command:

```
run('Statistics_accuracy_calculation');
```

Reproduced Result:

Paper's Table 6, Test Case 4

(E8): [15 human-minutes + 5 compute-minutes + 2GB disk]:
5th test case with different angle and 3 microphones

How to:

1. Navigate to the *04_additiontestcase_angle05_3mics* folder.
2. Read the *01_readme.txt* file.
3. Open *main.m* file in Matlab and run each line in order to calculate the results. The results will be saved into the *04_additiontestcase_angle05_3mics/recording01_keys_interpolated* folder.
4. The results will be illustrated in the prompted windows and the console.

Preparation:

Start the Parallel Pools in Matlab.

Execution:

Run *main.m* file.

Results:

We describe the results and the corresponding contents as follows:

- **E8.1**

Command:

```
run('Statistics_accuracy_calculation');
```

Reproduced Result:

Paper's Table 6, Test Case 5

(E9): [15 human-minutes + 5 compute-minutes + 2GB disk]:
test case with 1-m distance

How to:

1. Navigate to the 04_additiontestcase_distance1m folder.
2. Read the 01_readme.txt file.
3. Open main.m file in Matlab and run each line in order to calculate the results. The results will be saved into the 04_additiontestcase_distance1m/recording01_keys_interpolated folder.
4. The results will be illustrated in the prompted windows and the console.

Preparation:

Start the Parallel Pools in Matlab.

Execution:

Run main.m file.

Results:

We describe the results and the corresponding contents as follows:

- **E9.1**

Command:

```
run('Statistics101_round3.m');
```

Reproduced Result:

Paper's Figure 22 top

(E10): [15 human-minutes + 5 compute-minutes + 2GB disk]: test case with 2-m distance

How to:

1. Navigate to the 04_additiontestcase_distance2m folder.
2. Read the 01_readme.txt file.
3. Open main.m file in Matlab and run each line in order to calculate the results. The results will be saved into the 04_additiontestcase_distance2m/recording01_keys_interpolated folder.
4. The results will be illustrated in the prompted windows and the console.

Preparation:

Start the Parallel Pools in Matlab.

Execution:

Run main.m file.

Results:

We describe the results and the corresponding contents as follows:

- **E10.1**

Command:

```
run('YZProcessing05_statistics_remoutlier');
```

Reproduced Result:

Paper's Figure 23 top

- **E10.2**

Command:

```
run('YZProcessing07_2ndroundstatistics');
```

Reproduced Result:

Paper's Figure 23 middle

- **E10.3**

Command:

```
run('YZProcessing09_3rdroundstatistics');
```

Reproduced Result:

Paper's Figure 23 bottom

- **E10.4**

Command:

```
run('Statistics101_round3.m');
```

Reproduced Result:

Paper's Figure 22 bottom

(E11): [20 human-minutes + 5 compute-minutes + 2GB disk]: non-line-of-sight test case using a laptop

How to:

1. Navigate to the 04_additiontestcase_nloslaptop folder.
2. Read the 01_readme.txt file.
3. Open main.m file in Matlab and run each line in order to calculate the results. The results will be saved into the 04_additiontestcase_nloslaptop/recording01_keys_interpolated folder.
4. The results will be illustrated in the prompted windows and the console.

Preparation:

Start the Parallel Pools in Matlab.

Execution:

Run main.m file.

Results:

We describe the results and the corresponding contents as follows:

- **E11.1**

Command:

```
run('YZProcessing05_statistics_remoutlier');
```

Reproduced Result:

Paper's Figure 21 top

- **E11.2**

Command:

```
run('YZProcessing07_2ndroundstatistics');
```

Reproduced Result:

Paper's Figure 21 middle

- **E11.3**

Command:

```
run('YZProcessing13_5throunstatistics.m');
```

Reproduced Result:

Paper's Figure 21 bottom

- **E11.4**

Command:

```
run('Statistics_accuracy_calculation');
```

Reproduced Result:

Paper's Table 5

A.5 Notes on Reusability

There is a lack of an existing reference study of acoustic side-channel keystroke attacks with publicly available datasets. To address this issue, we publish the dataset, benchmark results, and the software of Auditory Eyesight to reproduce the results.

This artifact is suitable for various research purposes. The dataset can be used to benchmark different acoustic-channel keyboard attack methods in the future. Future works can investigate integrating additional signal processing or other extracted features to improve the attack performance.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20220926. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at <https://secartifacts.github.io/usenixsec2023/>.