Randomized Aggregatable Privacy-Preserving Ordinal Response (RAPPOR)

Crowdsourcing statistics from end-user client software with strong privacy guarantees

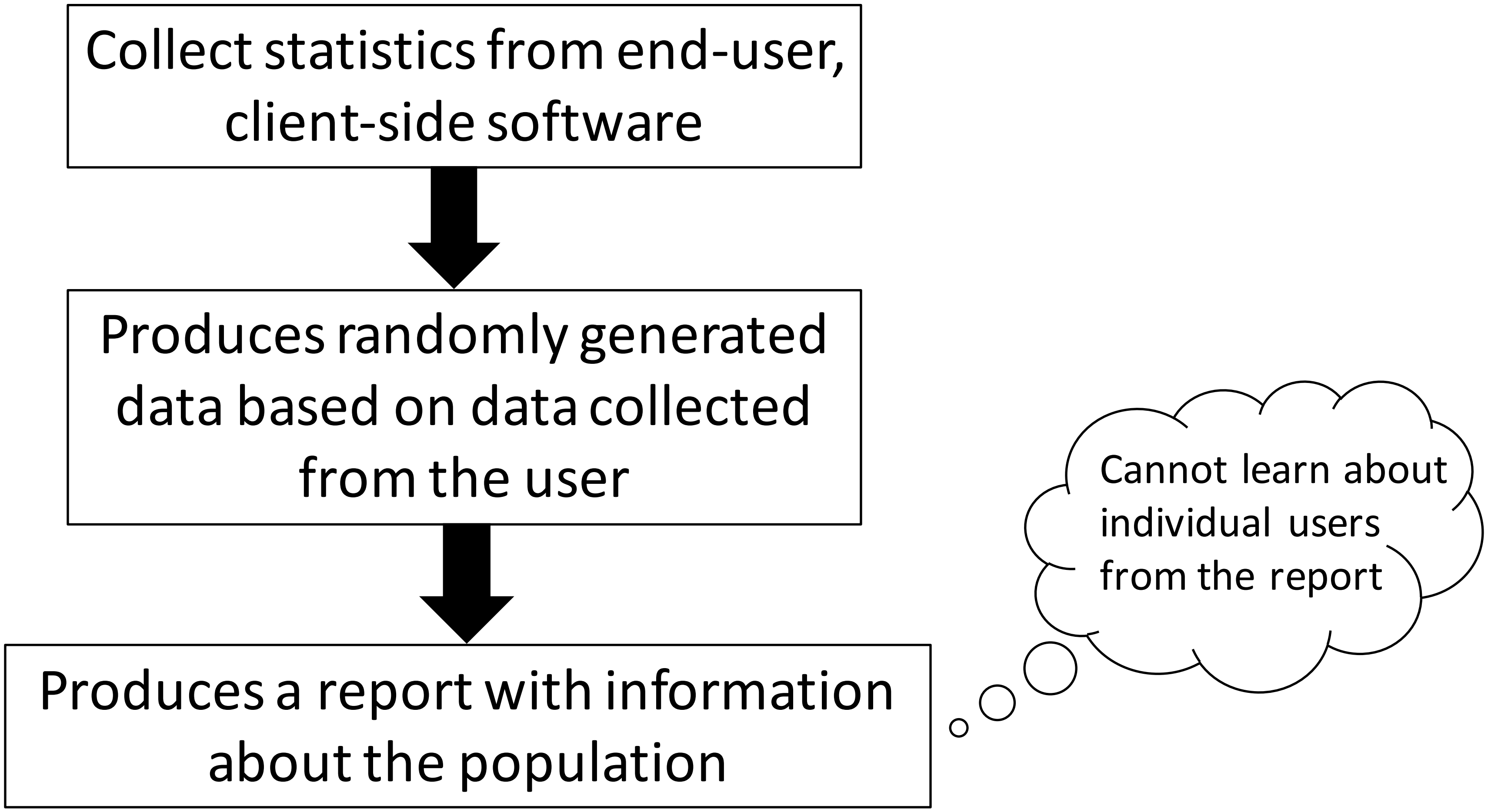
Motivating Idea:
Surveying people on sensitive topics

"Are you a member of the Communist party?"

- Surveyor: Asks participant to flip a fair coin in secret.
  - If Heads: Answer "yes"
  - If Tails: Tell the truth
- Participant retains strong deniability for "Yes" answers
  - Likely due to coin coming up heads

Goal: Preserve privacy of survey participants while still collecting accurate statistics

What is RAPPOR?


Collect statistics from end-user, client-side software
Produces randomly generated data based on data collected from the user
Produces a report with information about the population
Cannot learn about individual users from the report

Who might use RAPPOR?

- Cloud service operators
  - Need current statistics about their users' activity and client-side software
  - Observe how often some software features are used
    - Measure performance and failure
  - Better security and abuse protection for users

- Users of RAPPOR
  - Google's Chrome Web browser

Crowdsourcing statistics with RAPPOR

- RAPPOR responses (from clients) are bit strings
  - Each bit corresponds to a randomized response for one of the client’s properties

1	**1**	**1**	**0**
Male	communist	Between 18-35 yrs old	Between 36-50 yrs old

0	**0**	**0**	**1**
Female	~~communist~~	Between 18-35 yrs old	Between 36-50 yrs old

Crowdsourcing statistics with RAPPOR

- RAPPOR responses (from clients) are bit strings
  - Each bit corresponds to a randomized response for one of the client’s properties

1	**1**	**1**	**0**
Male	communist	Between 18-35 yrs old	Between 36-50 yrs old

0	**0**	**0**	**1**
Female	~~communist~~	Between 18-35 yrs old	Between 36-50 yrs old

Crowdsourcing statistics with RAPPOR

- RAPPOR responses (from clients) are bit strings
  - Each bit corresponds to a randomized response for one of the client's properties

1	1	1	0
Male	communist	Between 18-35 yrs old	Between 36-50 yrs old

0	0	0	1
Female	~~communist~~	Between 18-35 yrs old	Between 36-50 yrs old

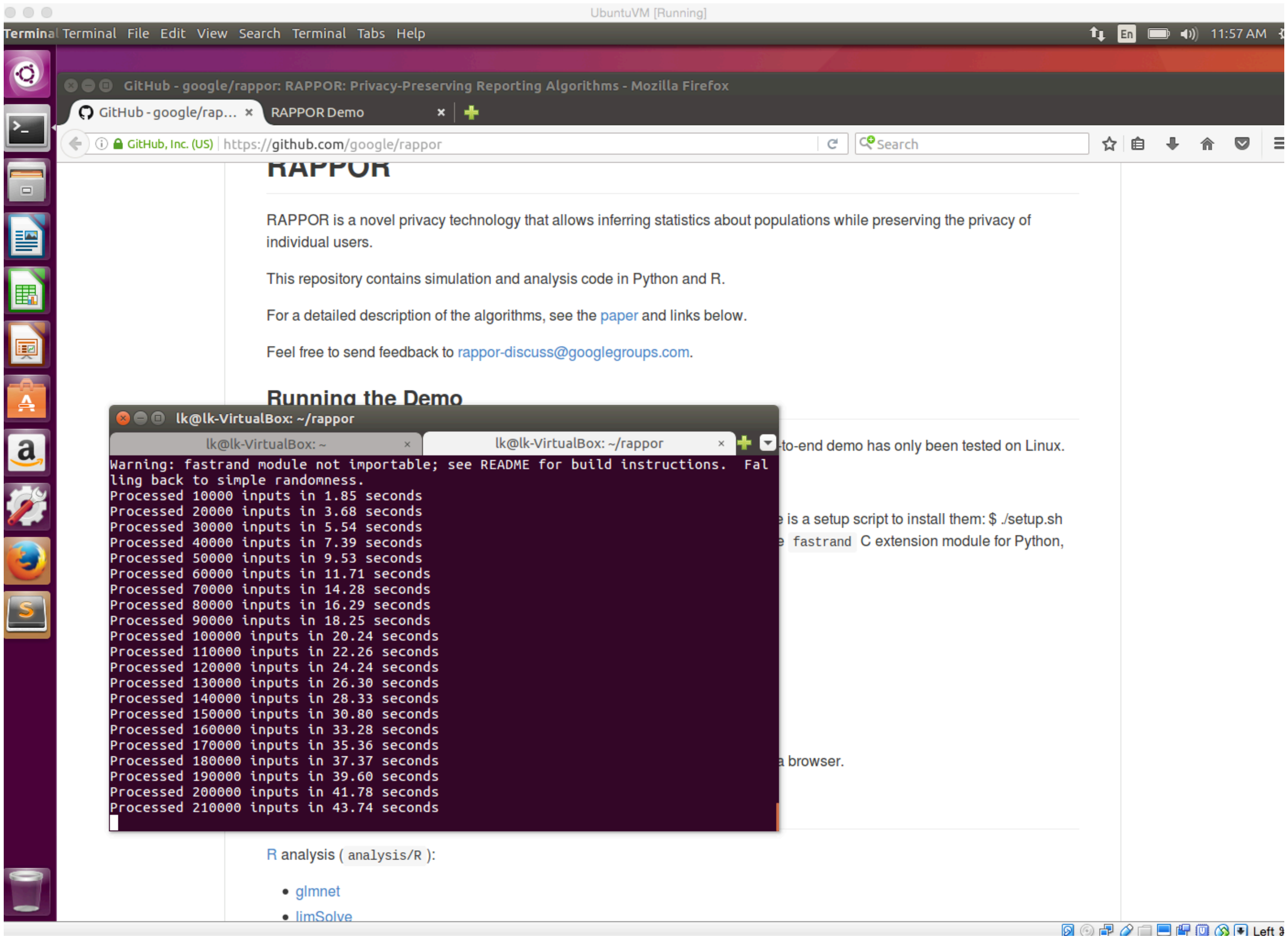
Collecting statistics on categories that cannot be enumerated ahead of time

- Using ***Bloom filters***
- Probabilities data structure that can be used to test whether a response is in a set
- Query returns either:
  - Possibly in set
  - Definitely not in set

Explicit trade-offs between different attack models: *Tunable privacy protection*

Attacker	**Capabilities**	**Response**
Basic	Only has access to a single report	Can be stopped with a single round of randomized response
Windowed	Has access to multiple reports over time from the same user	Careful modification of the traditional randomized response techniques needed
Insider	Complete access to all clients reports	Hardest to stop, but most difficult to execute in practice

Demo


UbuntuVM [Running]
Terminal Terminal File Edit View Search Terminal Tabs Help
11:57 AM
GitHub - google/rappor: RAPPOR: Privacy-Preserving Reporting Algorithms - Mozilla Firefox
GitHub - google/rap...
RAPPOR Demo
GitHub, Inc. (US) https://github.com/google/rappor
Search
lk@lk-VirtualBox: ~/rappor
lk@lk-VirtualBox: ~
lk@lk-VirtualBox: ~/rappor

RAPPOR

RAPPOR is a novel privacy technology that allows inferring statistics about populations while preserving the privacy of individual users.

This repository contains simulation and analysis code in Python and R.

For a detailed description of the algorithms, see the paper and links below.

Feel free to send feedback to rappor-discuss@googlegroups.com.

Running the Demo

-to-end demo has only been tested on Linux.

is a setup script to install them: $ ./setup.sh

`fastrand` C extension module for Python,

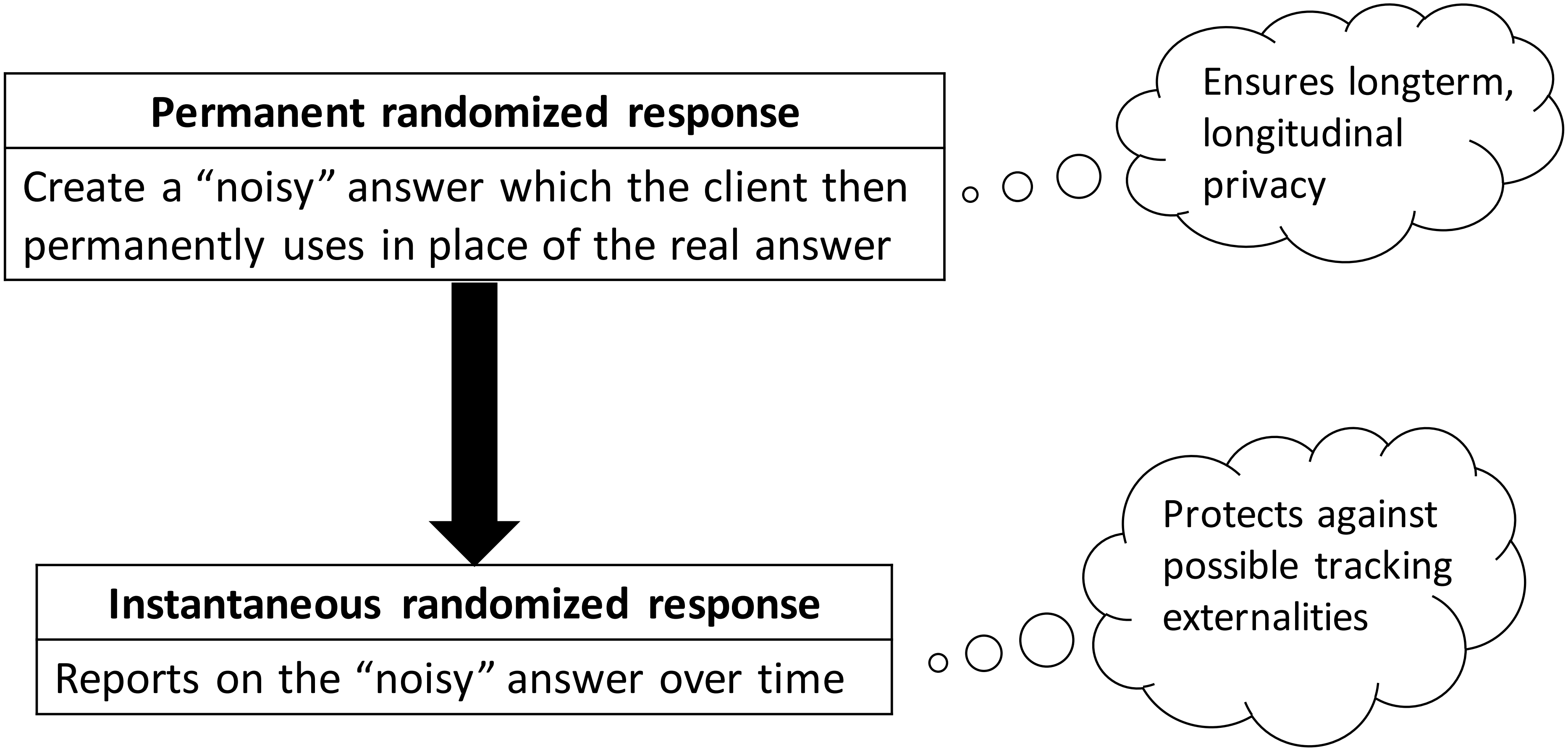
browser.

R analysis (`analysis/R`):

- glmnet
- limSolve

```
Warning: fastrand module not importable; see README for build instructions.  Fal
ling back to simple randomness.
Processed 10000 inputs in 1.85 seconds
Processed 20000 inputs in 3.68 seconds
Processed 30000 inputs in 5.54 seconds
Processed 40000 inputs in 7.39 seconds
Processed 50000 inputs in 9.53 seconds
Processed 60000 inputs in 11.71 seconds
Processed 70000 inputs in 14.28 seconds
Processed 80000 inputs in 16.29 seconds
Processed 90000 inputs in 18.25 seconds
Processed 100000 inputs in 20.24 seconds
Processed 110000 inputs in 22.26 seconds
Processed 120000 inputs in 24.24 seconds
Processed 130000 inputs in 26.30 seconds
Processed 140000 inputs in 28.33 seconds
Processed 150000 inputs in 30.80 seconds
Processed 160000 inputs in 33.28 seconds
Processed 170000 inputs in 35.36 seconds
Processed 180000 inputs in 37.37 seconds
Processed 190000 inputs in 39.60 seconds
Processed 200000 inputs in 41.78 seconds
Processed 210000 inputs in 43.74 seconds
```

How RAPPOR works



Permanent randomized response

Create a “noisy” answer which the client then permanently uses in place of the real answer

Ensures longterm, longitudinal privacy

Instantaneous randomized response

Reports on the “noisy” answer over time

Protects against possible tracking externalities

References

- Installation
  - https://github.com/google/rappor
- Online Demo
  - http://google.github.io/rappor/examples/report.html