A Metric for the Evaluation and Comparison of Keylogger Performance

Tobias Fiebig, Janis Danisevskis and Marta Piekarska

Technische Universität Berlin, FG Security in Telecommunications

CSET 2014, San Diego, 18th of August 2014





Introduction

This talk presents our work on "A Metric for the Evaluation and Comparison of Keylogger Performance":

Introduction

This talk presents our work on "A Metric for the Evaluation and Comparison of Keylogger Performance":

- Why we did it.

Introduction

This talk presents our work on "A Metric for the Evaluation and Comparison of Keylogger Performance":

- Why we did it.
- How we did it.

Introduction

This talk presents our work on "A Metric for the Evaluation and Comparison of Keylogger Performance":

- Why we did it.
- How we did it.
- And how *you* can contribute.

Motivation

- So in the beginning... we had this new vector for a kind of heuristic, not 100% precise keylogger...

Motivation

- So in the beginning... we had this new vector for a kind of heuristic, not 100% precise keylogger...
- ...and wanted to compare it's performance with other approaches.

Motivation

- So in the beginning... we had this new vector for a kind of heuristic, not 100% precise keylogger...
- ...and wanted to compare it's performance with other approaches.
- However, neither were a lot of these other keyloggers available as source, nor were the performance metrics applicable and/or portable.

Motivation

- So in the beginning... we had this new vector for a kind of heuristic, not 100% precise keylogger...
- ...and wanted to compare it's performance with other approaches.
- However, neither were a lot of these other keyloggers available as source, nor were the performance metrics applicable and/or portable.
- Solution: Build our own.

Contribution

- A method to evaluate a keylogger's performance.

Contribution

- A method to evaluate a keylogger's performance.
- Thorough description of how to apply this method.

Contribution

- A method to evaluate a keylogger's performance.
- Thorough description of how to apply this method.
- Demo code. :-)

Contribution

- A method to evaluate a keylogger's performance.
- Thorough description of how to apply this method.
- Demo code. :-)
- Hopefully starting a discussion on reproduceability and comparability in scientific publications.

Design

- Found an issue in a smartphone GPU library.

Design

- Found an issue in a smartphone GPU library.
- Can read bitmaps attached to the GPU as unprivileged process.

Design

- Found an issue in a smartphone GPU library.
- Can read bitmaps attached to the GPU as unprivileged process.

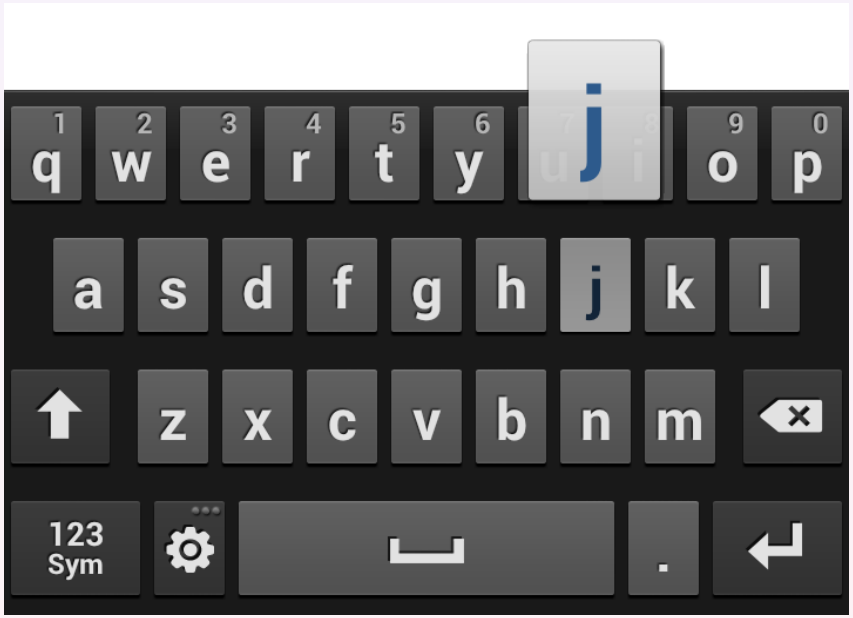


Figure: Tooltips are such bitmaps.

Implementation

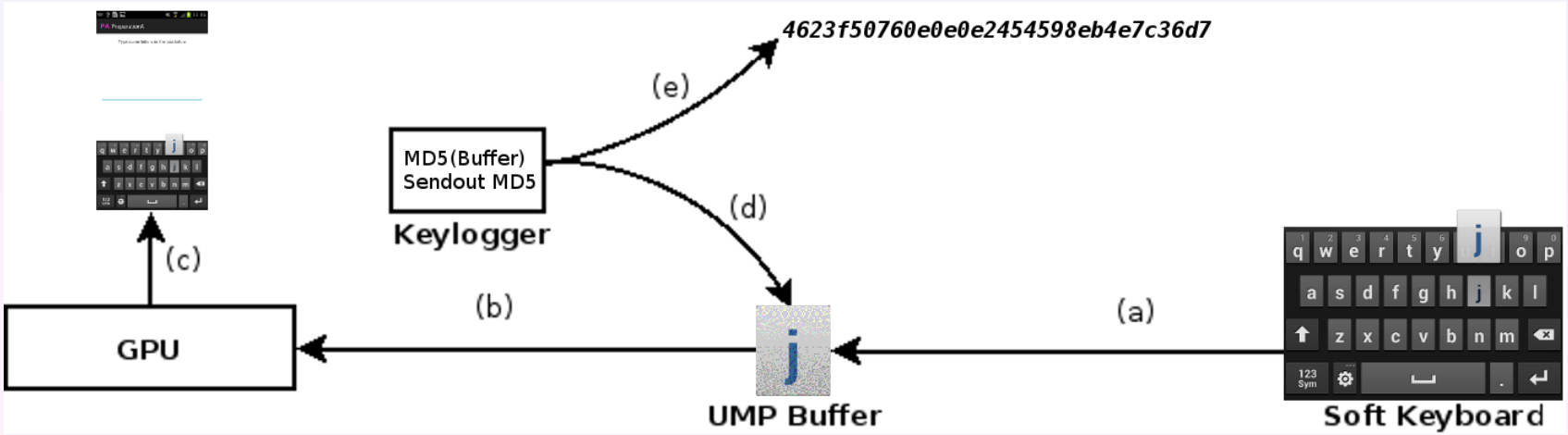


Figure: How the keylogger works.

Previous Work

- Before heuristic keyloggers: We got out what we put in.

Previous Work

- Before heuristic keyloggers: We got out what we put in.
- How long it takes to recover passwords [Lin et al., 2014].

Previous Work

- Before heuristic keyloggers: We got out what we put in.
- How long it takes to recover passwords [Lin et al., 2014].
- Probability based metric, mostly cornered around this approach; omitting checking what the users actually typed [Simon and Anderson, 2013].

Previous Work

- Before heuristic keyloggers: We got out what we put in.
- How long it takes to recover passwords [Lin et al., 2014].
- Probability based metric, mostly cornered around this approach; omitting checking what the users actually typed [Simon and Anderson, 2013].
- Using metrics designed for a different purpose [Xu et al., 2013].

Per input length inverted DLD

- Use Damerau-Levenshtein distance [Damerau, 1964, Levenshtein, 1966] as basis.

Per input length inverted DLD

- Use Damerau-Levenshtein distance [Damerau, 1964, Levenshtein, 1966] as basis.
- Counts the number of operations (insert, delete, substitute and transpose) needed to change $string_1$ to $string_2$.

Per input length inverted DLD

- Use Damerau-Levenshtein distance [Damerau, 1964, Levenshtein, 1966] as basis.
- Counts the number of operations (insert, delete, substitute and transpose) needed to change $string_1$ to $string_2$.
- Simplified: $dl(str_1, str_2) = \sum o_d + \sum o_i + \sum o_s + \sum o_t$

Challenges with this metric.

Issue: Gets naturally higher, the longer the input is.

Challenges with this metric.

Issue: Gets naturally higher, the longer the input is.

Solution: Divide by the length of one (defined) input string:

$$\frac{dl(real, captured)}{|characters(captured)|}$$

Challenges with this metric.

Issue: Gets naturally higher, the longer the input is.

Solution: Divide by the length of one (defined) input string:

$$\frac{dl(real, captured)}{|characters(captured)|}$$

Issue: Now lower means better. This is counter intuitive.

Challenges with this metric.

Issue: Gets naturally higher, the longer the input is.

Solution: Divide by the length of one (defined) input string:

$$\frac{dl(real,captured)}{|characters(captured)|}$$

Issue: Now lower means better. This is counter intuitive.

Solution: Subtract from 1, higher becomes better:

$$1 - \frac{dl(real,captured)}{|characters(captured)|}$$

Experimental Setting - Requirements and Solutions.

- Reproduceability. Others should be able to do the same things we did, ideally with their own keylogger.

⇒ Thoroughly document what and why we do it; share our code.



Experimental Setting - Requirements and Solutions.

- Comparability. When others do what we did, they should be able to compare their results to our's.

$\Rightarrow$ Test if this is actually the case by comparing two keyloggers with each other.



Figure: "Apple and Orange - they do not compare" by Michael Johnson

Experimental Setting - Requirements and Solutions.

- Reliability. We do not want external effects influencing our study, e.g. users miss-typing things.

$\Rightarrow$ We record what each user types.



Experimental Setting - Requirements and Solutions.

- Applicability. A keylogger study using a few hundred A's as input will probably have rather good results.

$\Rightarrow$ Create a text including passwords and "stuff people normally write".



Figure: Justin Baeder: https://www.flickr.com/photos/justinbaeder/5317820857/

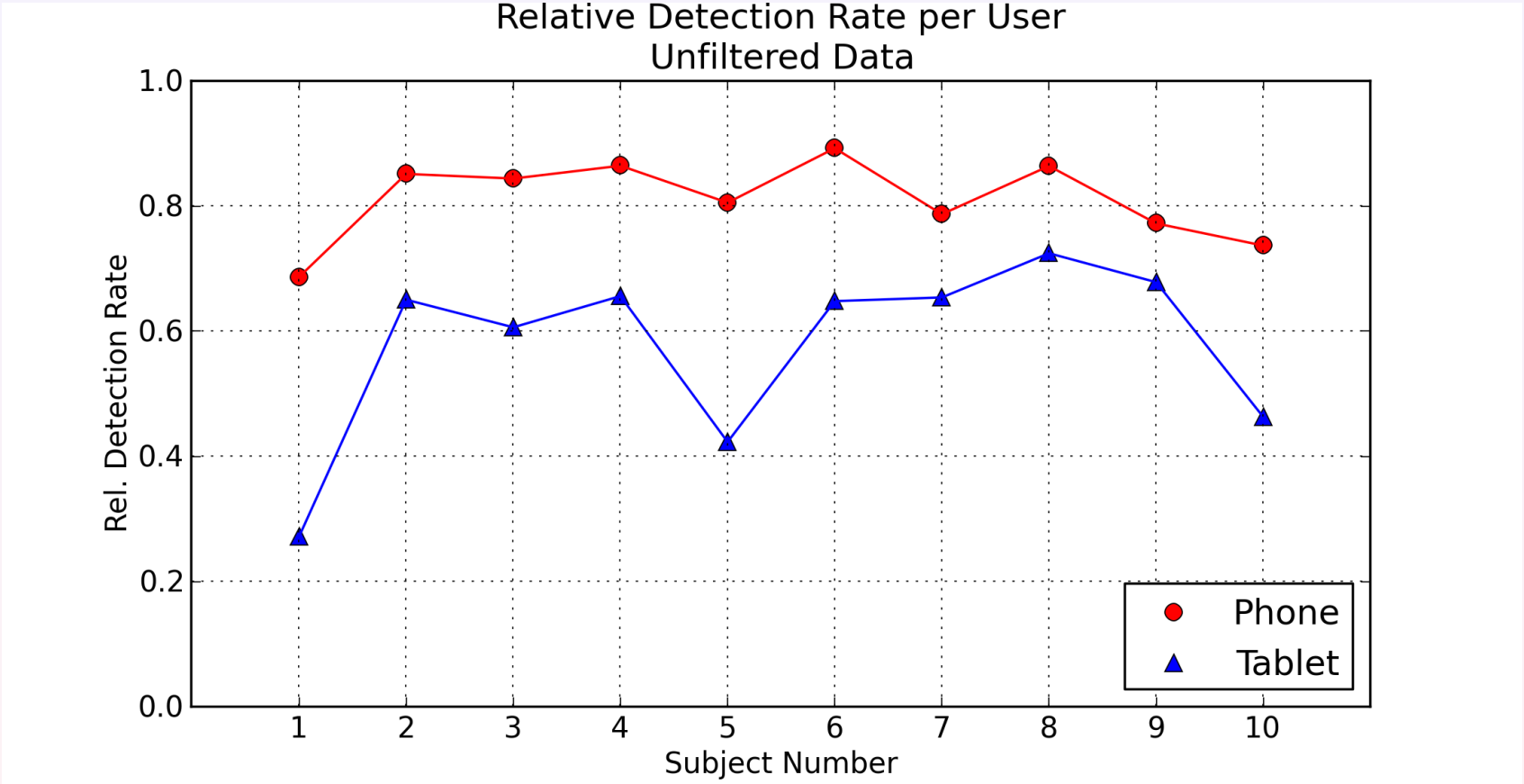
Tablet vs. Phone

As mentioned earlier, getting one's hands on a keylogger is kind of hard. So we simulate two distinct keyloggers:

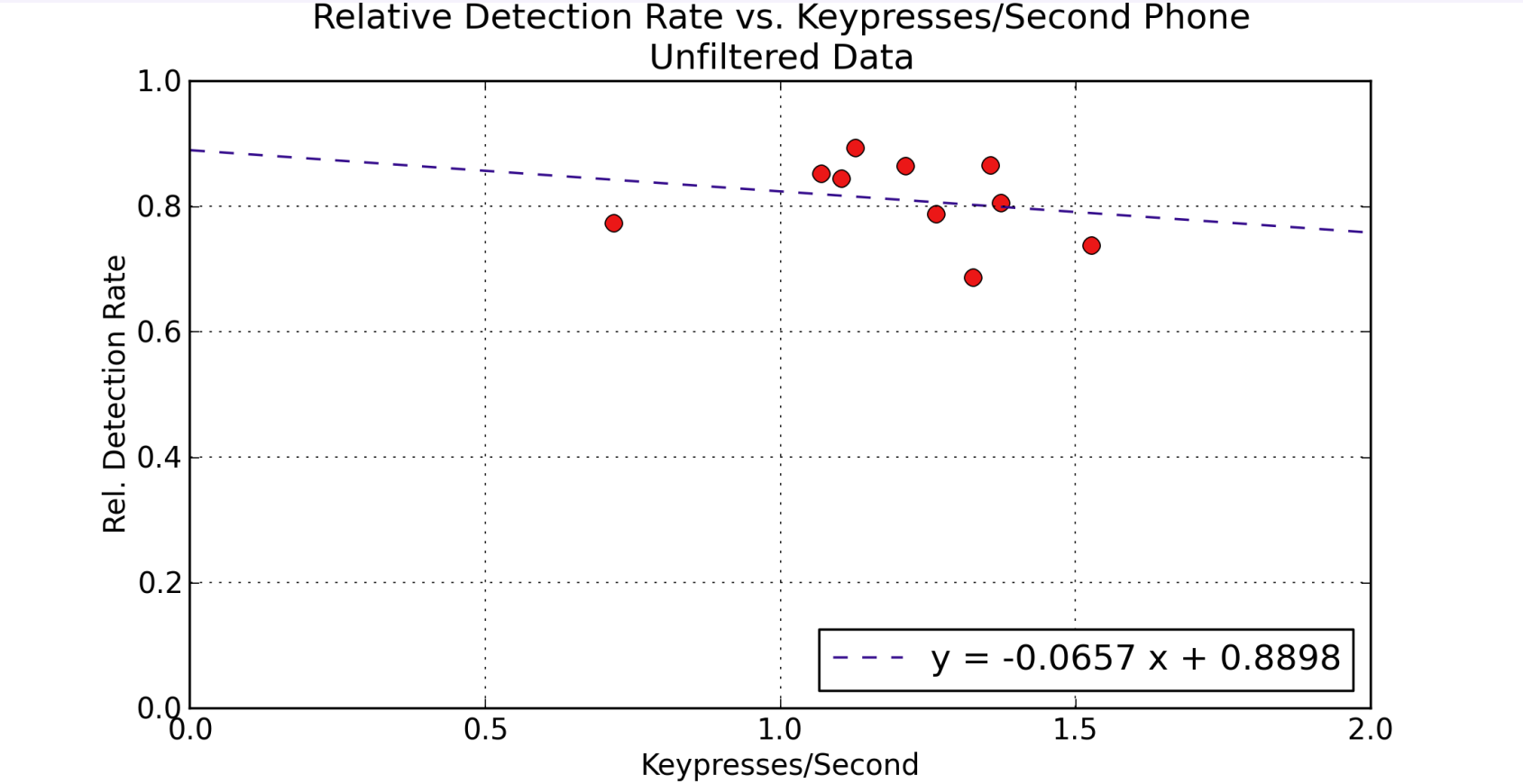
- Subjects exhibit a different typing speed due to the keyboard size.
- Implementation slightly differs between devices.
- The tablet has more CPU power.

Device	Android Version	Screen Size	PPI
Phone	4.0.4	4.8"	306ppi
Tablet	4.1.2	10.1"	149ppi

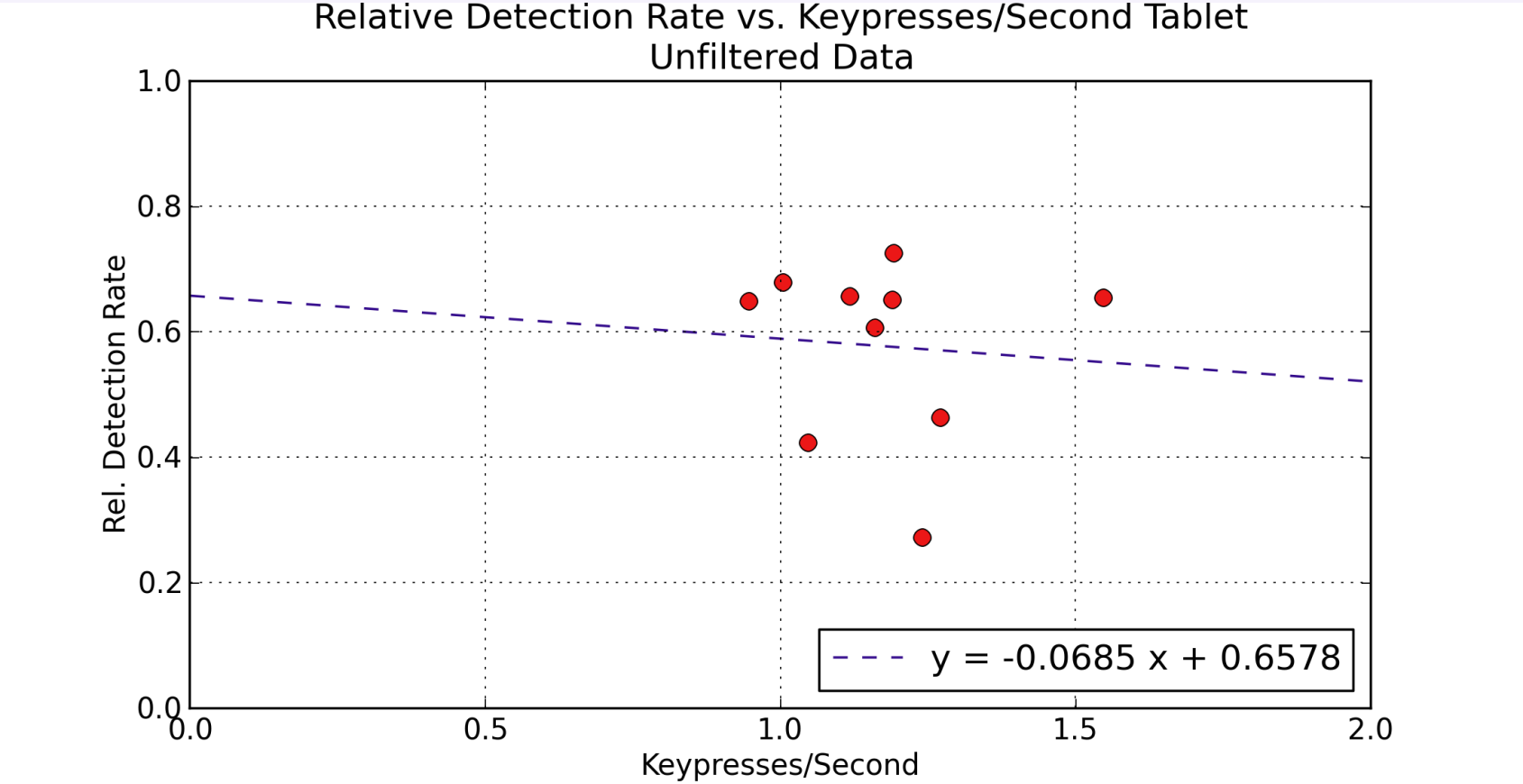
Results - Overview



Results - Typing speed Phone



Results - Typing speed Tablet



Results - What we learned.

- Apparently the keylogger works better on the phone than on the tablet.

Results - What we learned.

- Apparently the keylogger works better on the phone than on the tablet.
- The curves look like there is a connection between typing speed and detection rate, but the data does not support this.

Results - What we learned.

Recorded:	Thhisiisademotexxt.
Actual:	This is a demo text.

Filtering

- The analysis indicated some bias for the tablet implementation.

Filtering

- The analysis indicated some bias for the tablet implementation.
- Closer investigation revealed: Due to the stronger CPU the keylogger oversamples. Furthermore the space key is not recorded.

Filtering

- The analysis indicated some bias for the tablet implementation.
- Closer investigation revealed: Due to the stronger CPU the keylogger oversamples. Furthermore the space key is not recorded.
- Hence we filter the data. That is, we clean it of those effects introduced by the detected bias.

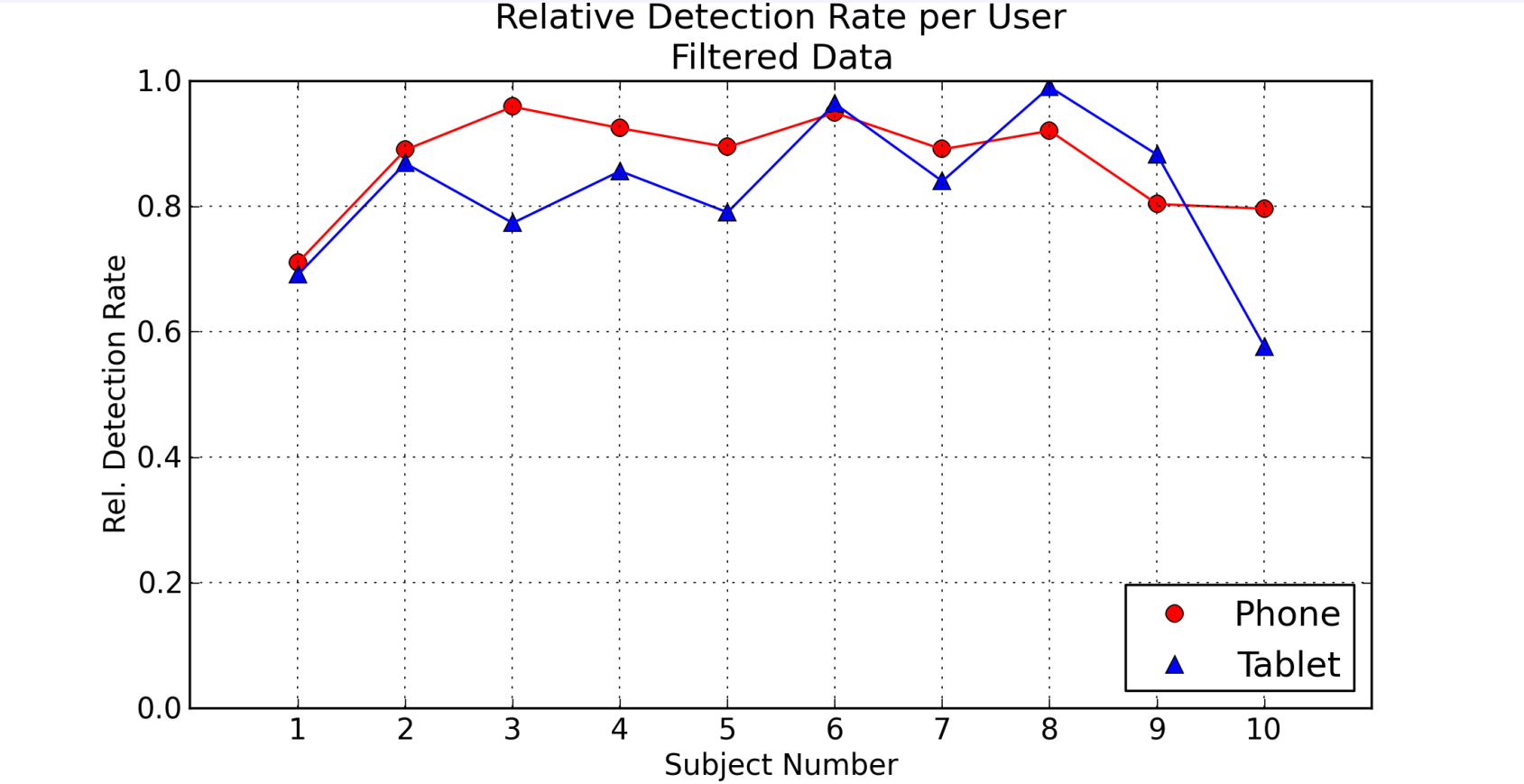
Filtering

- The analysis indicated some bias for the tablet implementation.
- Closer investigation revealed: Due to the stronger CPU the keylogger oversamples. Furthermore the space key is not recorded.
- Hence we filter the data. That is, we clean it of those effects introduced by the detected bias.
- In the case at hand this means the removal of all repeated characters and spaces.

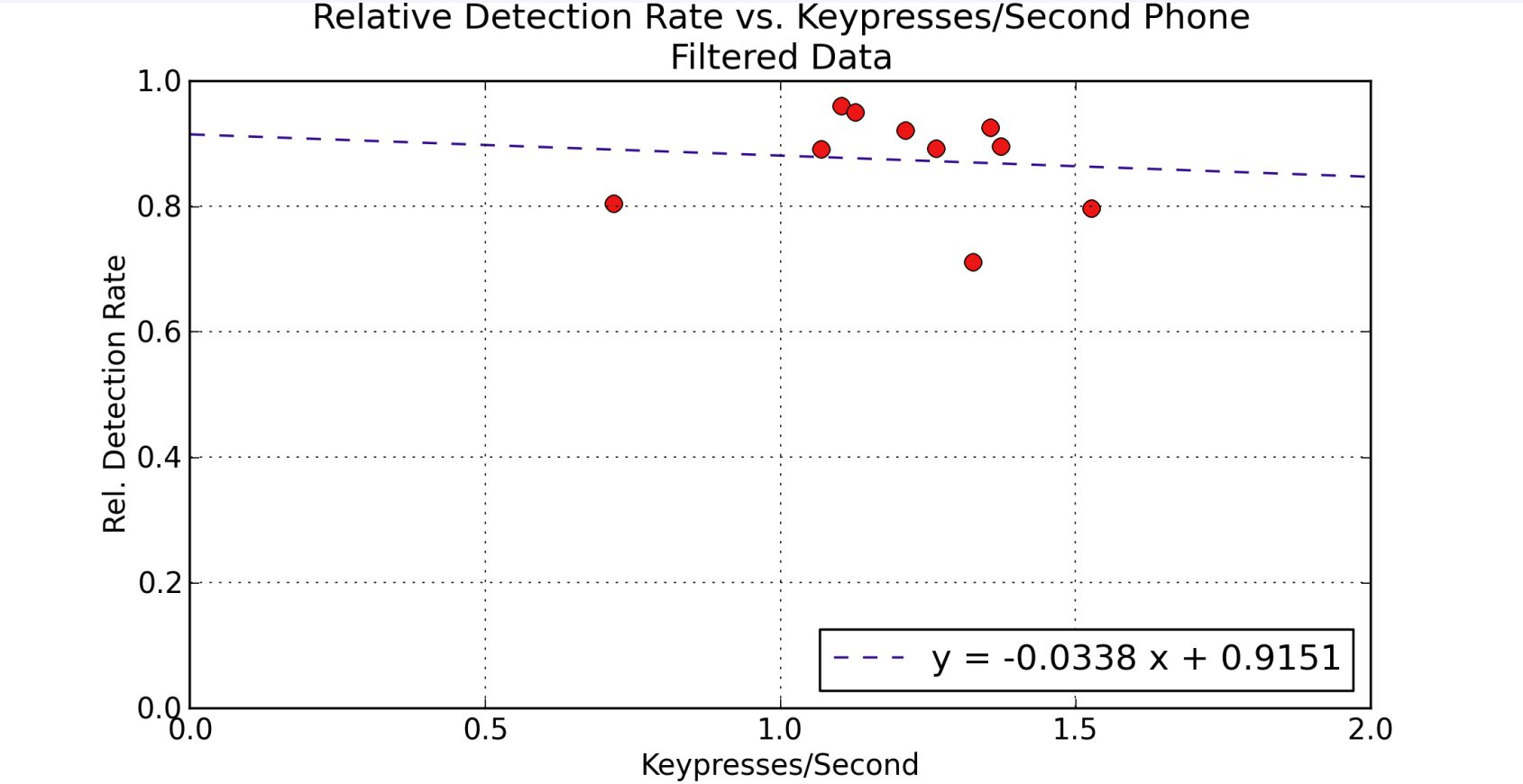
Filtering - Applied

Unfiltered:	Thhisiisademotexxt.
Actual:	This is a demo text.
Filtered:	Thisisademotext.
Actual Filtered:	Thisisademotext.

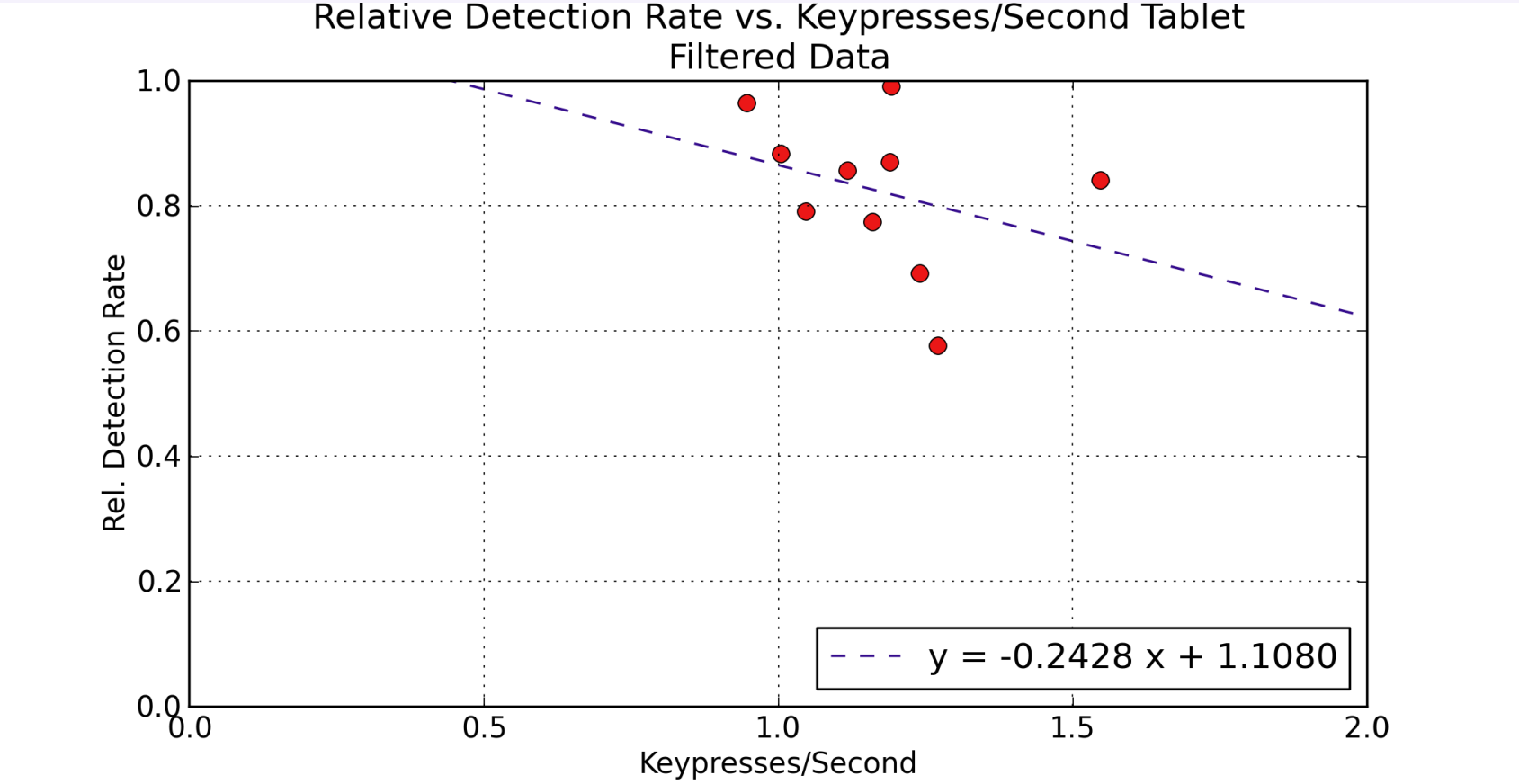
Results Filtered - Overview



Results Filtered - Typing speed Phone



Results Filtered - Typing speed Tablet



Results Filtered- What we learned.

- The keylogger on the tablet seems to be correlated to the user's typing speed.

Results Filtered- What we learned.

- The keylogger on the tablet seems to be correlated to the user's typing speed.
- The good performance on the filtered dataset indicates, that the identified issues are actually the cause of the overall worse performance.

On comparing Keyloggers...

- We could successfully compare two keylogger implementations.

On comparing Keyloggers...

- We could successfully compare two keylogger implementations.
- The recorded data allowed us to identify weak-spots in one implementation.

On comparing Keyloggers...

- We could successfully compare two keylogger implementations.
- The recorded data allowed us to identify weak-spots in one implementation.
- With this data we could also show that the implemented keylogger performs reasonably well.

Points for further discussion...

- There are many open challenges:

Points for further discussion...

- There are many open challenges:
  - Semantic Properties should be incorporated into the metric.

Points for further discussion...

- There are many open challenges:
  - Semantic Properties should be incorporated into the metric.
  - In the same manner the performance footprint of a keylogger should be incorporated as well.

Points for further discussion...

- There are many open challenges:
  - Semantic Properties should be incorporated into the metric.
  - In the same manner the performance footprint of a keylogger should be incorporated as well.
  - A database with evaluations for various presented keyloggers should be created for benchmarking purposes.
  - The keylogger-testing-app should be made more plug-able.

Points for further discussion...

- There are many open challenges:
  - Semantic Properties should be incorporated into the metric.
  - In the same manner the performance footprint of a keylogger should be incorporated as well.
  - A database with evaluations for various presented keyloggers should be created for benchmarking purposes.
  - The keylogger-testing-app should be made more plug-able.
- Relevant code and data analysis tools are publicly available at https://gitlab.sec.t-labs.tu-berlin.de/mobile-keylogger/metric

Bibliography I


Damerau, F. J. (1964).
A technique for computer detection and correction of spelling errors.
Communications of the ACM, 7(3):171--176.


Levenshtein, V. I. (1966).
Binary codes capable of correcting deletions, insertions and reversals.
In *Soviet physics doklady*, volume 10, page 707.


Lin, C.-C., Li, H., Zhou, X., and Wang, X. (2014).
Screenmilker: How to milk your android screen for secrets.
In *Proceedings of 21th USENIX Network Distributed System Security Symposium*. USENIX Association.


Simon, L. and Anderson, R. (2013).
Pin skimmer: Inferring pins through the camera and microphone.
In *Proceedings of the 3rd ACM workshop on Security and privacy in smartphones and mobile devices*. ACM.

Bibliography II



Xu, Y., Heinly, J., White, A. M., Monrose, F., and Frahm, J.-M. (2013).
Seeing double: reconstructing obscured typed input from repeated compromising reflections.
In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1063--1074. ACM.

Demo Text

This is a demo text. The demo text includes 23 passwords. Like this one 4%23$awD or this one 98%=!awdD. Just kidding. It is only two passwords. Is it not? Please add a new line with something you would consider a secure password. No need to remember it.