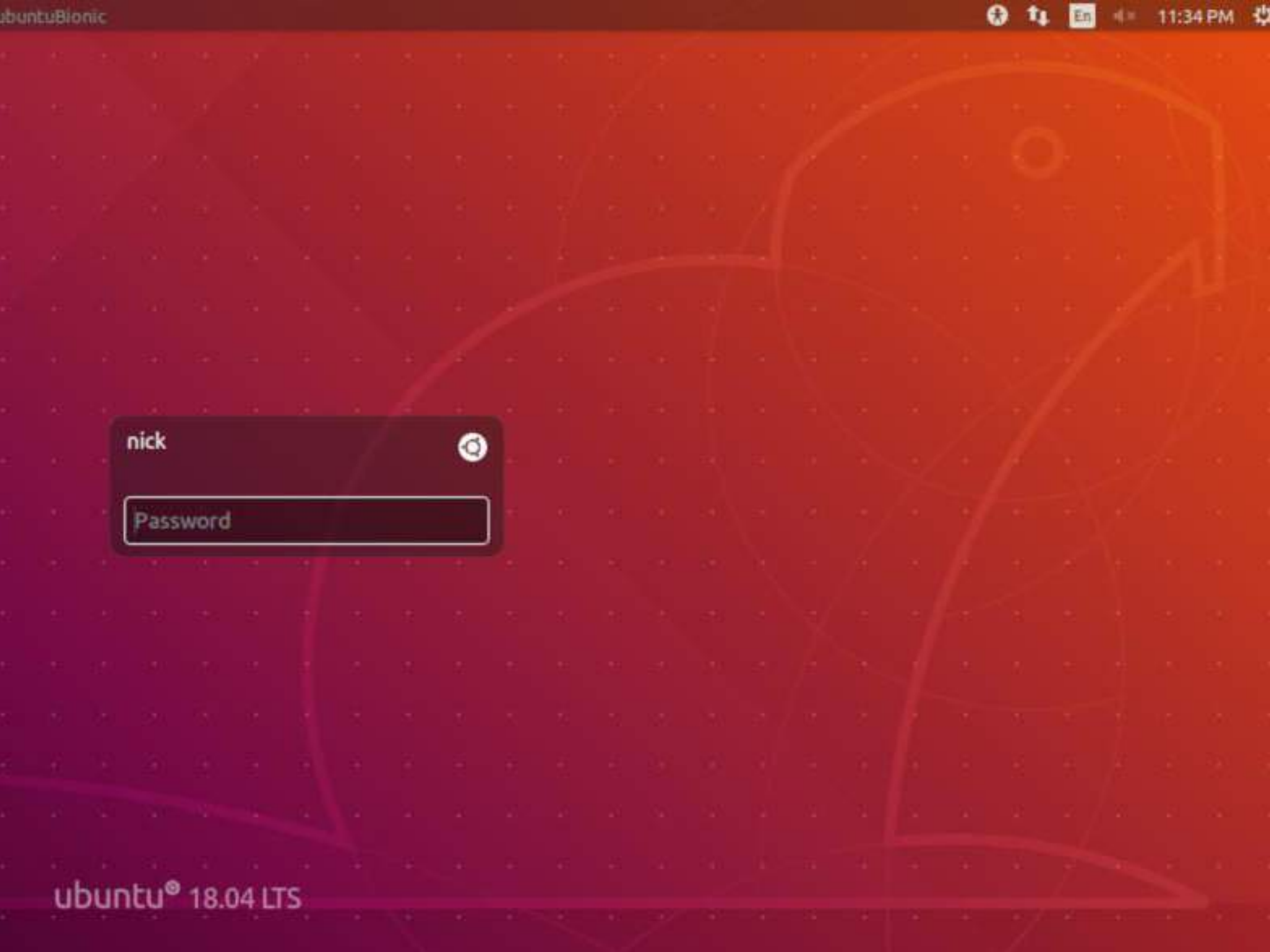
Side and Covert Channels: *the Dr. Jekyll and Mr Hyde of Modern Technologies*

Mauro Conti

SPRITZ
Security & Privacy
Research Group

Università
degli Studi
di Padova


ubuntuBionic
En
11:34 PM
nick
Password
ubuntu® 18.04 LTS

Outline

- Covert and Side Channels 101

- Network Traffic Analysis
  - *As a side channel: app and sensitive data inference*
- Energy Consumption
  - *As a side channel: user and app inference*
  - *As a covert channel: data exfiltration*

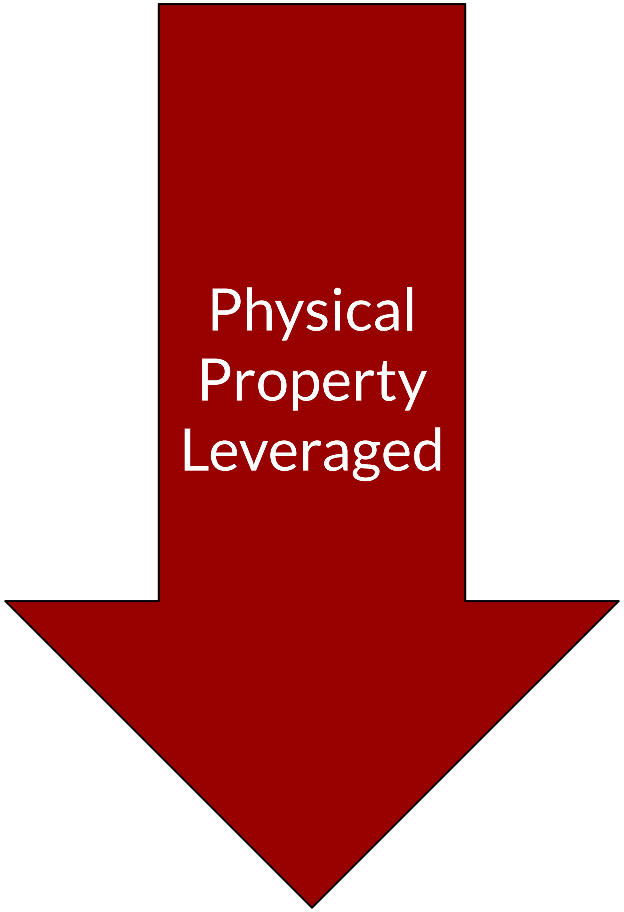
- Device Movement
  - *As a side channel: smartphone user authentication*
  - *Attacks against biometric authentication*
- Keystroke Timing
  - *As a side channel: text typed on keyboards*
- Acoustic Emanations
  - *As a side channel: text typed on keyboards*

Outline

- Covert and Side Channels 101

- Network Traffic Analysis
  - *As a side channel: app and sensitive data inference*
- Energy Consumption
  - *As a side channel: user and app inference*
  - *As a covert channel: data exfiltration*

- Device Movement
  - *As a side channel: smartphone user authentication*
  - *Attacks against biometric authentication*
- Keystroke Timing
  - *As a side channel: text typed on keyboards*
- Acoustic Emanations
  - *As a side channel: text typed on keyboards*



Outline

- Covert and Side Channels 101

- Network Traffic Analysis
  - *As a side channel: app and sensitive data inference*
- Energy Consumption
  - *As a side channel: user and app inference*
  - *As a covert channel: data exfiltration*

- Device Movement
  - *As a side channel: smartphone user authentication*
  - *Attacks against biometric authentication*
- Keystroke Timing
  - *As a side channel: text typed on keyboards*
- Acoustic Emanations
  - *As a side channel: text typed on keyboards*

Keystroke Inference

Outline

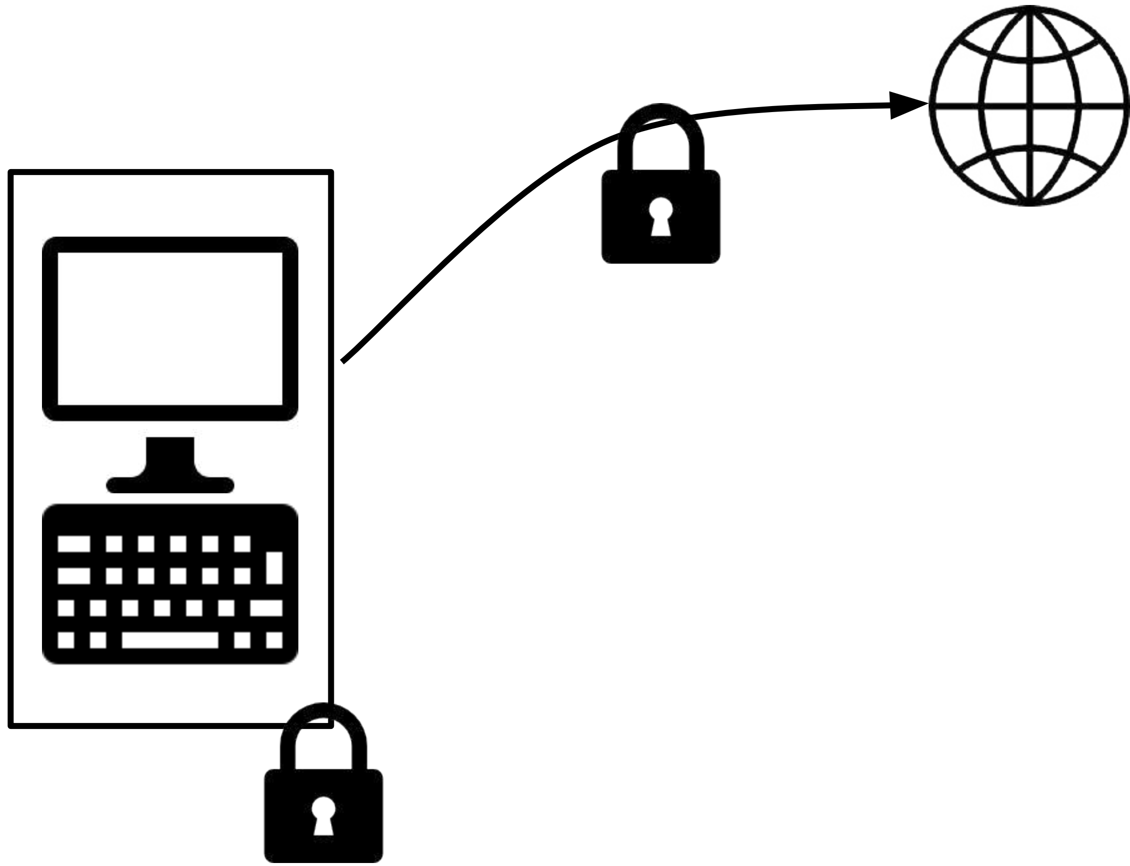
- **Covert and Side Channels 101**

- Network Traffic Analysis
  - *As a side channel: app and sensitive data inference*
- Energy Consumption
  - *As a side channel: user and app inference*
  - *As a covert channel: data exfiltration*

- Device Movement
  - *As a side channel: smartphone user authentication*
  - *Attacks against biometric authentication*
- Keystroke Timing
  - *As a side channel: text typed on keyboards*
- Acoustic Emanations
  - *As a side channel: text typed on keyboards*

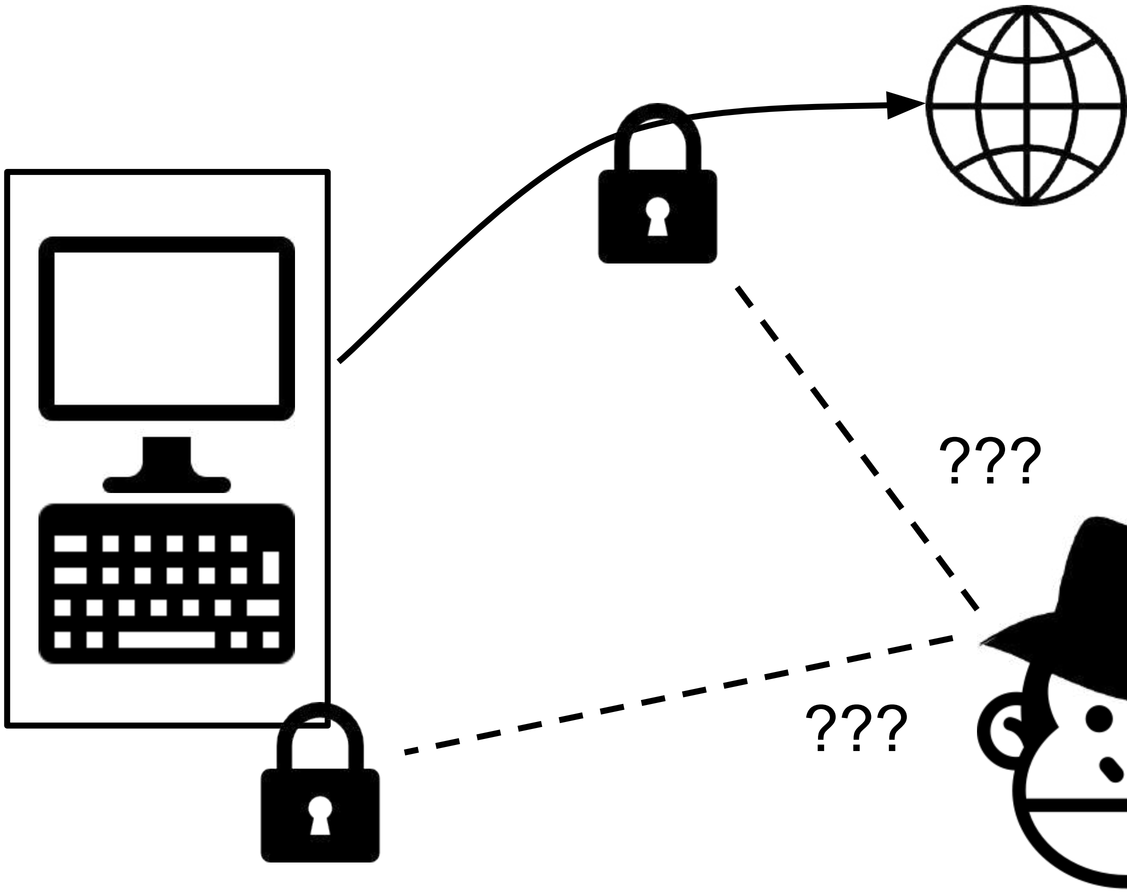
Side Channels

Devices, and network communication, are usually **protected** and **encrypted**



Devices, and network communication, are usually **protected** and **encrypted**

→ Difficult for **Attackers** to violate such protecion

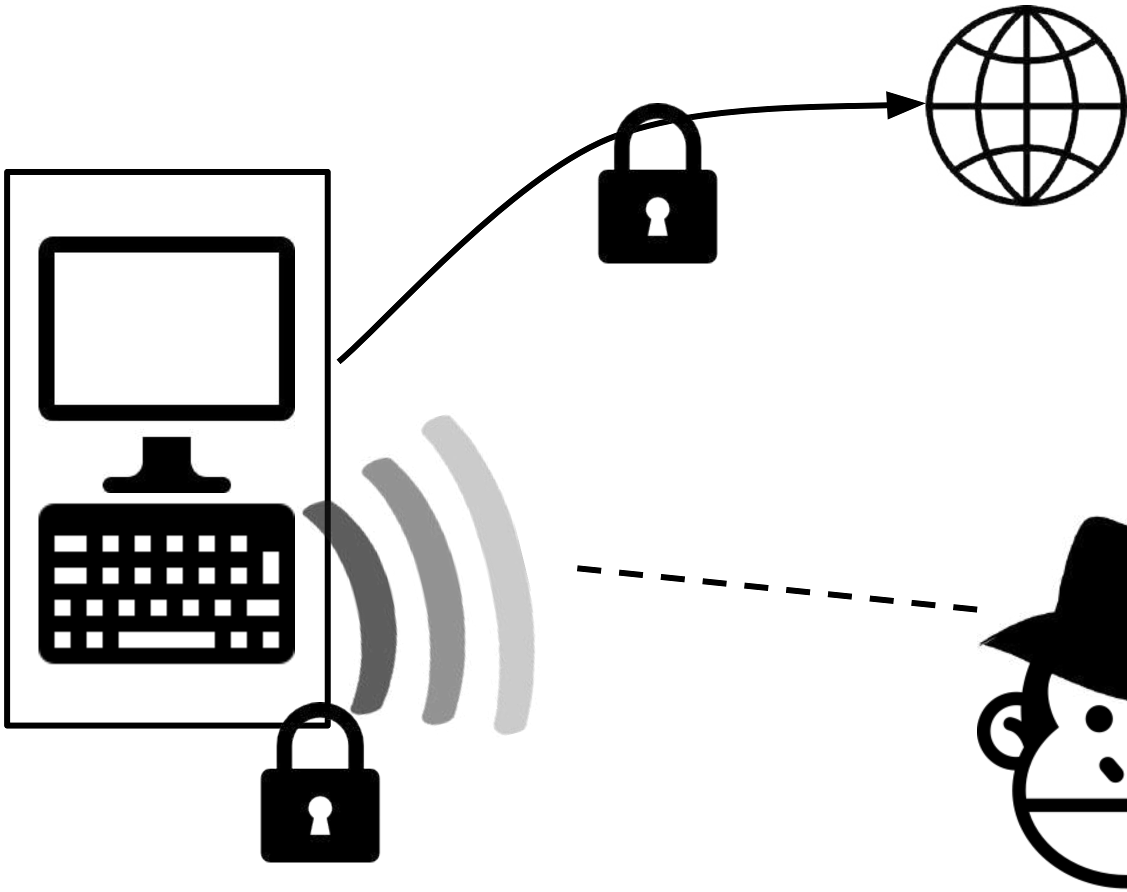


Side Channels

Observing emanations and patterns

Can reveal secrets!

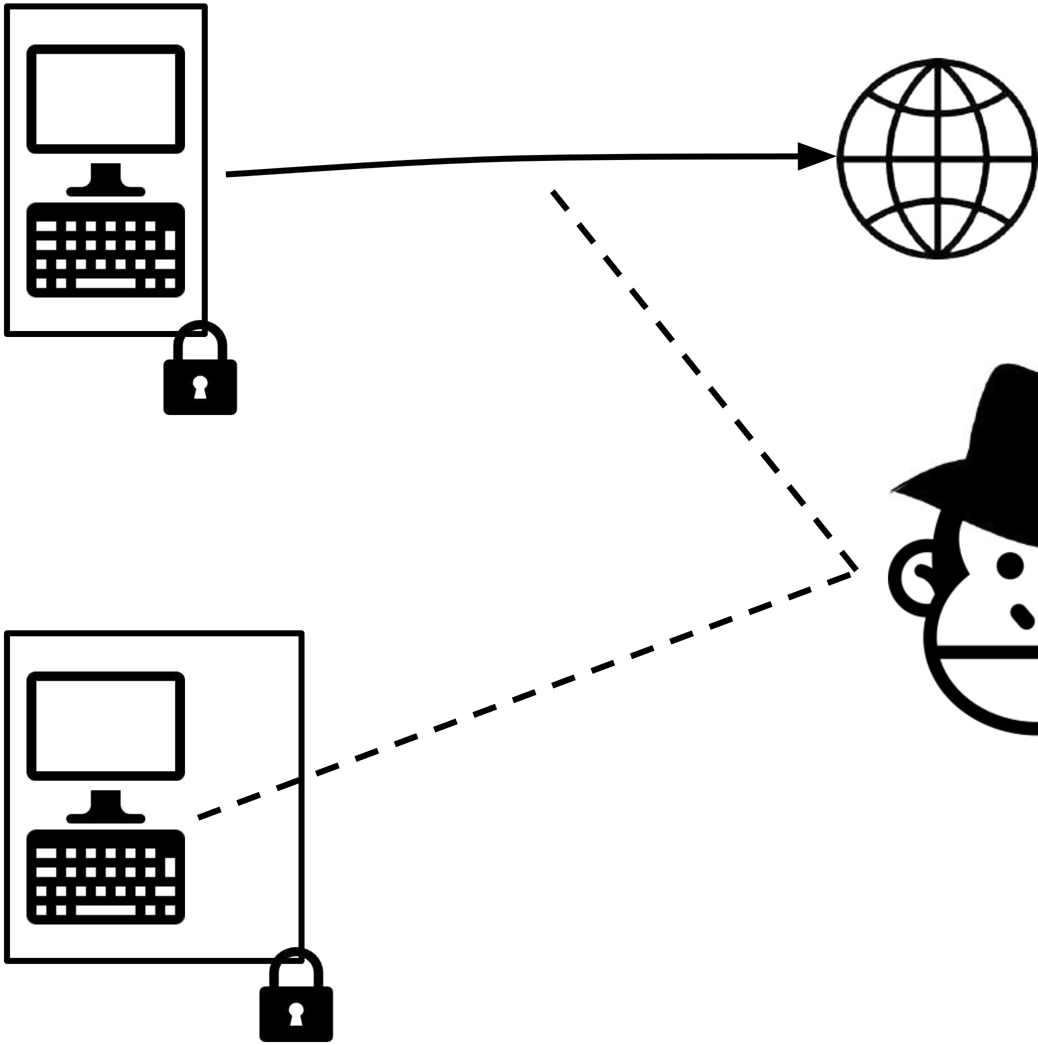
This is called a **side channel**



Covert Channels

Covert Channels are used to communicate stealthily.

Either to **avoid listeners in the middle**...



...or to exfiltrate information.

Outline

- Covert and Side Channels 101

- **Network Traffic Analysis**
  - ***As a side channel: app and sensitive data inference***
- Energy Consumption
  - *As a side channel: user and app inference*
  - *As a covert channel: data exfiltration*

- Device Movement
  - *As a side channel: smartphone user authentication*
  - *Attacks against biometric authentication*
- Keystroke Timing
  - *As a side channel: text typed on keyboards*
- Acoustic Emanations
  - *As a side channel: text typed on keyboards*

M. Conti, L. V. Mancini, R. Spolaor, N. V. Verde.

Can't you hear me knocking: Identification of user actions on Android apps via traffic analysis.

In ACM SIGSAC CODASPY 2015

V. F. Taylor, R. Spolaor, M. Conti, I. Martinovic.

AppScanner: Automatic Fingerprinting of Smartphone Apps From Encrypted Network Traffic.

In IEEE EuroSP 2016

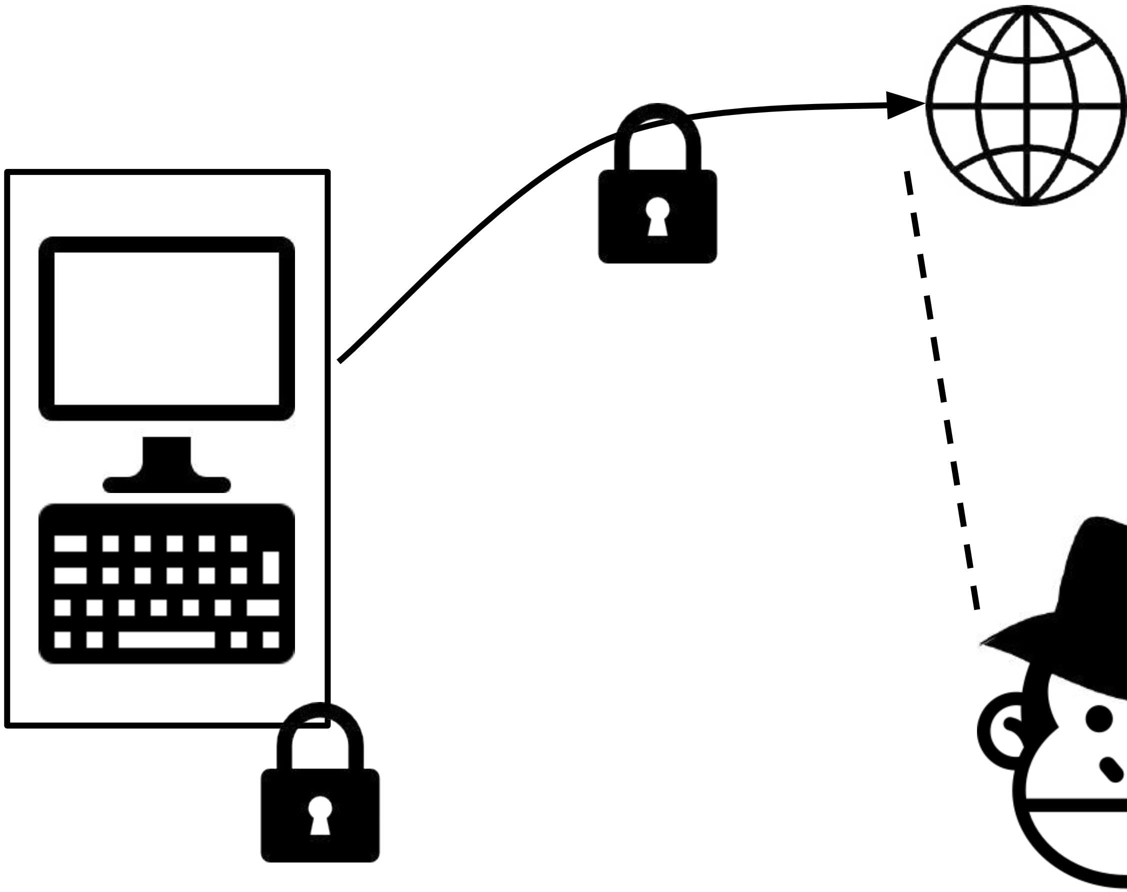
Traffic Analysis

Traffic patterns

Can reveal what we are doing!

Device-platform interaction reveals our actions

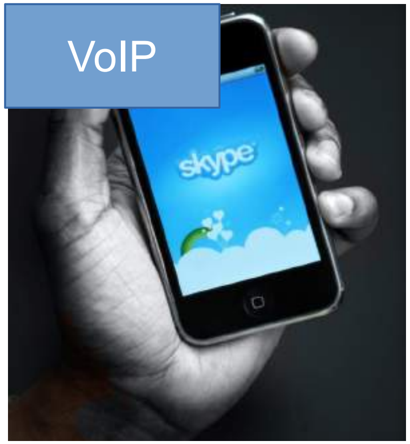
Called **traffic analysis**



Motivation

Encryption is not enough!

[Song et al. '11]

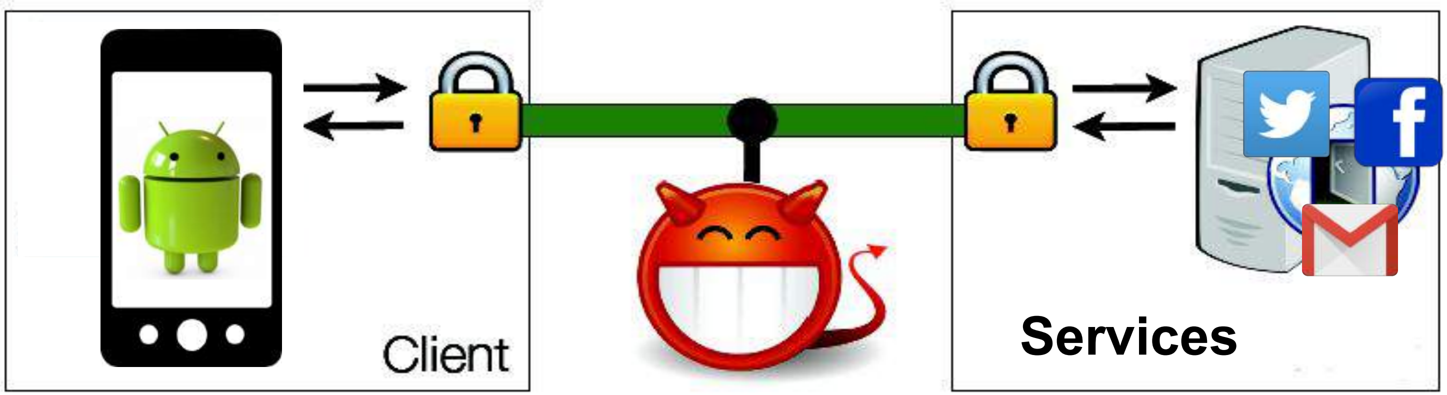


[Wright et al. '08]

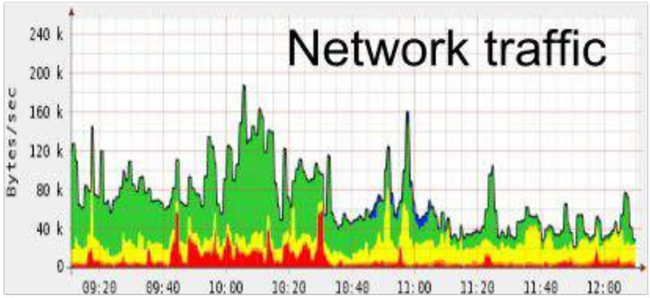
Attacker's observations

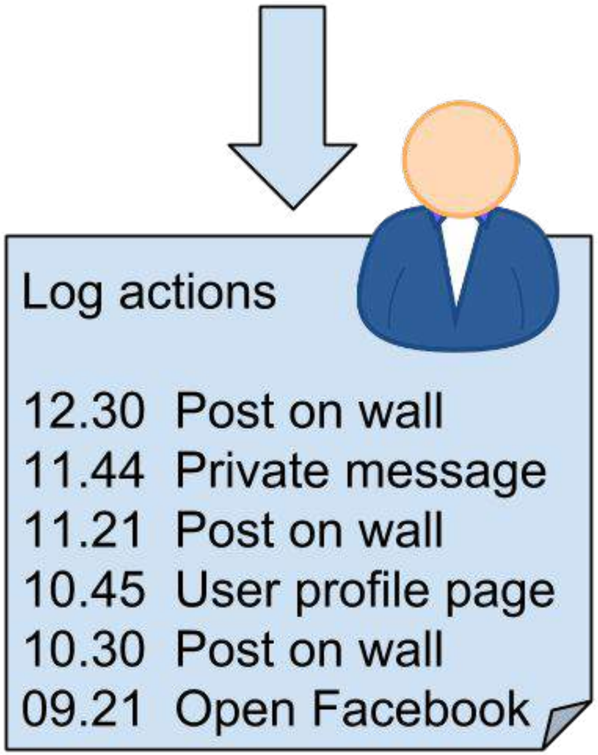
- Coarse features:
  - Packet lengths
  - Packet directions
  - Packet timings
  - ....

Enable Traffic Analysis Attacks

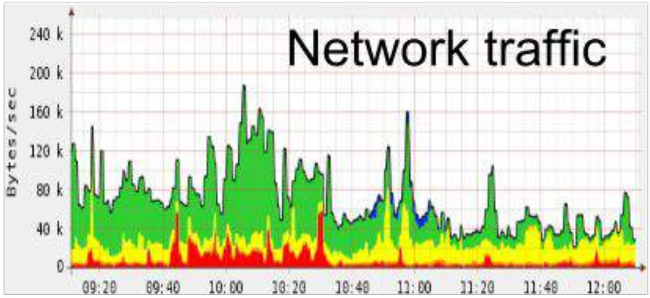


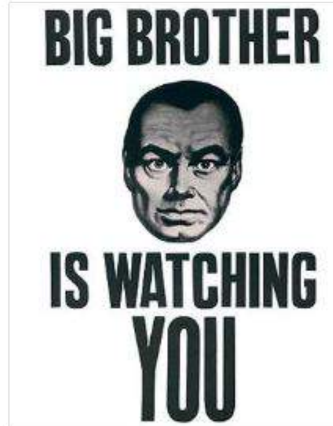
Attack scenario

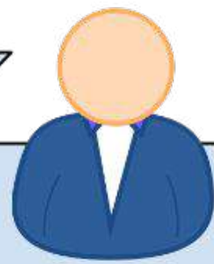




Attack scenario



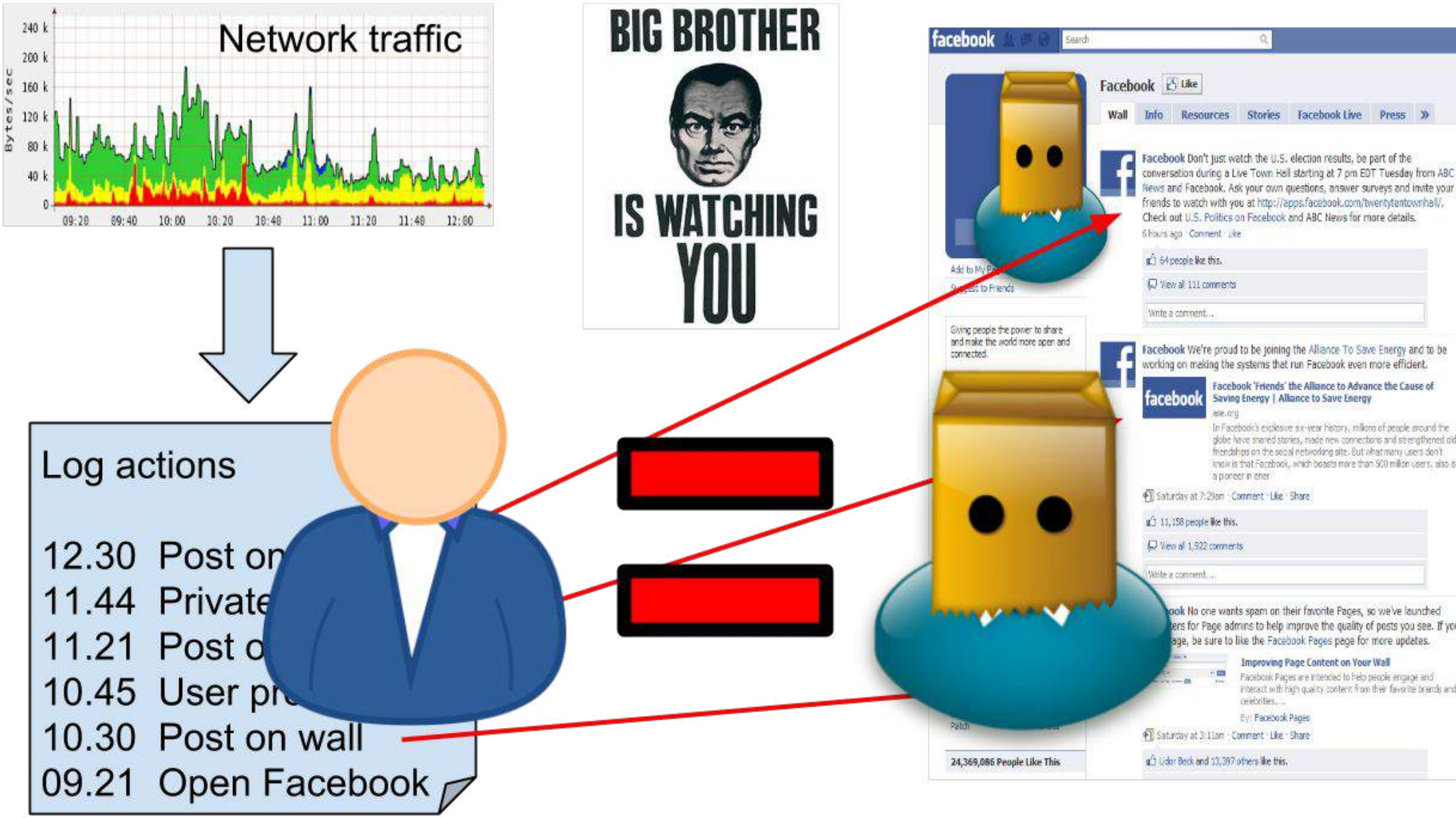




Log actions

12.30 Post on wall
11.44 Private message
11.21 Post on wall
10.45 User profile page
10.30 Post on wall
09.21 Open Facebook

Attack scenario



Other attack scenarios

- To identify communicating parties
  - from sending/receiving pattern
- Behavioural profiling
  - to improve fingerprintings
  - for marketing reasons
  - ...

The goal

Can an attacker recognize actions that a user performs on some android app by analyzing the **encrypted network traffic**?

Contribution

- We prove that it is possible, with an accuracy $> 95\%$
- Traffic analysis using **machine learning** techniques

Key Concepts

Interactions

Input on a device

E.g., tap, swipe, key press

used to achieve

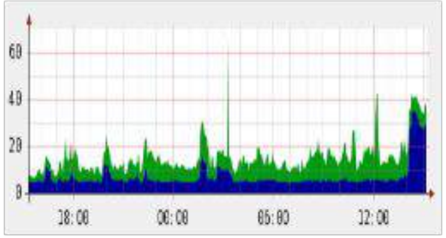
User actions

Operation on apps

E.g., send an email, open a page

produce

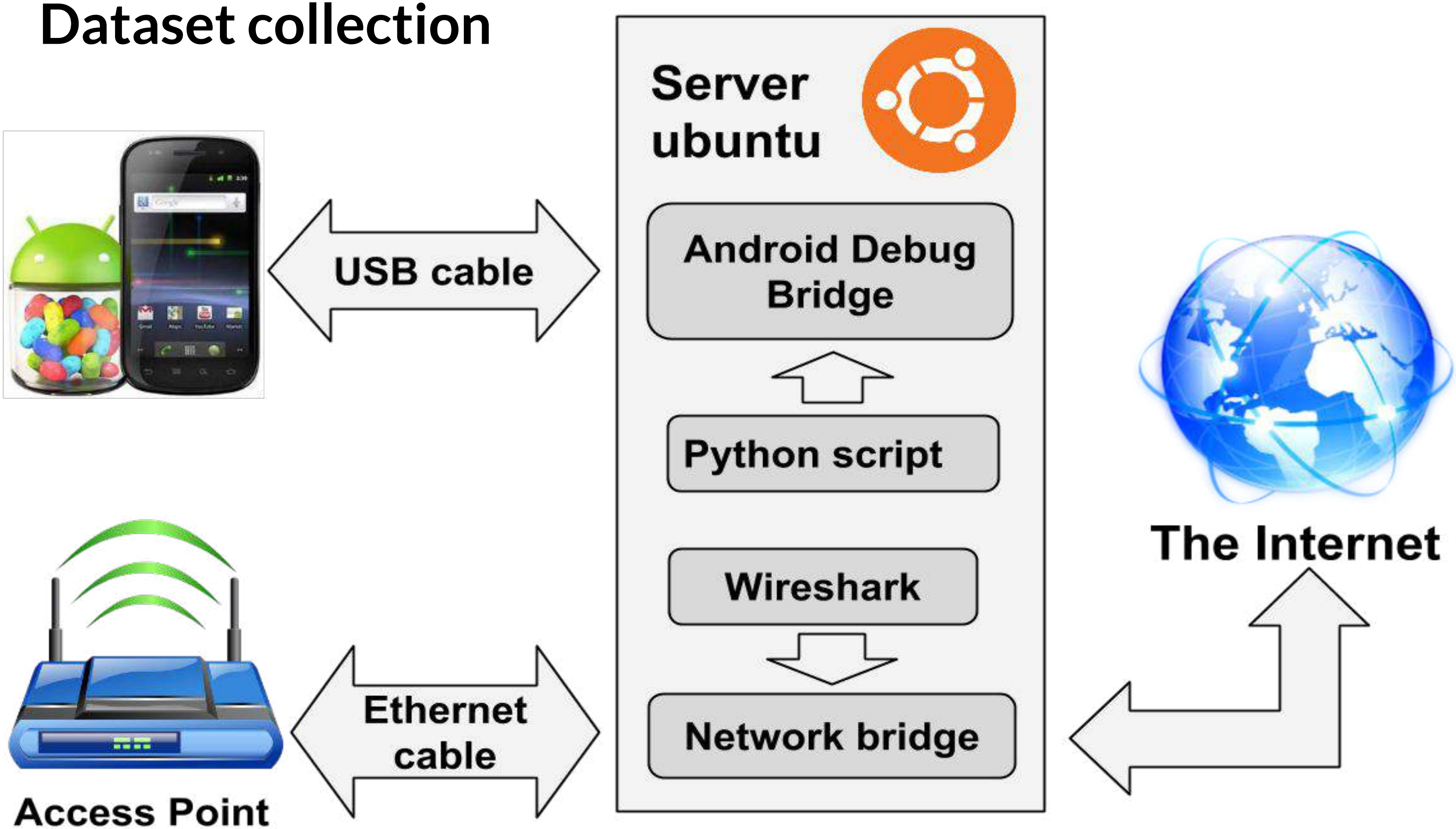
Network flows



Sequence of packets

Couple of IP addresses and ports

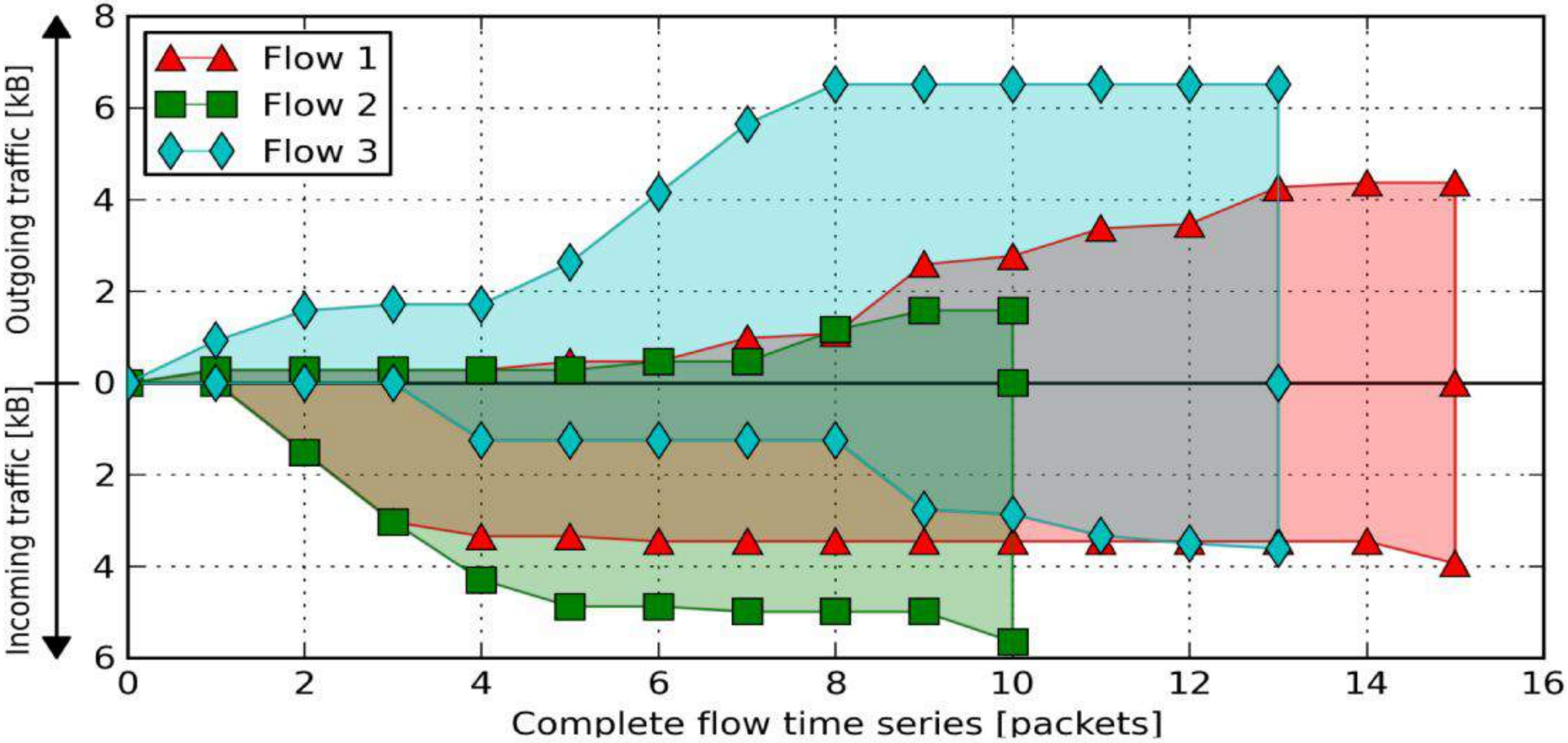
Dataset collection



SPRITZ Security & Privacy Research Group

Università degli Studi di Padova

Network Traffic Flows Representation



Flow ID	Flow time series
Flow 1	[282, -1514, -1514, -315, 188, -113, 514, 96, 1514, 179, 603, 98, 801, 98, -477]
Flow 2	[282, -1514, -1514, -1266, -582, 188, -113, 692, 423, -661]
Flow 3	[926, 655, 136, -1245, 913, 1514, 1514, 863, -1514, -107, -465, -172, -111]

Can't you hear me knocking (CODASPY '14, TIFS '15)

SPRITZ Security & Privacy Research Group
Università degli Studi di Padova

The framework

Labeled Dataset

Phase 1.
Training

Phase 2.
Testing

Network Traffic Pre-processor

User Action Classifier

Random Forest

Predictions
- Tweet sent
- email answered
- tweeter contact opened...

Training phase

1. Unsupervised learning →**Clusters** of similar flows
   - **Dynamic Time Warping** (DTW) [Müller 2007] as metric
   - The **number of clusters** is a parameter to tune

2. Training set building
   - User actions → Classes
   - Cluster labels → Features

IDs	user actions	cluster 0	cluster 1	...	cluster k	....	cluster N-1	cluster N
001	send mail	0	1	...	1	...	0	0
002	send mail	0	1	...	1	...	0	0
003	send reply	1	0	...	2	...	1	0
....	...	...	...	...	...	...	...	...

3. Supervised learning → Random Forest **classifier**

Evaluation phase

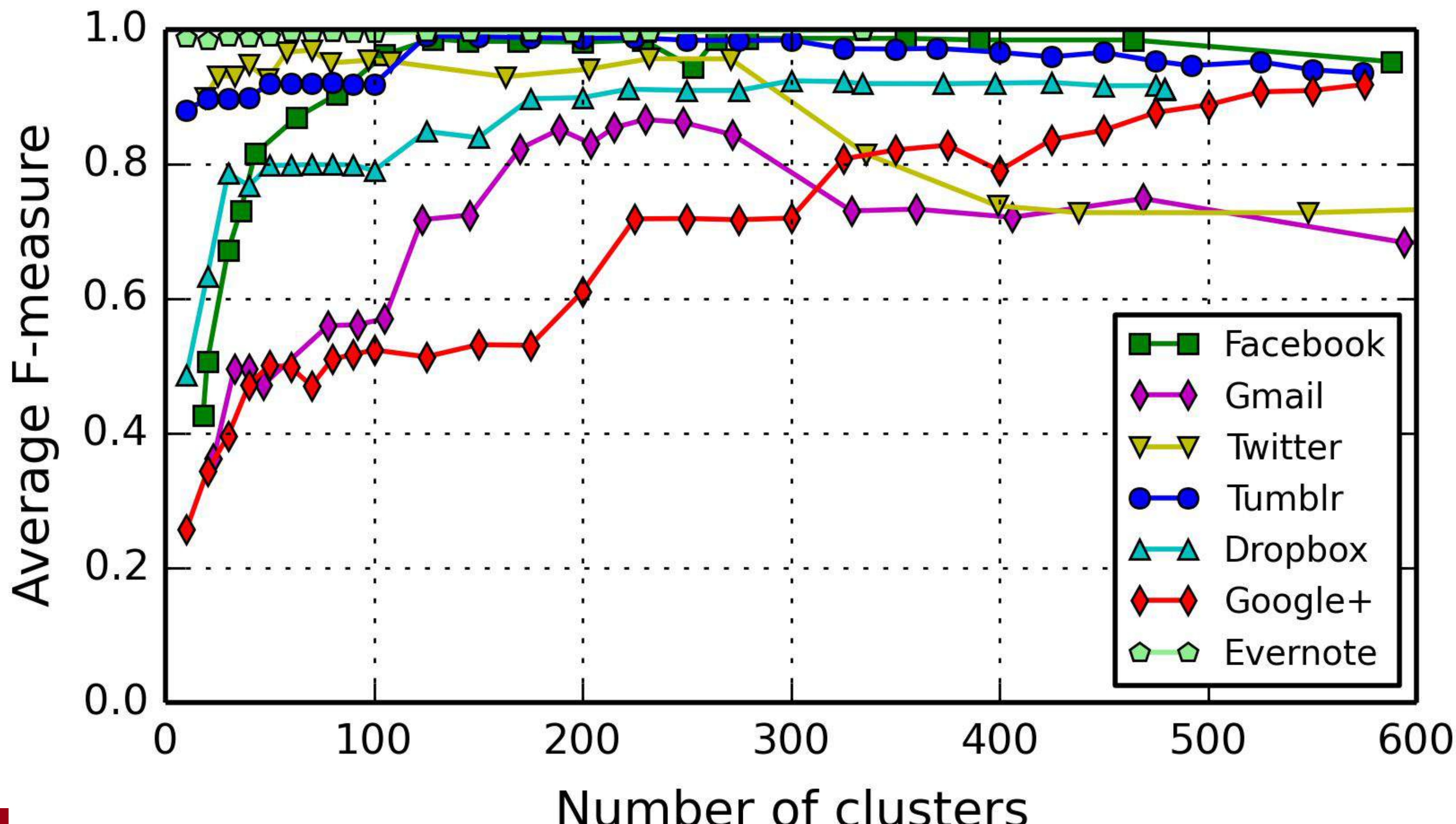
1. User actions produce **unseen flows**

2. Assign each **unseen flow** to a **cluster**
   - clusters used in **training** phase and **DTW** as metric

3. Test set building
   - (similarly to training set)
   - User actions → **unknown classes**
   - Cluster labels → Features

4. User action **recognition**

Accuracy vs. number of clusters

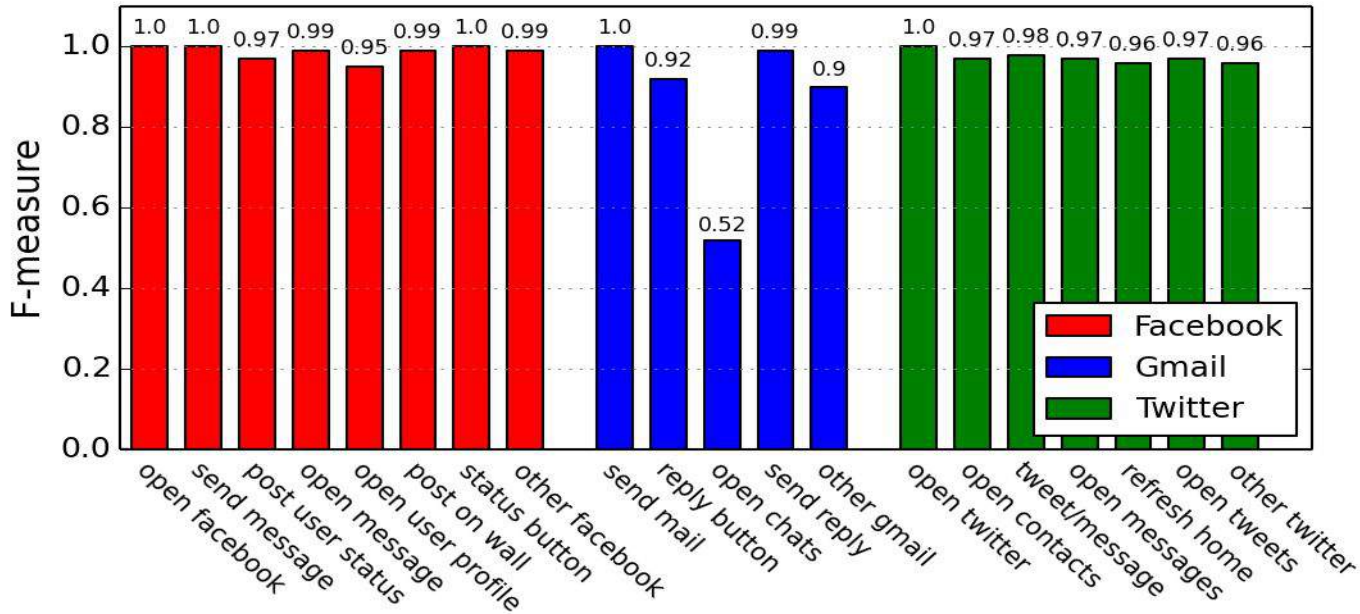


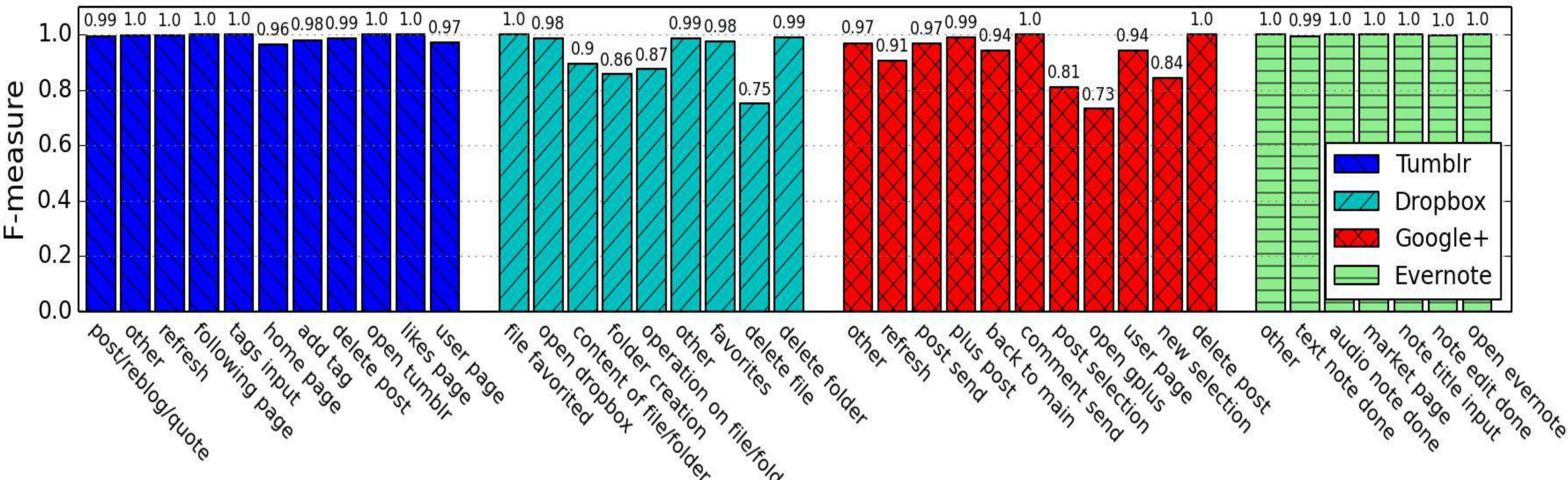
Can't you hear me knocking (CODASPY '14, TIFS '15)

SPRITZ Security & Privacy Research Group

Università degli Studi di Padova

Accuracy per user action





Conclusions

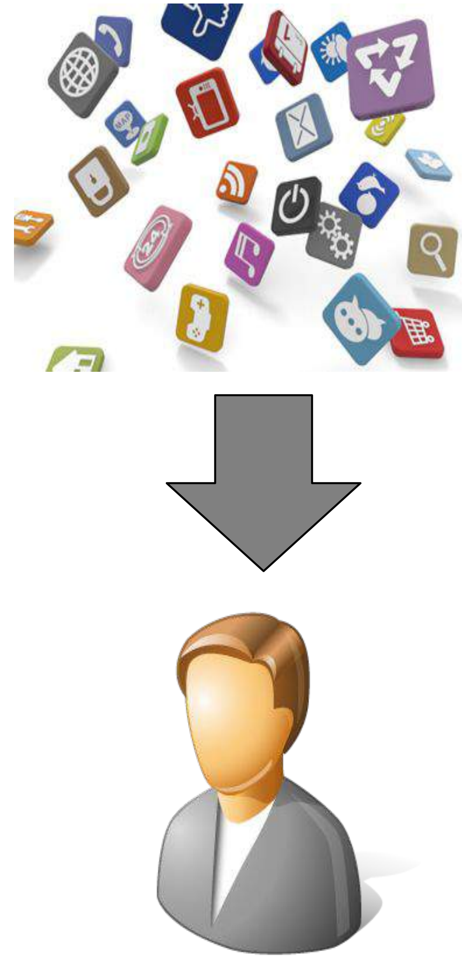
- Encryption does not hide communication patterns
  - We shown that user actions performed on Android apps can be detected by analyzing the encrypted network traffic
- Attackers can leverage our framework to undermine user privacy:
  - Learn user habits
  - Gain commercial or intelligence advantage against some competitor
  - Attribution of social network pseudonyms
- Countermeasures to this type of attacks are needed...

AppScanner
(IEEE EuroS&P ‘16)

Motivation (1)

From the set of **apps installed** on a device can be inferred private information about her **owner:**

- Age
- Sex
- Religion
- Relationship status
- Spoken languages
- Countries of interest

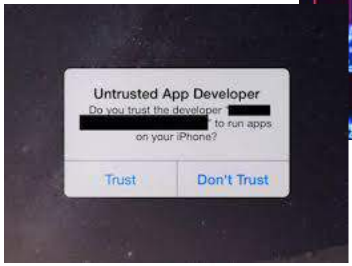


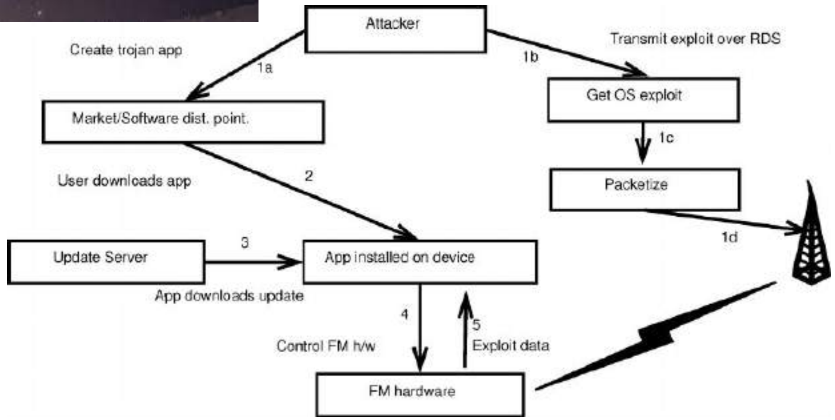
S. Seneviratne, A. Seneviratne, P. Mohapatra, A. Mahanti. “Predicting User Traits From a Snapshot of Apps Installed on a Smartphone” in ACM SIGMOBILE Mobile Computing and Communications Review 2014.

Motivation (2)

Knowing a presence of a specific app
- Hence specific vulnerabilities

Possible ad-hoc attacks
- E.g., zero day exploits





Motivation

- Given a target app X
- Identify the presence of X in a mobile device
- Using network traffic analysis

Motivation

- Given a target app X
- Identify the presence of X in a mobile device
- Using network traffic analysis

It isn’t so easy!

Motivation

- Given a target app X
- Identify the presence of X in a mobile device
- Using network traffic analysis

It isn't so easy!

- Encryption → Payload inspection is not feasible

Motivation

- Given a target app X
- Identify the presence of X in a mobile device
- Using network traffic analysis

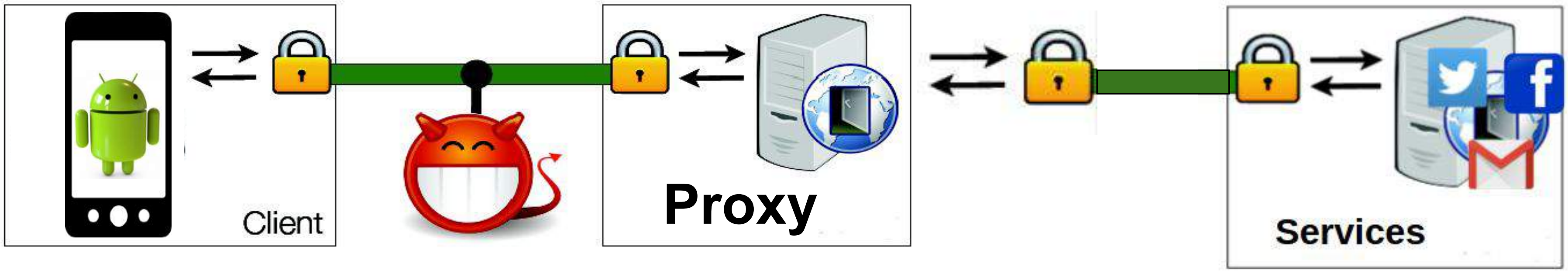
It isn't so easy!

- Encryption → Payload inspection is not feasible
- Owner of Destination IP ≠ App
  - Content Delivery Network (CDN)
  - Proxy

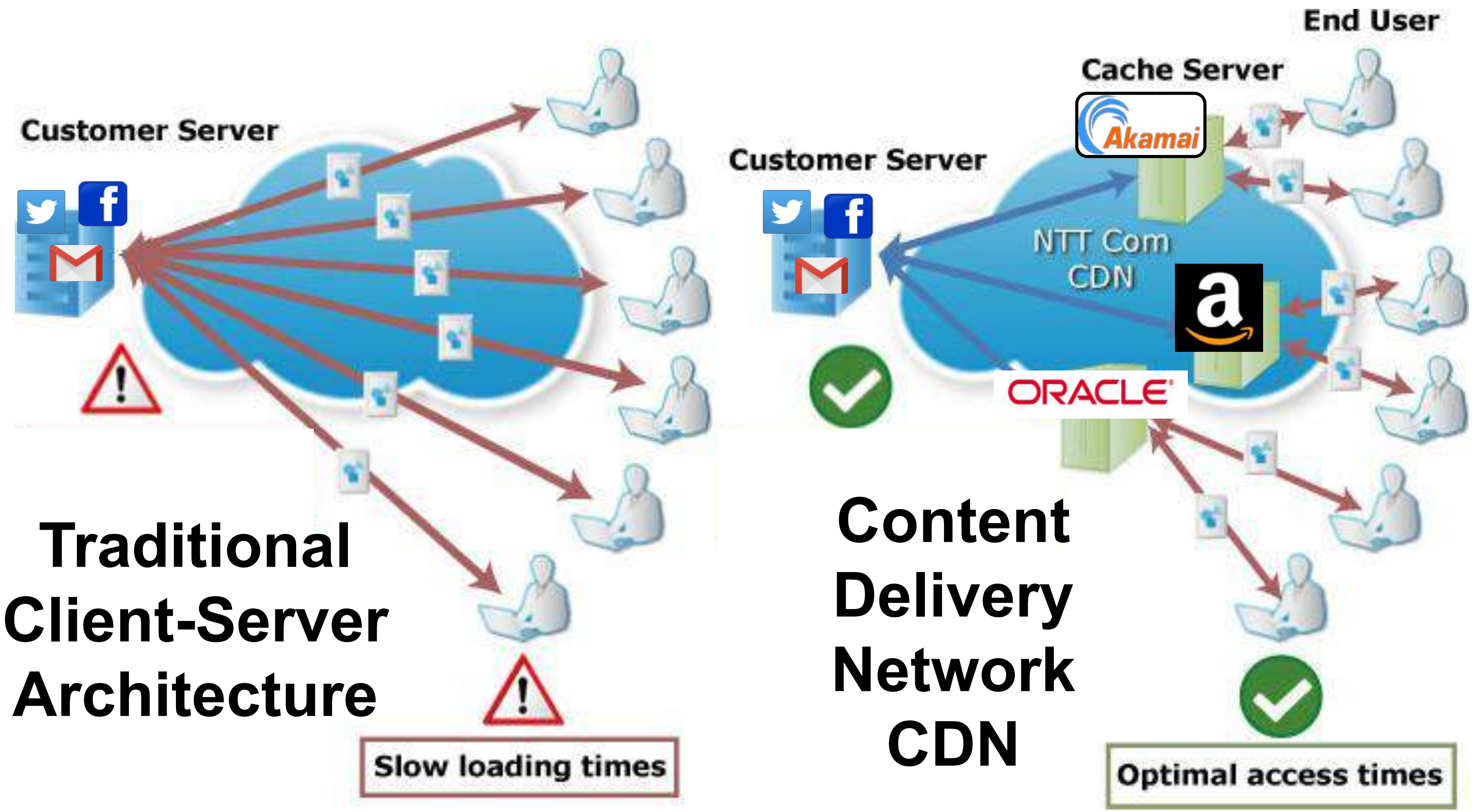
Attacker's observations (similarly to the previous work)

- Packet length
- Packet directions
- Packet timings

Enable Traffic Analysis Attacks



AppScanner (IEEE EuroS&P '16)



AppScanner
(IEEE EuroS&P '16)

Three different approaches proposed:

Three different approaches proposed:

1. **Per flow** length classification
   - A classifier for each length
   - No out-of-order packets resiliency, but fast

AppScanner (IEEE EuroS&P '16)

Three different approaches proposed:

1. **Per flow** length classification
   - A classifier for each length
   - No out-of-order packets resiliency, but fast

2. **Large Multi-class** classification
   - Uses statistics on network flows
   - It works on a **set of apps**
   - **High Accuracy** and out-of-order packets resiliency, but slow

AppScanner
(IEEE EuroS&P '16)

SPRITZ
Security & Privacy
Research Group

Università
degli Studi
di Padova

Three different approaches proposed:

1. **Per flow** length classification
   - A classifier for each length
   - No out-of-order packets resiliency, but fast

2. **Large Multi-class** classification
   - Uses statistics on network flows
   - It works on a **set of apps**
   - **High Accuracy** and out-of-order packets resiliency, but slow

3. **Per App** classification
   - Uses statistics on network flows
   - It focuses on a **specific app**
   - Binary classification (app is present of not)

Building the dataset

TCP Packets captured

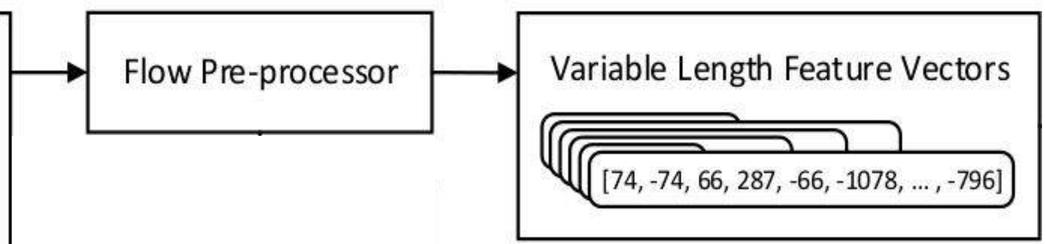
SOURCE_IP	DEST_IP	PROTO	LEN
192.168.137.2	23.23.162.140	TCP	74
23.23.162.140	192.168.137.2	TCP	74
192.168.137.2	23.23.162.140	TCP	66
192.168.137.2	23.23.162.140	TLSv1	287
23.23.162.140	192.168.137.2	TCP	66
23.23.162.140	192.168.137.2	TLSv1	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	114
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TLSv1	796

SPRITZ Security & Privacy Research Group
Università degli Studi di Padova

Building the dataset

TCP Packets captured

SOURCE_IP	DEST_IP	PROTO	LEN
192.168.137.2	23.23.162.140	TCP	74
23.23.162.140	192.168.137.2	TCP	74
192.168.137.2	23.23.162.140	TCP	66
192.168.137.2	23.23.162.140	TLSv1	287
23.23.162.140	192.168.137.2	TCP	66
23.23.162.140	192.168.137.2	TLSv1	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	114
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TLSv1	796



SPRITZ Security & Privacy Research Group

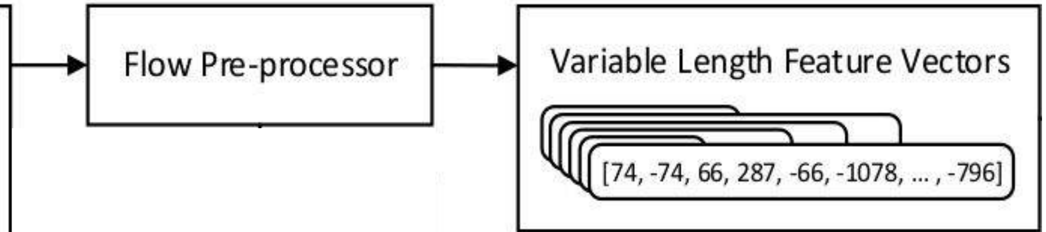
Università degli Studi di Padova

Building the dataset

TCP Packets captured

SOURCE_IP	DEST_IP	PROTO	LEN
192.168.137.2	23.23.162.140	TCP	74
23.23.162.140	192.168.137.2	TCP	74
192.168.137.2	23.23.162.140	TCP	66
192.168.137.2	23.23.162.140	TLSv1	287
23.23.162.140	192.168.137.2	TCP	66
23.23.162.140	192.168.137.2	TLSv1	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	114
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TLSv1	796

Per Flow approach (1)



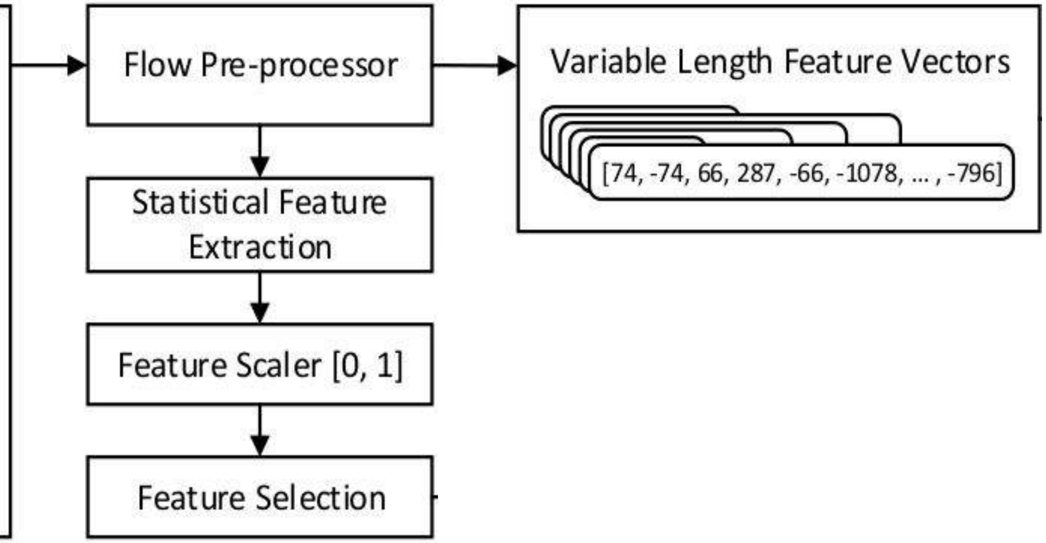
AppScanner (IEEE EuroS&P '16)

Building the dataset

TCP Packets captured

SOURCE_IP	DEST_IP	PROTO	LEN
192.168.137.2	23.23.162.140	TCP	74
23.23.162.140	192.168.137.2	TCP	74
192.168.137.2	23.23.162.140	TCP	66
192.168.137.2	23.23.162.140	TLSv1	287
23.23.162.140	192.168.137.2	TCP	66
23.23.162.140	192.168.137.2	TLSv1	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	114
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TLSv1	796

Per Flow approach (1)



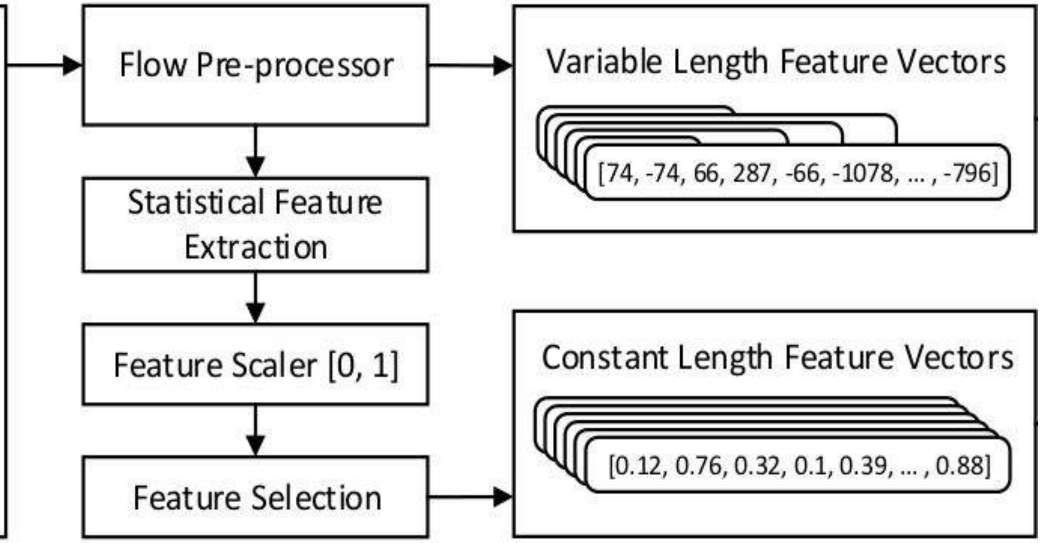
SPRITZ Security & Privacy Research Group
Università degli Studi di Padova

Building the dataset

TCP Packets captured

SOURCE_IP	DEST_IP	PROTO	LEN
192.168.137.2	23.23.162.140	TCP	74
23.23.162.140	192.168.137.2	TCP	74
192.168.137.2	23.23.162.140	TCP	66
192.168.137.2	23.23.162.140	TLSv1	287
23.23.162.140	192.168.137.2	TCP	66
23.23.162.140	192.168.137.2	TLSv1	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TCP	114
23.23.162.140	192.168.137.2	TCP	1078
23.23.162.140	192.168.137.2	TLSv1	796

Per Flow approach (1)

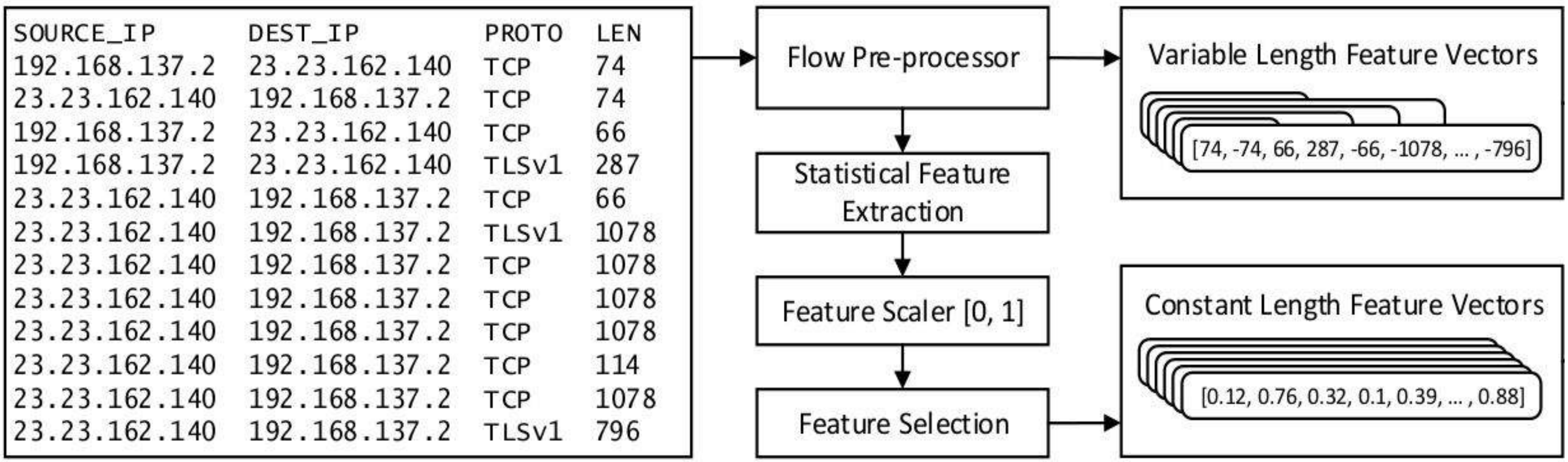


SPRITZ Security & Privacy Research Group
Università degli Studi di Padova

Building the dataset

TCP Packets captured

Per Flow approach (1)



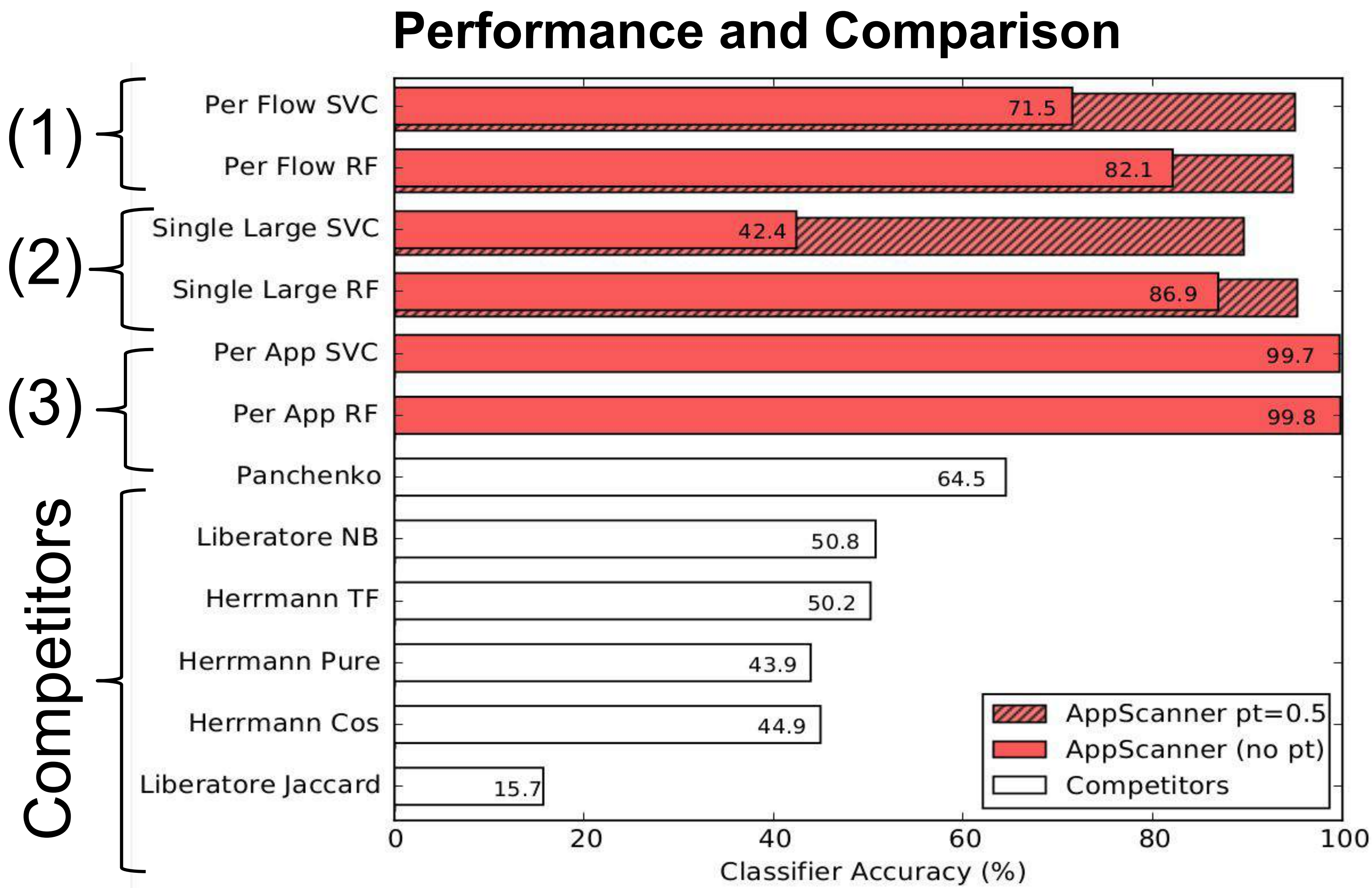
Statistical approaches (2, 3)

SPRITZ Security & Privacy Research Group
Università degli Studi di Padova

Improving the accuracy of AppScanner

- Classification performed on **each** network traffic flow
- We aim to identify an app →many flows available
- Flow →Classifier prediction → (App, Probability of prediction)
- Applying a **probability threshold** (PT)
  - Filter out flows with **uncertain predictions**
  - Increase classification accuracy tuning PT

AppScanner (IEEE EuroS&P '16)



Outline

- Covert and Side Channels 101

- Network Traffic Analysis
  - *As a side channel: app and sensitive data inference*
- **Energy Consumption**
  - ***As a side channel: user and app inference***
  - *As a covert channel: data exfiltration*

- Device Movement
  - *As a side channel: smartphone user authentication*
  - *Attacks against biometric authentication*
- Keystroke Timing
  - *As a side channel: text typed on keyboards*
- Acoustic Emanations
  - *As a side channel: text typed on keyboards*

M. Conti, M. Nati, E. Rotundo, R. Spolaor.

Mind The Plug! Laptop-User Recognition Through Power Consumption.

In ACM AsiaCCS 2016 workshop IoTPTS 2016

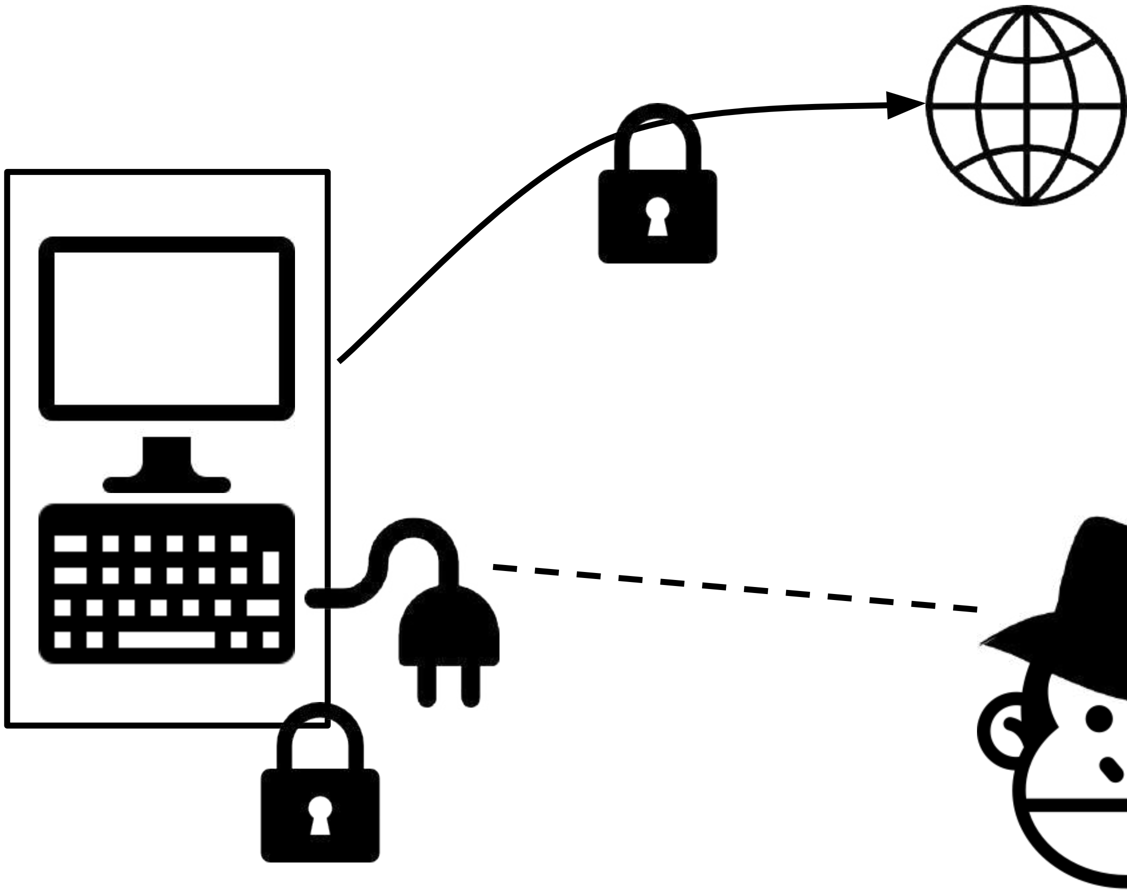
Power Consumption Side Channel

Power consumption

Can reveal what we are doing!

Device drains different power depending on our actions

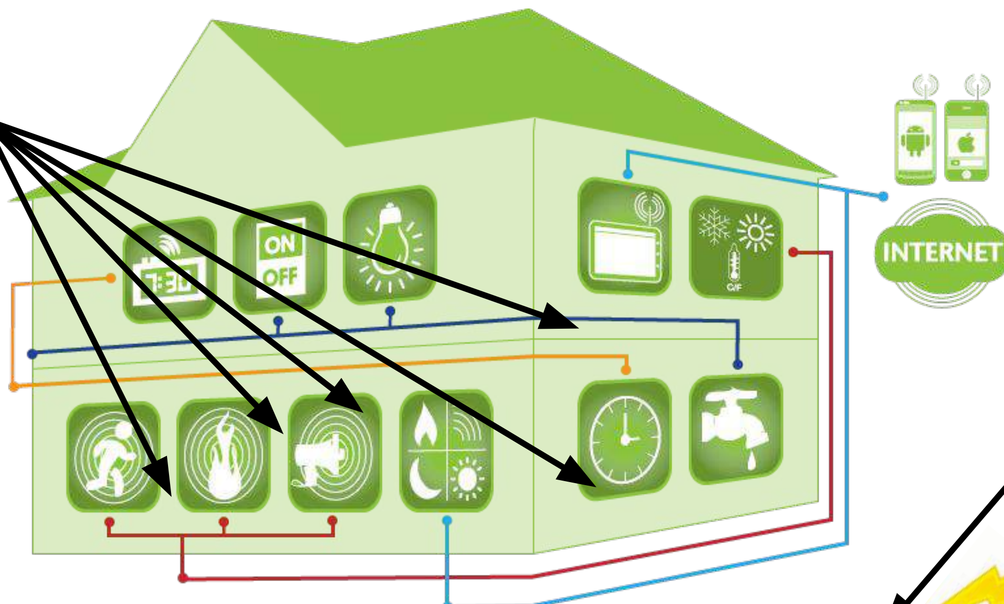
Works on **laptops** and **mobile**



Smartbuilding

Internet of Things applied not only to industry, but also to buildings, such as houses and **offices**

Wall-socket level sensors



household level sensors

Smartgrid

Wall-socket smartmeters

- Smartmeters are able to measure the electric quantities of the plugged appliances
    - **Reactive Power**
    - **RMS Current**
    - **Voltage**
    - **Phase**
- IoT testbed in University of Surrey (UK)
- Limitation:
    - only **1Hz** of sampling rate

Definition of "Laptop-User"

A **Laptop-user** is made of the **combination** of:

- Laptop
- Software installed and running
- User behavior

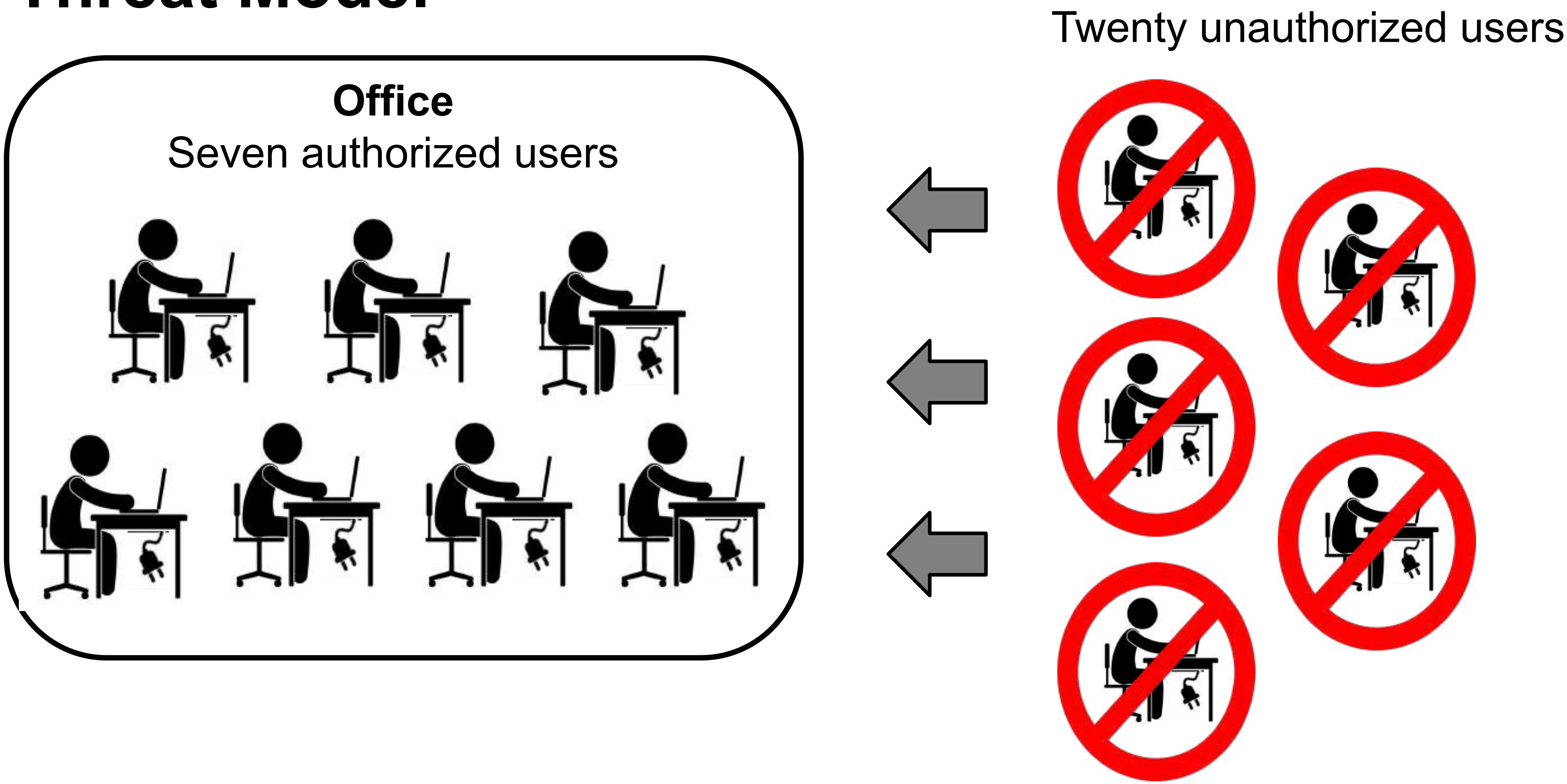
Goal & Motivation

Is it possible to recognize a **Laptop-user** from its energy consumption?

This can bring:

- **Benefit on smartbuilding automation**,
  - context-aware environments can automatically adjust and trigger predefined actions or services
    - e.g., according to the presence of a specific user
  - Detect un-authorized users
- **Threat to user privacy**,
  - it is possible to locate and trace a user

Threat Model

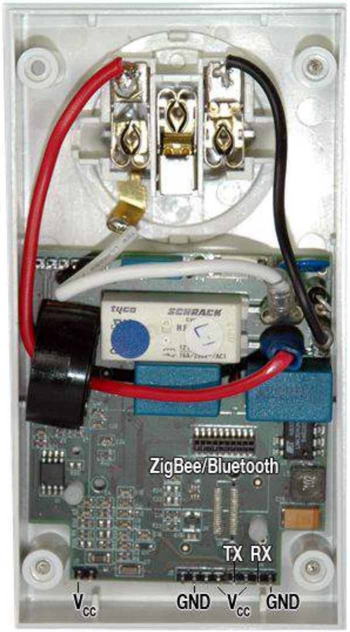


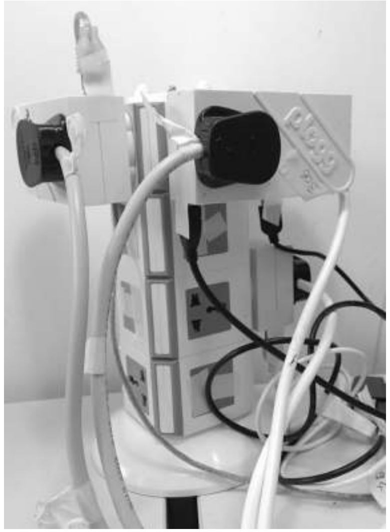
We aim to:

- Recognize whether the user is in the “authorized” set
- Identify the specific user in the “authorized” set

Laptop-users Recognition

Multiclass classification (8 classes)
- The **seven authorized** laptop-users
- The **intruders** (as a single class)





Classification in three steps:
1. 10-fold cross validation for **parameters selection**
2. Performance **evaluation** on a disjoint test set
3. Classification **validation**

Mind the Plug! (IoTPTS @AsiaCCS ’16)

SPRITZ
Security & Privacy
Research Group

Università
degli Studi
di Padova

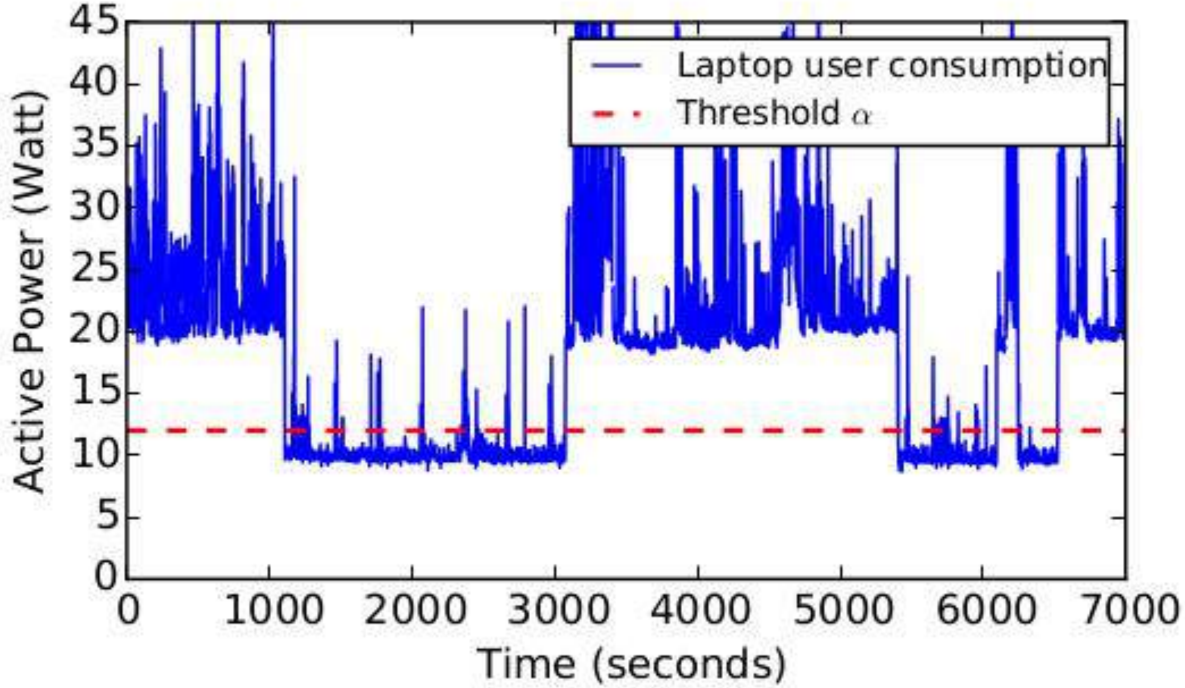
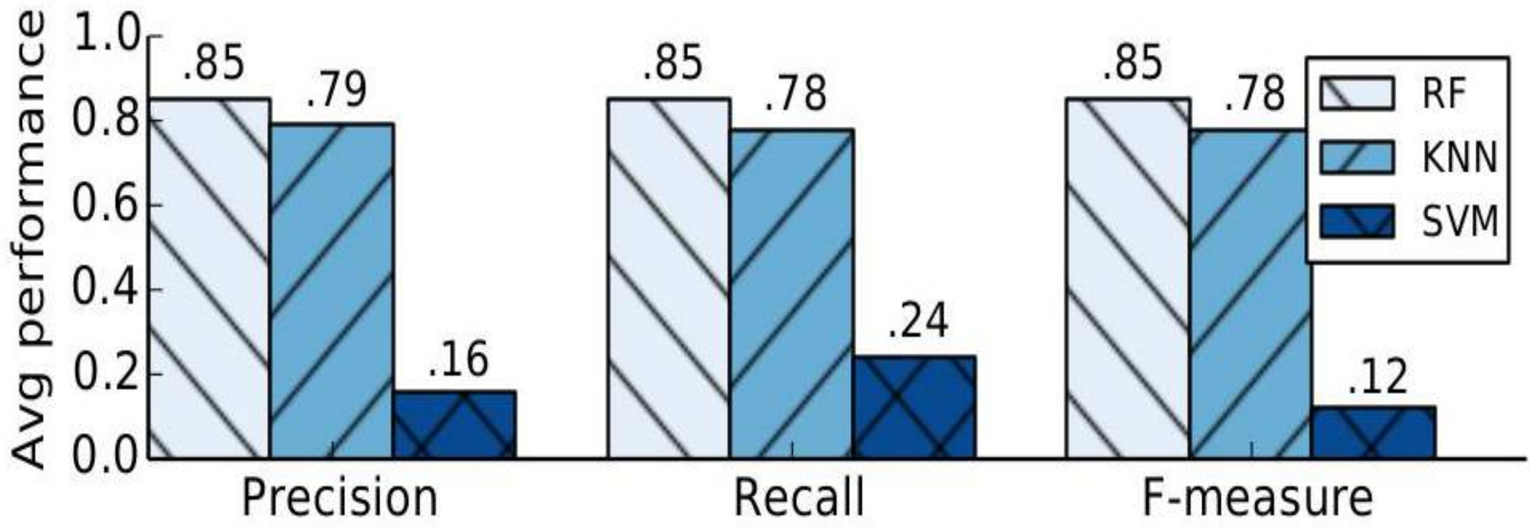


Figure 2: Example of *Active Power* trace (continuous blue line) and the lower-cutting threshold $\alpha = 12$ Watt (dashed red line). Samples under α are low-energy timespans in which the user does not use the laptop.

Mind the Plug! (IoTPTS @AsiaCCS ’16)



85% of F-measure with Random Forest classifier

Classification validation

Classifiers label all segments in the testset

- Bad for False Positive rate (FPR)

We can leverage also the prediction probability

- Since classifiers output also their **confidence**

Tuning prediction probability threshold

- It can reduce False Positives

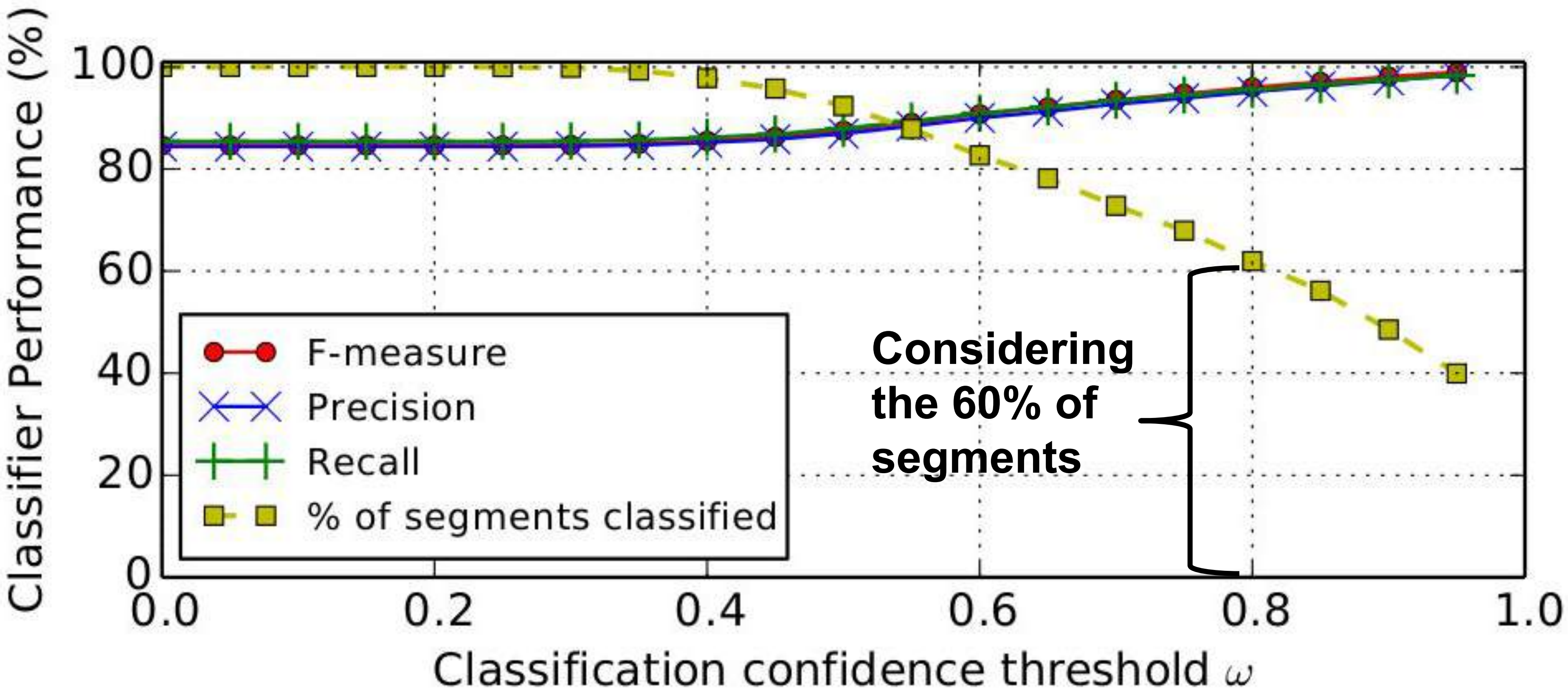
Other implications:

- MTPlug can be more conservative
- May take more segments to identify some laptop-user

SPRITZ Security & Privacy Research Group

Università degli Studi di Padova

Classification validation results



Considering the 60% of segments

Limitations and Future work

Structural limitation: The plogg wall-socket sensors have a low sampling rate
Solution: Adopt a new generation wall-socket sensors

Data limitation: we collected data of seven users (office)
Solution: Collect more data in order to assess the feasibility of authentication system based on energy consumption

Outline

- Covert and Side Channels 101

- Network Traffic Analysis
  - *As a side channel: app and sensitive data inference*
- **<u>Energy Consumption</u>**
  - *As a side channel: user and app inference*
  - ***<u>As a covert channel: data exfiltration</u>***

- Device Movement
  - *As a side channel: smartphone user authentication*
  - *Attacks against biometric authentication*
- Keystroke Timing
  - *As a side channel: text typed on keyboards*
- Acoustic Emanations
  - *As a side channel: text typed on keyboards*

R Spolaor, L Abudahi, V Moonsamy, M Conti, R Poovendran.

No Free Charge Theorem: a Covert Channel via USB Charging Cable on Mobile Devices.

In ACNS 2017

Presented at Black Hat Europe 2018

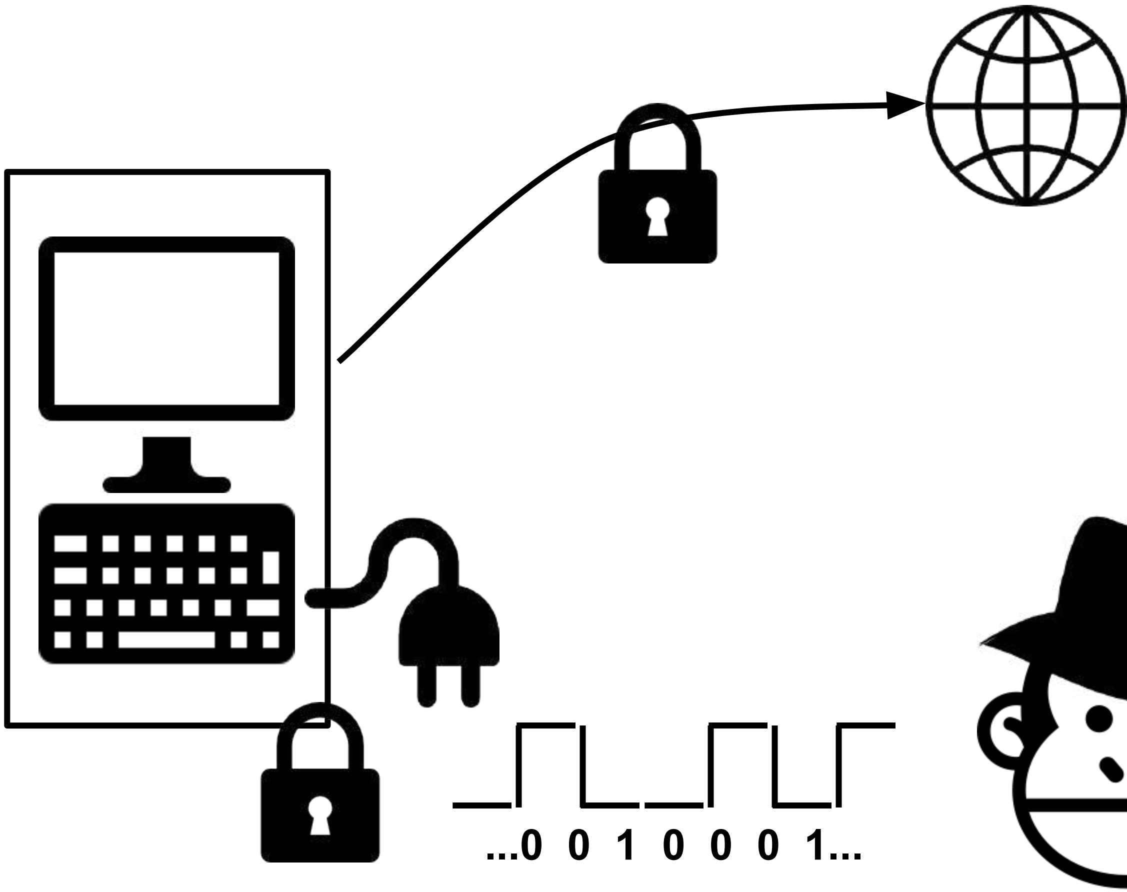
SPRITZ Security & Privacy Research Group
Università degli Studi di Padova

Power consumption

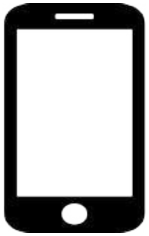
Can be used as a covert channel

Malware makes device drain more/less power to communicate with a **malicious power outlet**

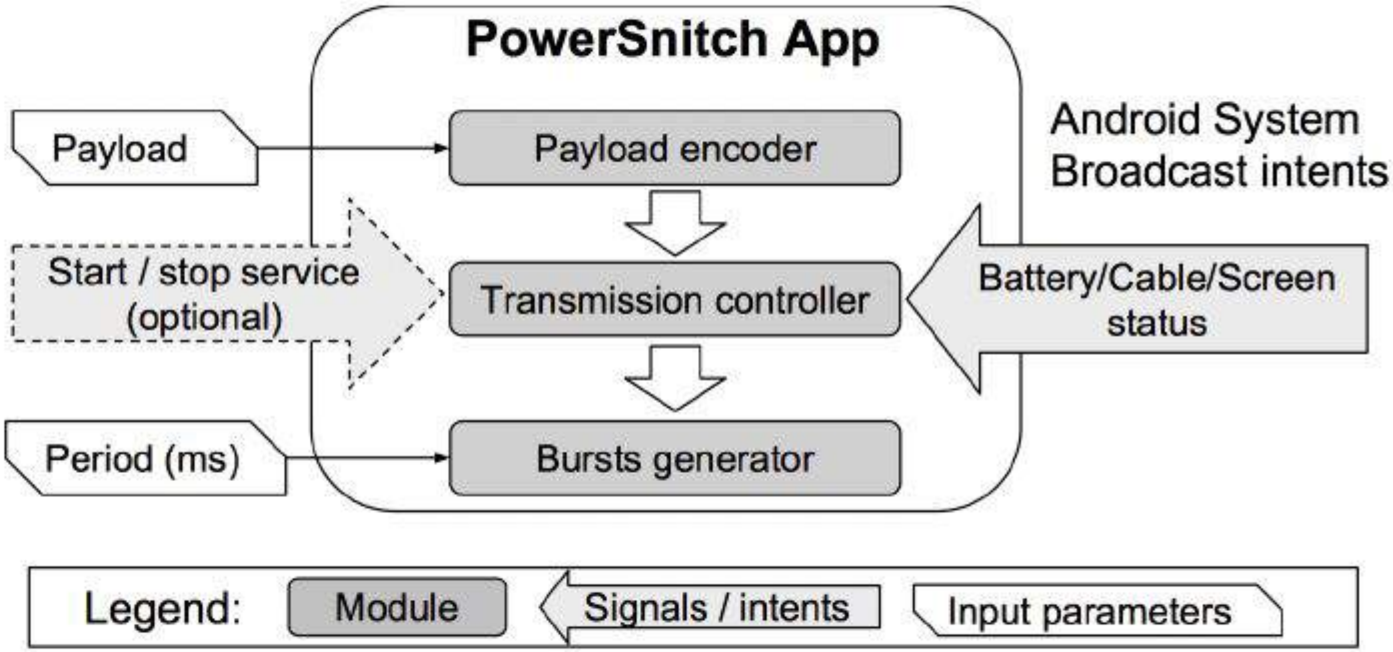
Thus **exfiltrating secrets**

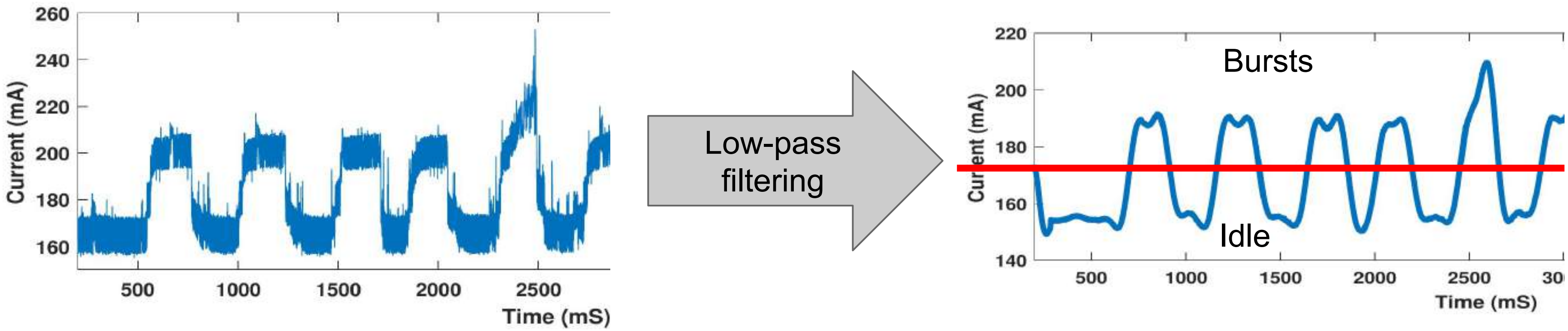


No Free Charge Theorem: a Covert Channel via USB Charging Cable on Mobile Devices



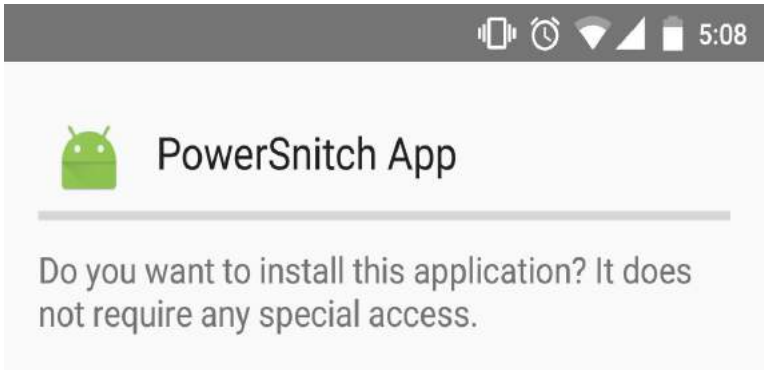
PowerSnitch Application





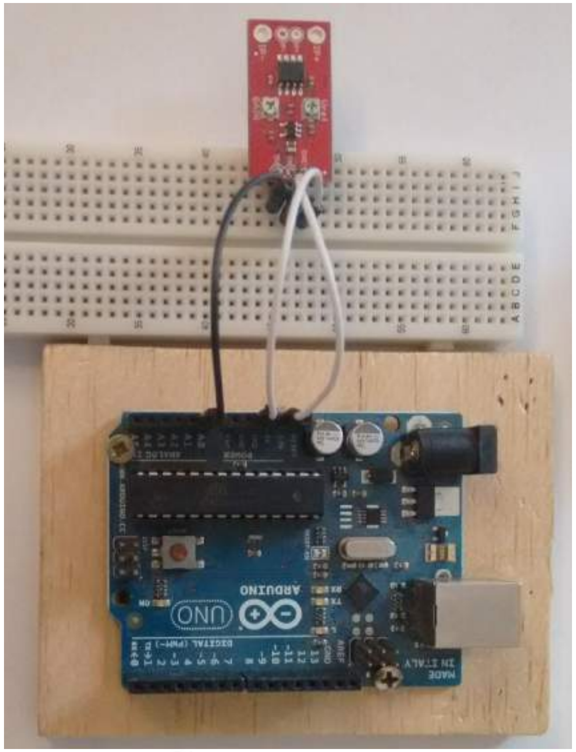
Results in terms of Bit Error Ratio (BER)

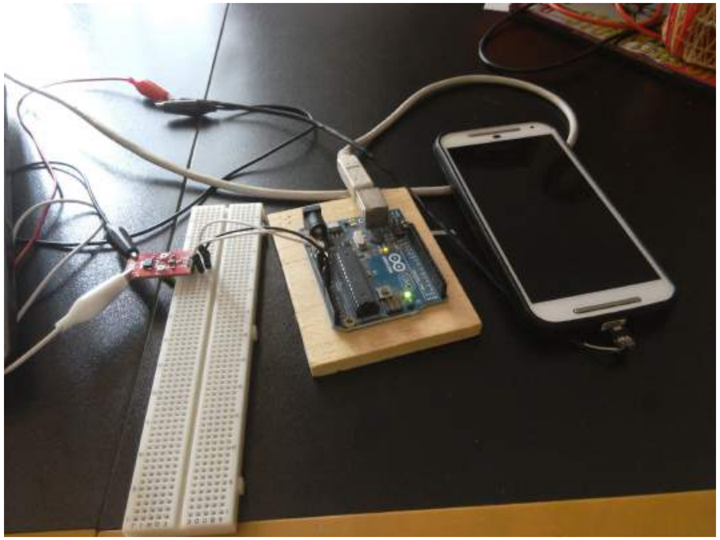
Device	Period (milliseconds)					
	1000	900	800	700	600	500
Nexus 4	13.5	0.78	0.0	0.0	13.33	16.21
Nexus 5	21.0	0.0	0.95	36.82	40.35	13.4
Nexus 6	1.07	0.0	0.21	0.0	4.05	7.42
Samsung S5	12.5	13.5	13.31	16.33	17.9	21.42

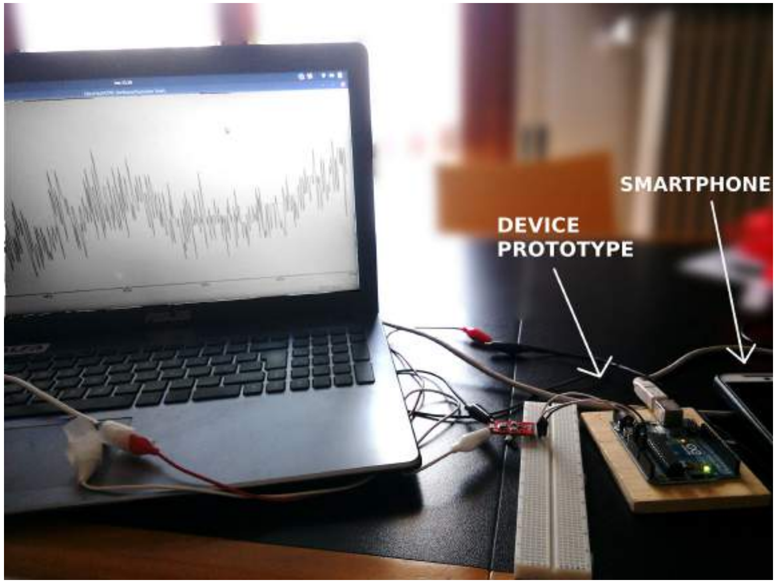


PowerSnitch app does not require any permission !!!

Power Bank Prototype







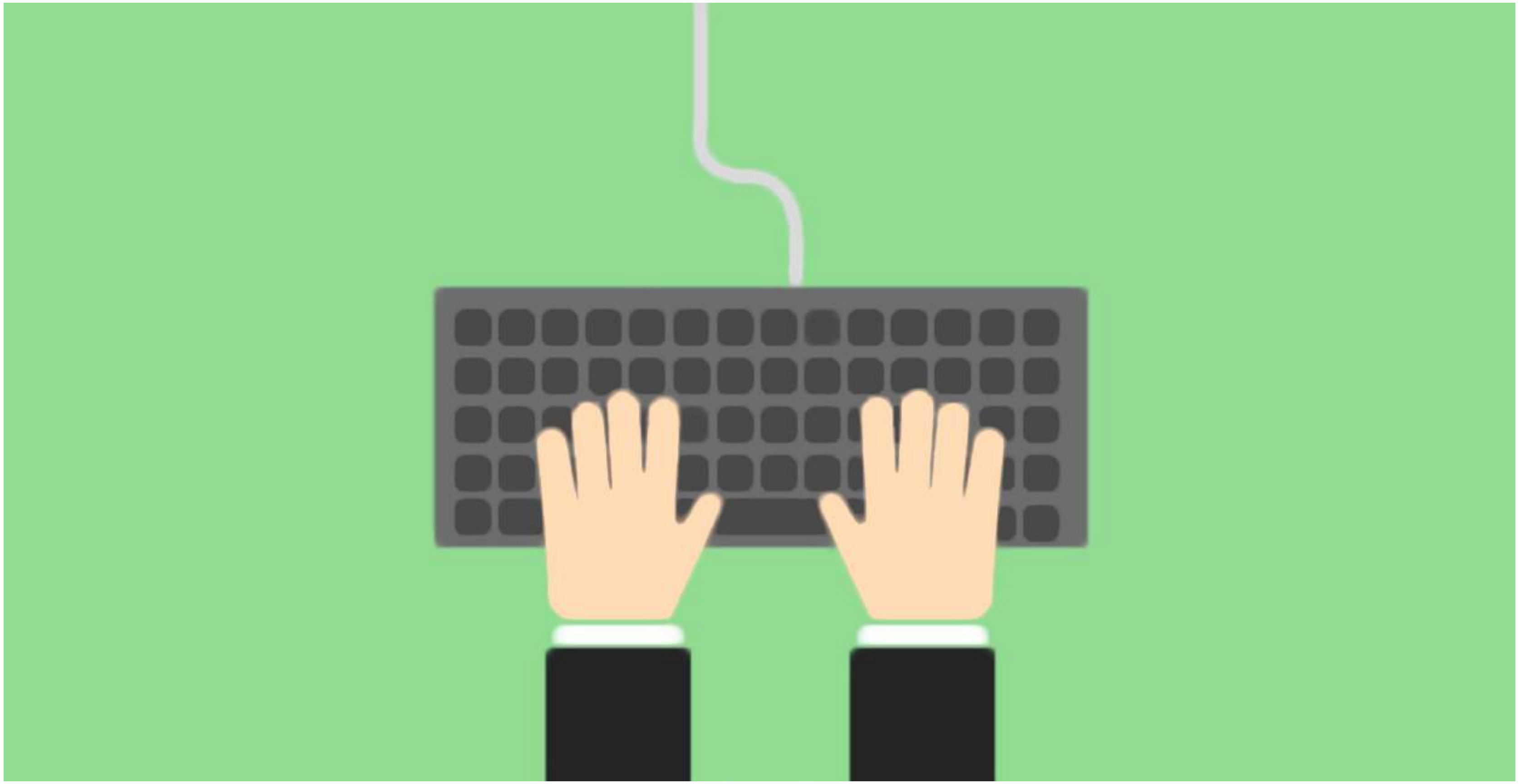
Outline

- Covert and Side Channels 101

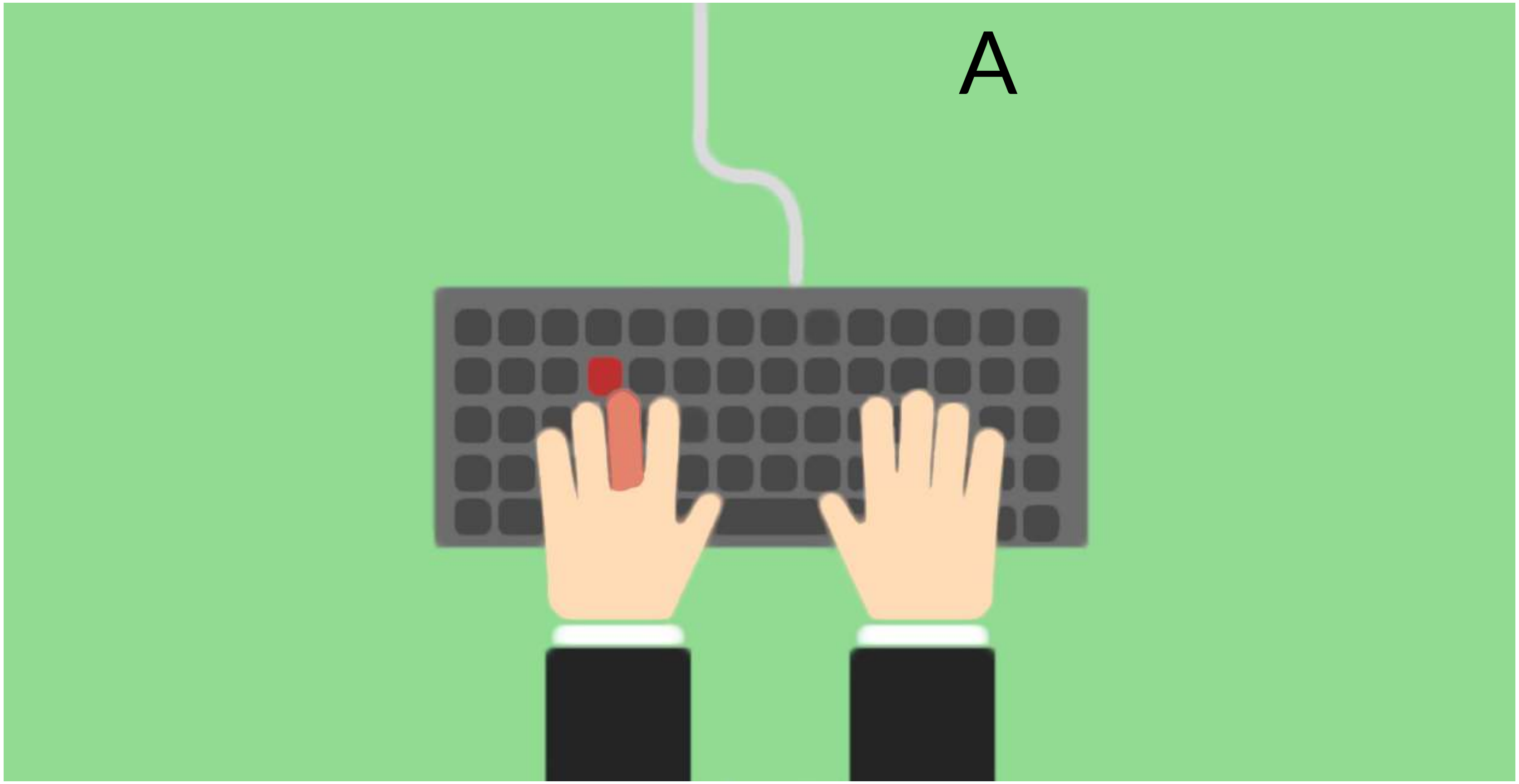
- Network Traffic Analysis
  - *As a side channel: app and sensitive data inference*
- Energy Consumption
  - *As a side channel: user and app inference*
  - *As a covert channel: data exfiltration*

- **Device Movement**
  - ***As a side channel: smartphone user authentication***
  - *Attacks against biometric authentication*
- Keystroke Timing
  - *As a side channel: text typed on keyboards*
- Acoustic Emanations
  - *As a side channel: text typed on keyboards*

Keystroke Dynamics 101



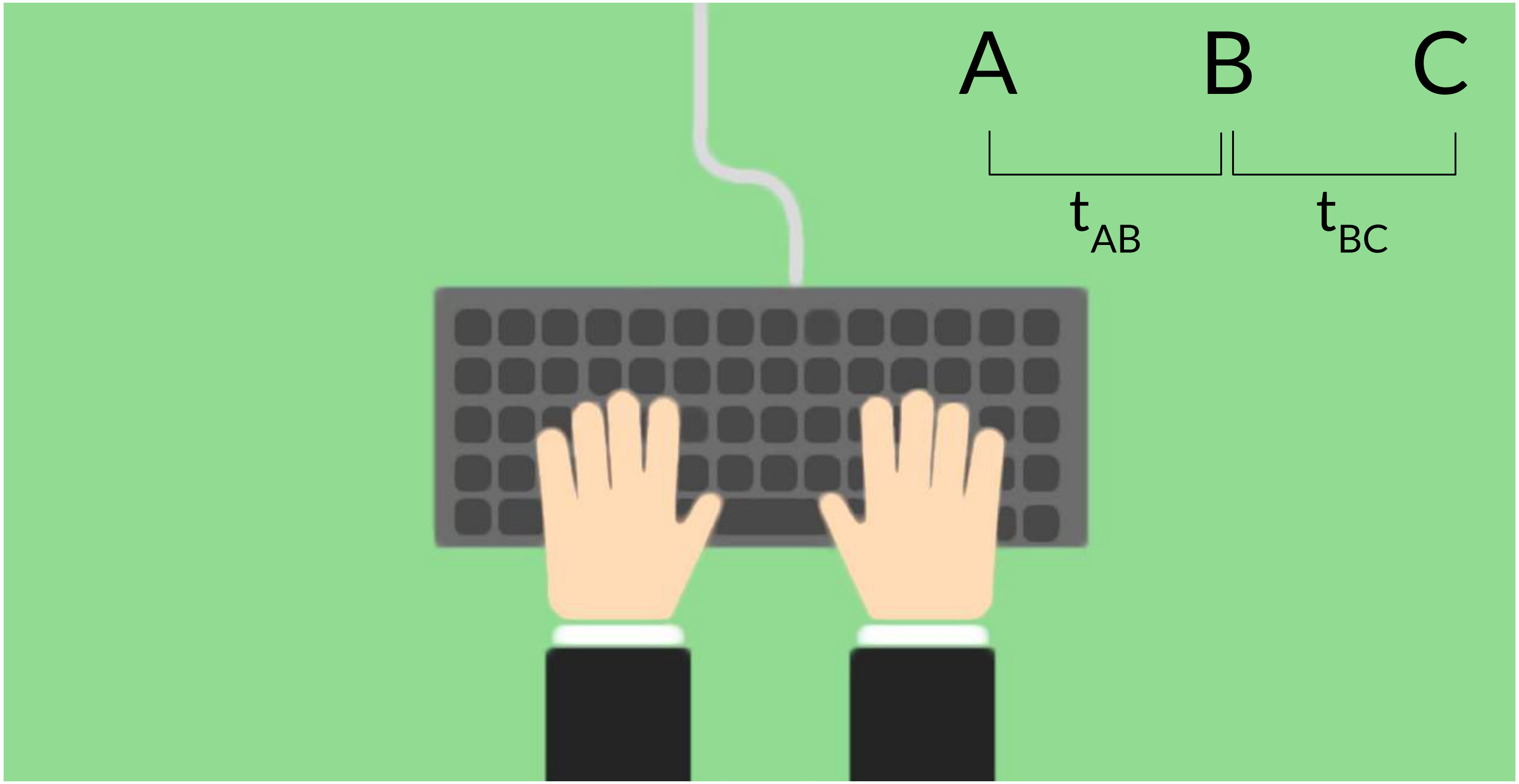
Keystroke Dynamics 101



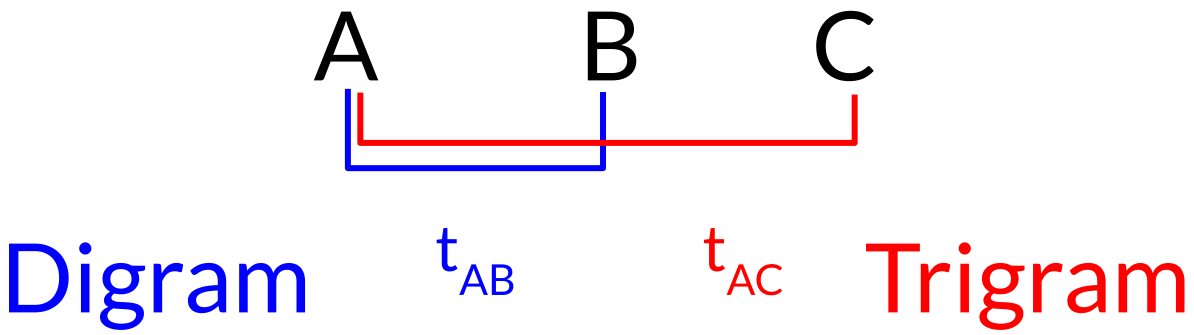
Keystroke Dynamics 101

Keystroke Dynamics 101

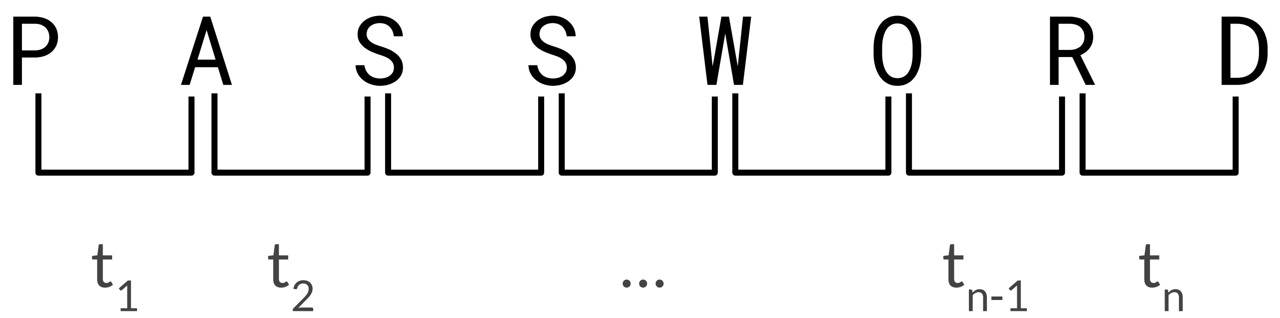
Keystroke Dynamics 101



Keystroke Dynamics 101



Keystroke Dynamics 101



- Inter-keystroke times as a personal *signature*
- Used as biometric in authentication systems

Kamil Majdanik, Cristiano Giuffrida, Mauro Conti, Herbert Bos.

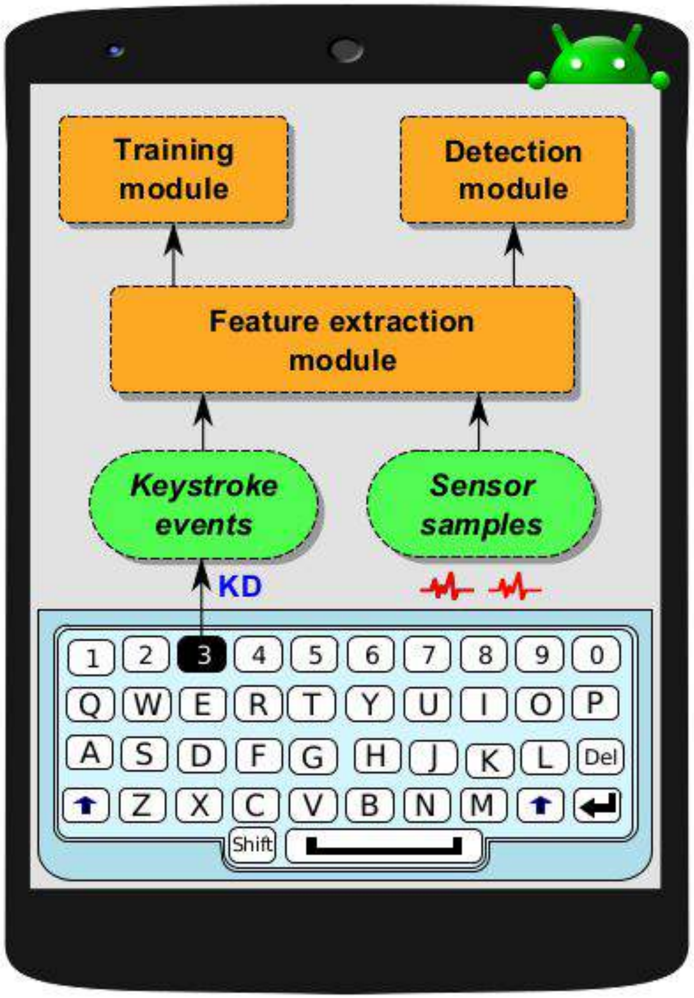
I Sensed It Was You: Authenticating Mobile Users with Sensor-enhanced Keystroke Dynamics.

In DIMVA 2014

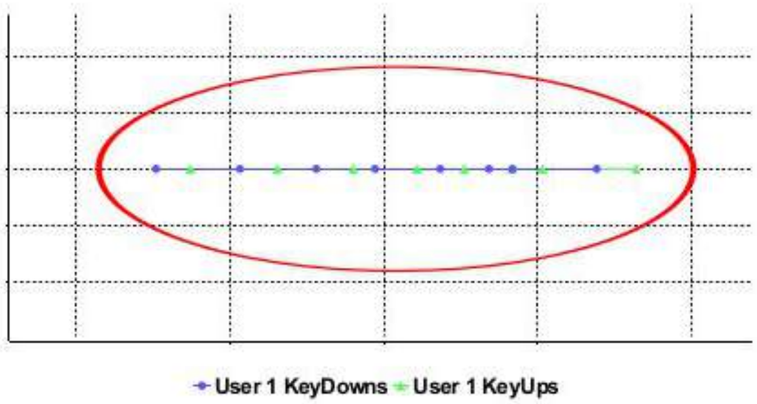
Our system: Unagi

User authentication with
Sensor enhanced
Keystroke Dynamics

Scenario: User typing 'HELLO'

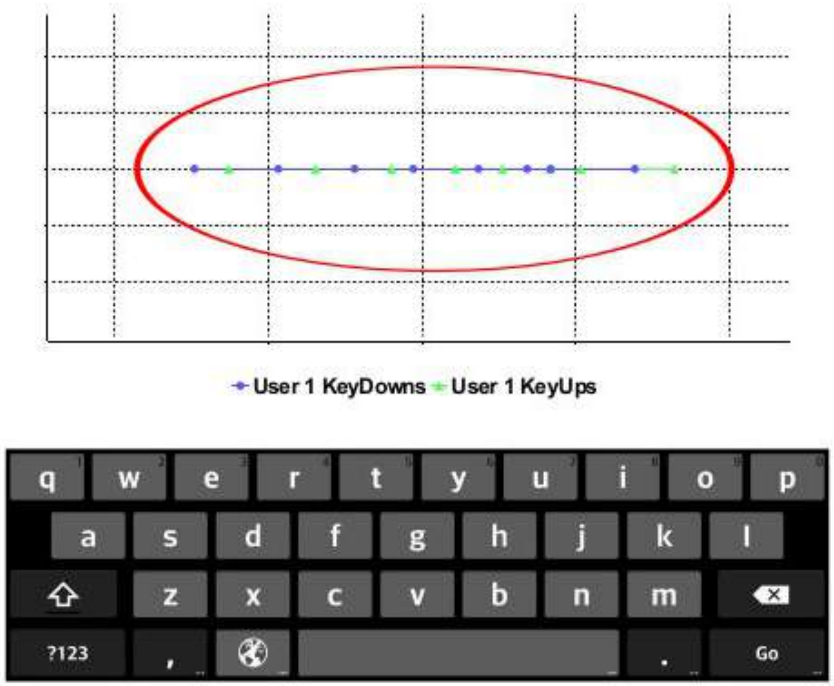


I Sensed It Was You

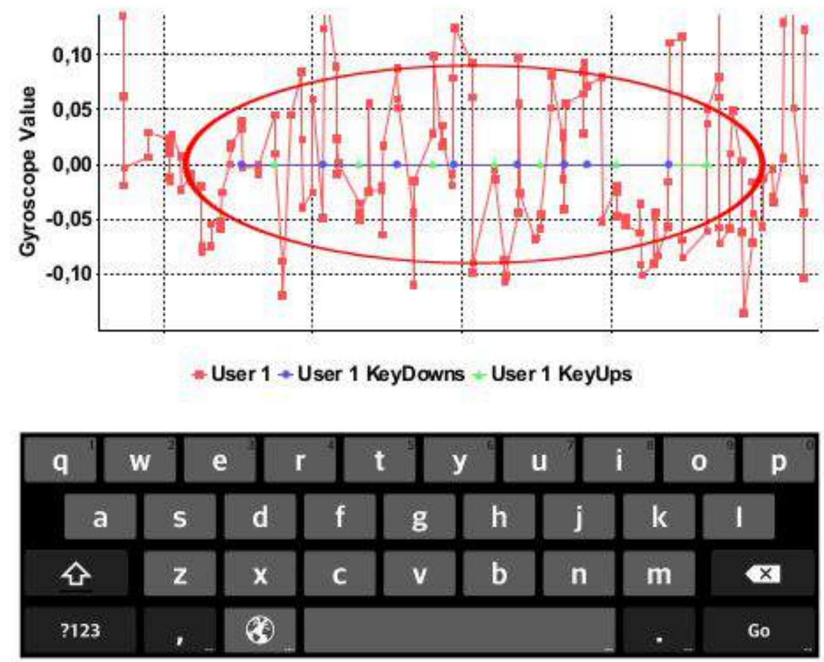


Keystroke dynamics

I Sensed It Was You



Keystroke dynamics

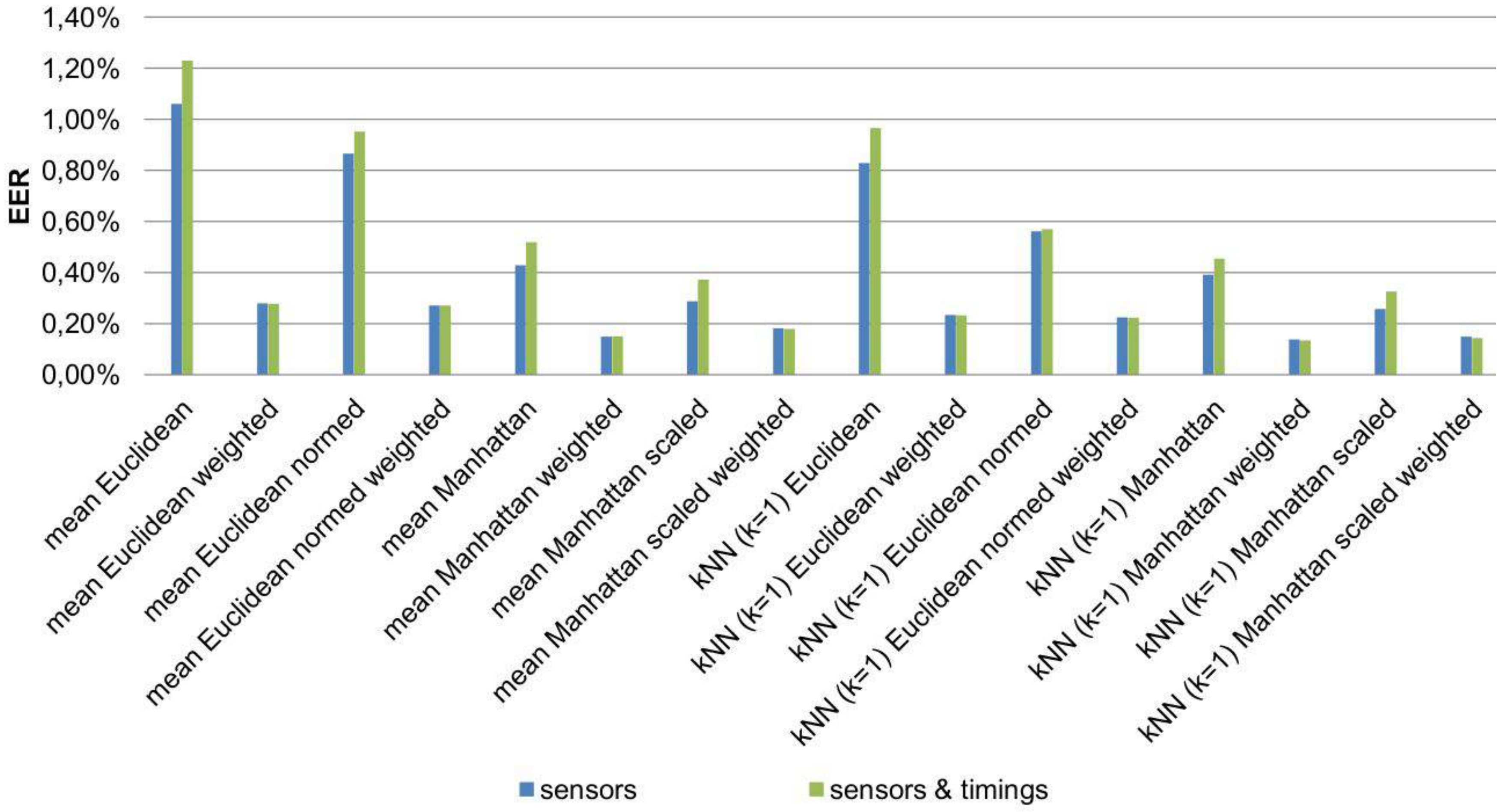


Sensor-enhanced keystroke dynamics

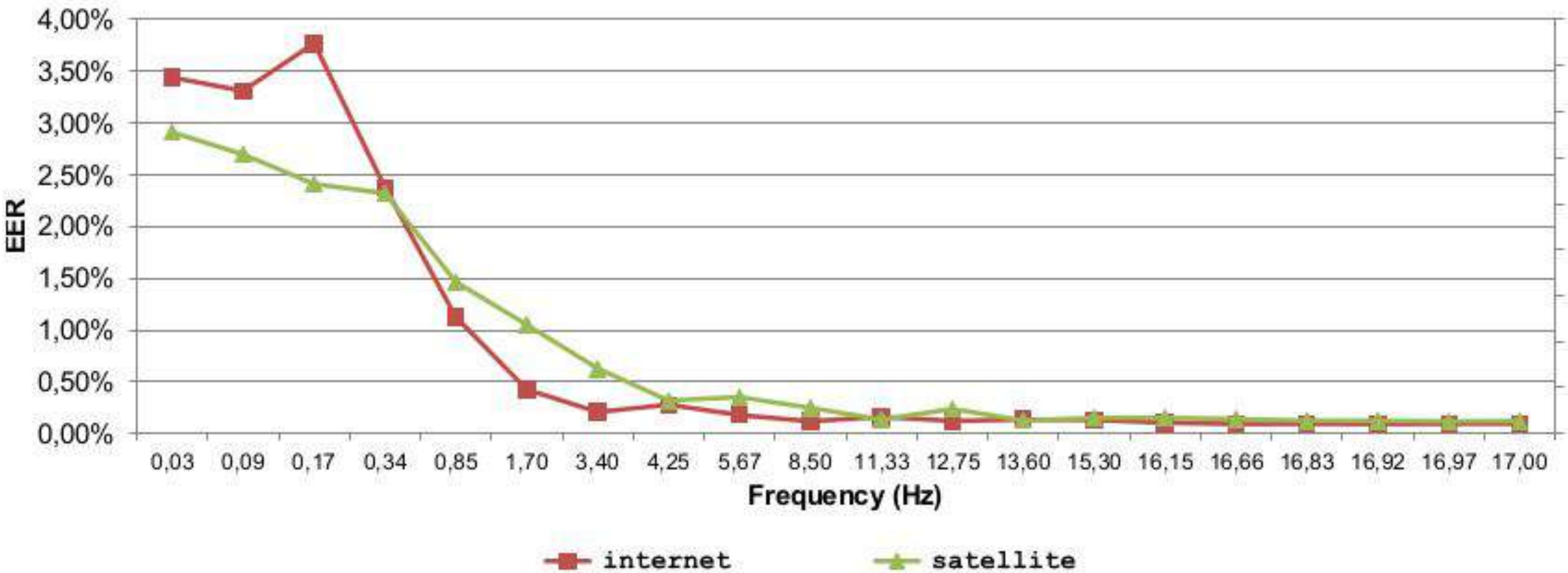
SPRITZ Security & Privacy Research Group

Università degli Studi di Padova

Accuracy (EER) for different considered algorithms



Accuracy vs. Sensors Sampling Frequency



EER - Equal Error Rate (rate at which both acceptance and rejection errors are equal)

Key Results

- Movement sensors are suitable for biometric authentication
- Sensors can dramatically enhance keystroke dynamics accuracy
- Effective even with short passwords and low sampling frequencies

Future work

- Applicability to free-text authentication
- Robustness against statistical attacks

Outline

SPRITZ Security & Privacy Research Group
Università degli Studi di Padova

- Covert and Side Channels 101

- Network Traffic Analysis
  - *As a side channel: app and sensitive data inference*
- Energy Consumption
  - *As a side channel: user and app inference*
  - *As a covert channel: data exfiltration*

- **<u>Device Movement</u>**
  - *As a side channel: smartphone user authentication*
  - ***<u>Attacks against biometric authentication</u>***
- Keystroke Timing
  - *As a side channel: text typed on keyboards*
- Acoustic Emanations
  - *As a side channel: text typed on keyboards*

V. D. Stanciu, R. Spolaor, M. Conti, C. Giuffrida

On the Effectiveness of Sensor-enhanced Keystroke Dynamics Against Statistical Attacks

in ACM CODASPY 2016

Previous work - Drawbacks

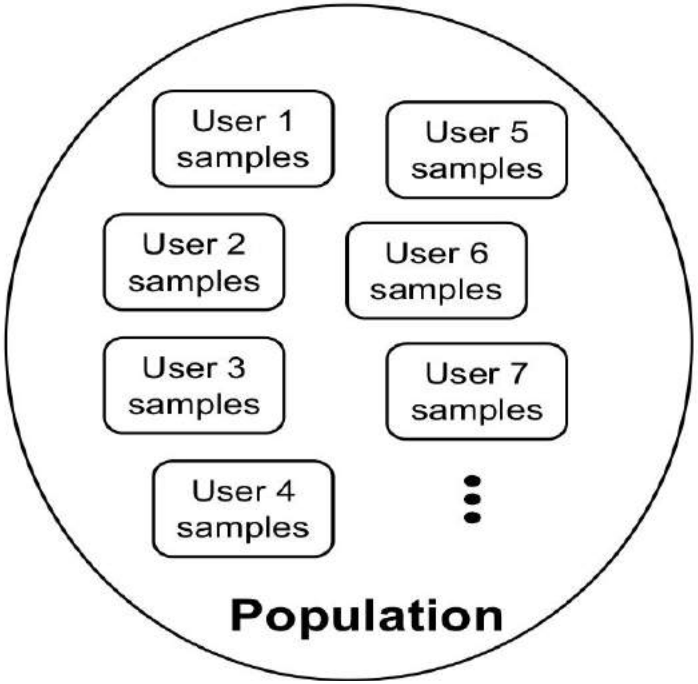
The previous **behavioral biometric authentication** system relies on:

- Secret of the password
- **Keystroke dynamics** (touch gestures)
- **Accelerometer** and **Gyroscope** sensors data

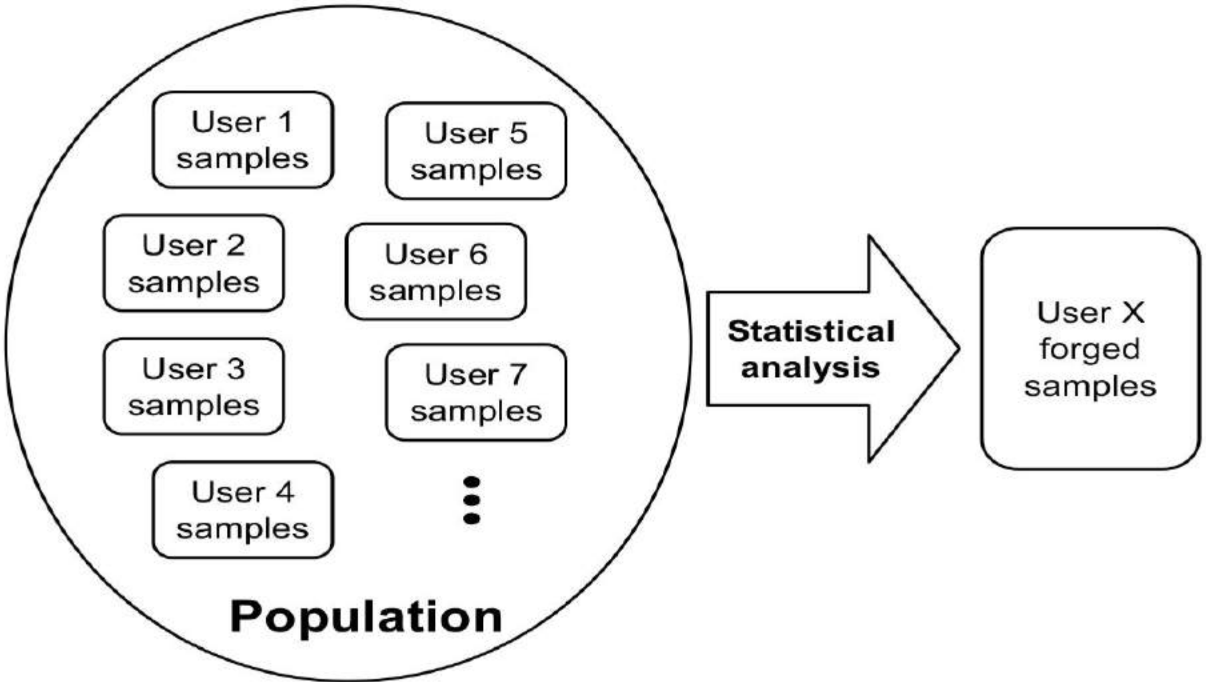
Previous work: we used kNN (with k=1) and mean values combined with several metrics (e.g., euclidean, Manhattan)

Question: is our system resilient to **Statistical attacks**?

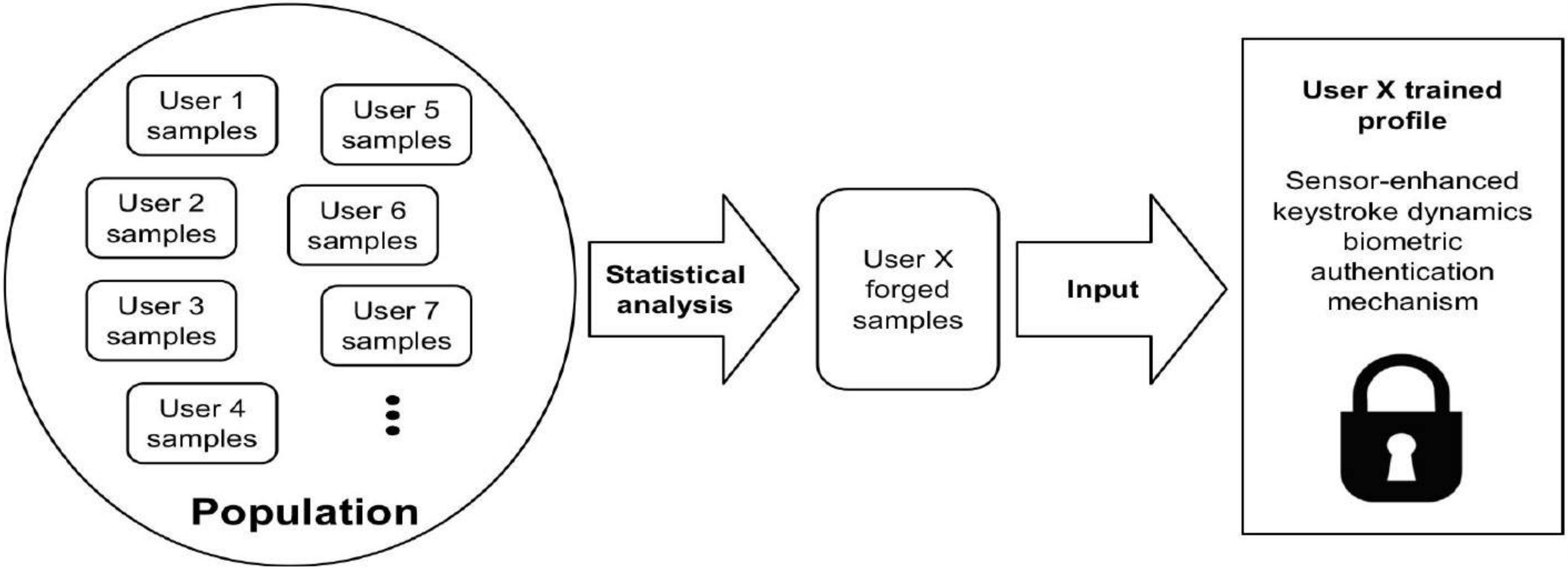
Statistical Attack



Statistical Attack



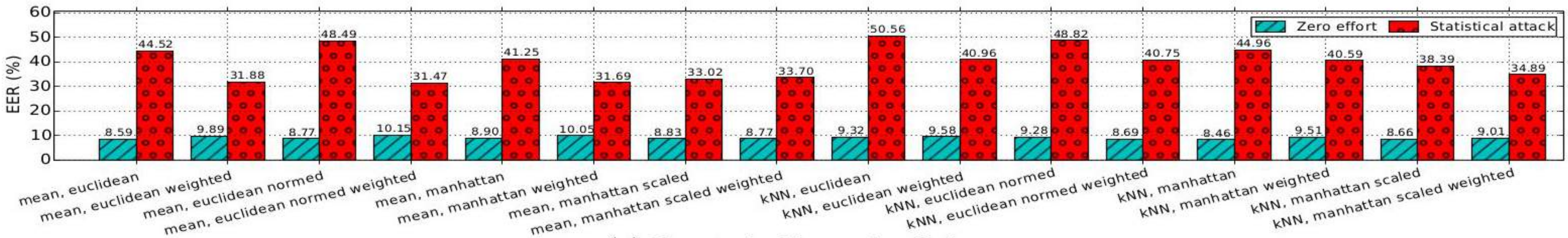
Statistical Attack



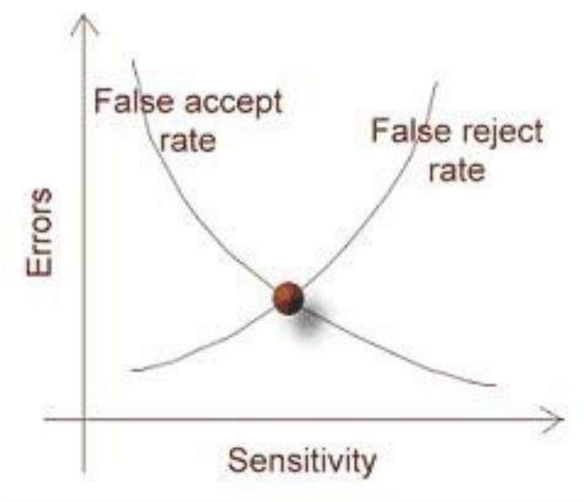
SPRITZ Security & Privacy Research Group

Università degli Studi di Padova

low Equal Error Rate (EER) == accurate authentication method



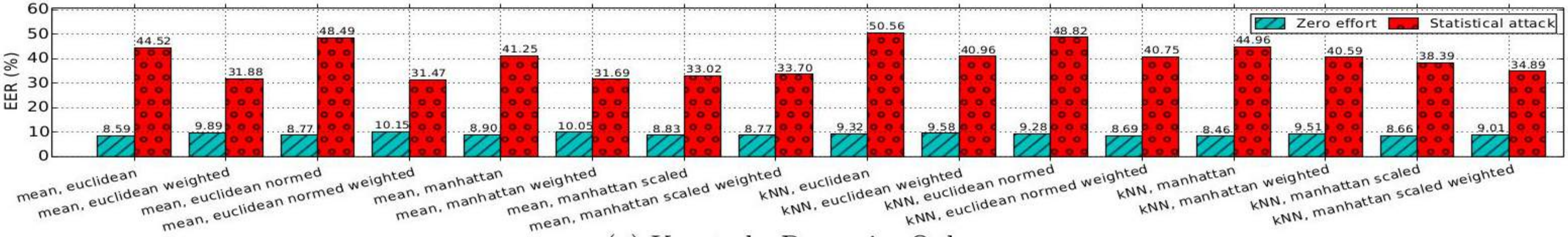
(a) Keystroke-Dynamics Only.



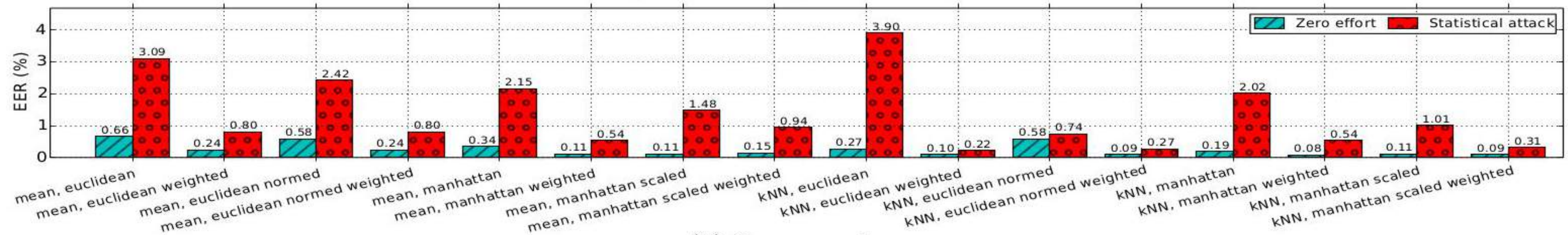
SPRITZ Security & Privacy Research Group

Università degli Studi di Padova

low Equal Error Rate (EER) == accurate authentication method



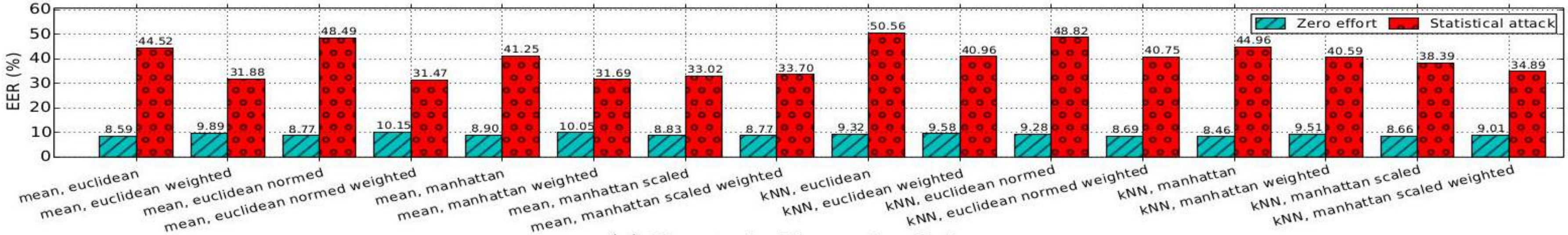
(a) Keystroke-Dynamics Only.



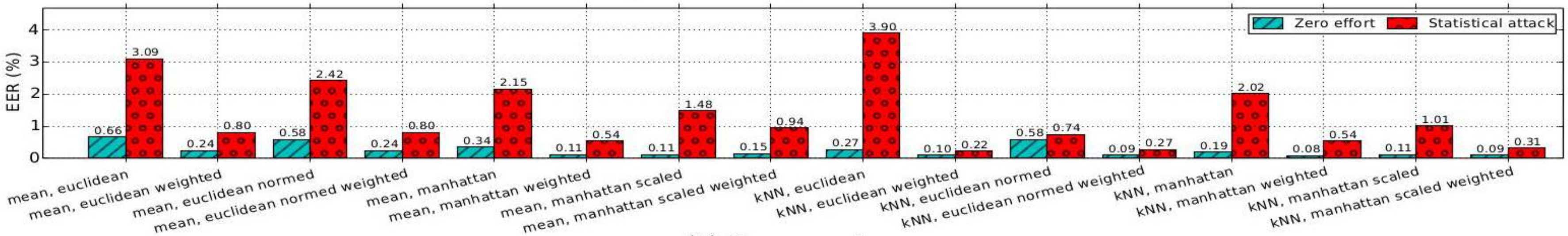
(b) Sensors only.

Results

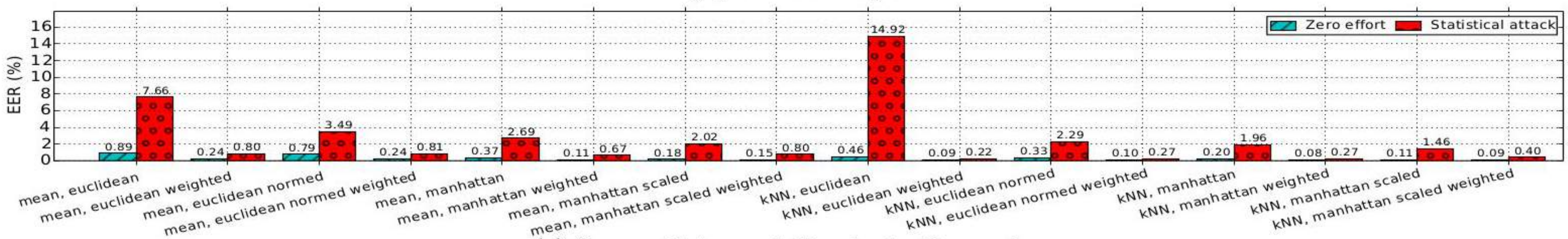
low Equal Error Rate (EER) == accurate authentication method



(a) Keystroke-Dynamics Only.



(b) Sensors only.



(c) Sensor-Enhanced Keystroke-Dynamics.

Outline

- Covert and Side Channels 101

- Network Traffic Analysis
  - *As a side channel: app and sensitive data inference*
- Energy Consumption
  - *As a side channel: user and app inference*
  - *As a covert channel: data exfiltration*

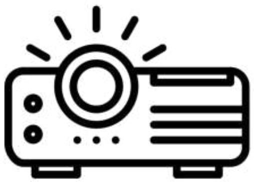
- Device Movement
  - *As a side channel: smartphone user authentication*
  - *Attacks against biometric authentication*
- **Keystroke Timing**
  - ***As a side channel: text typed on keyboards***
- Acoustic Emanations
  - *As a side channel: text typed on keyboards*

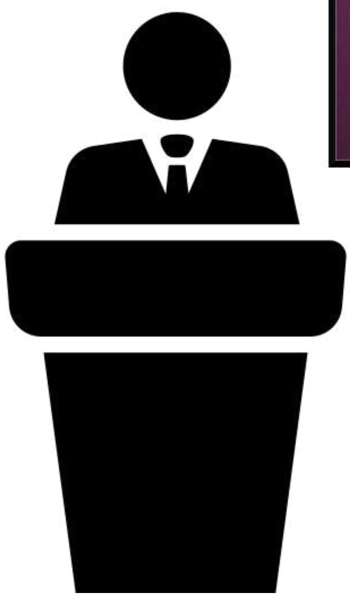
Kiran Balagani, Mauro Conti, Paolo Gasti, Martin Georgiev, Tristan Gurtler, Daniele Lain, Charissa Miller, Kendall Molas, Nikita Samarin, Eugen Saraci, Gene Tsudik, Lynn Wu

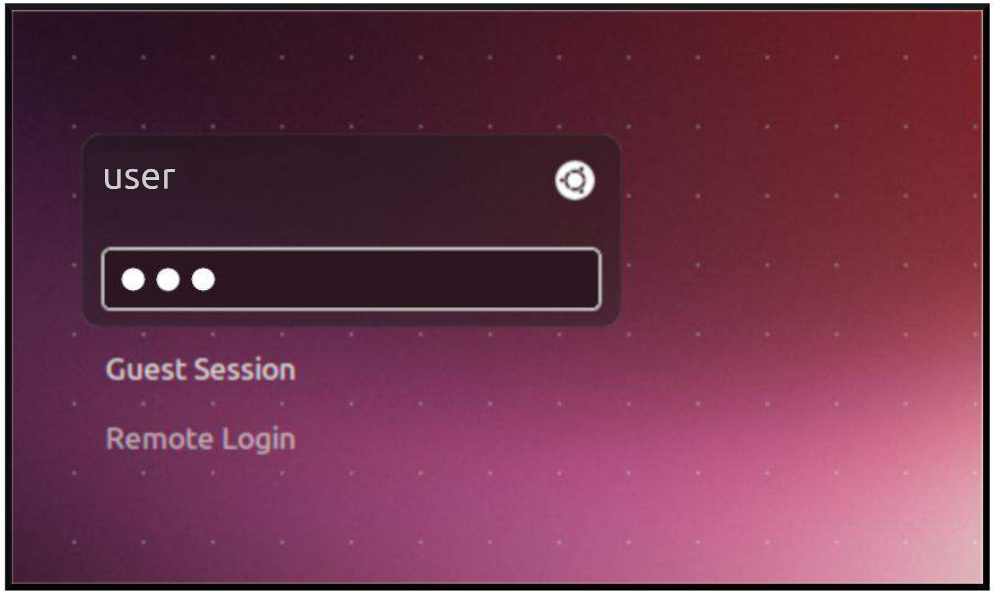
SILK-TV: Secret Information Leakage From Keystroke Timing Videos.

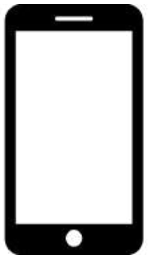
In ESORICS 2018

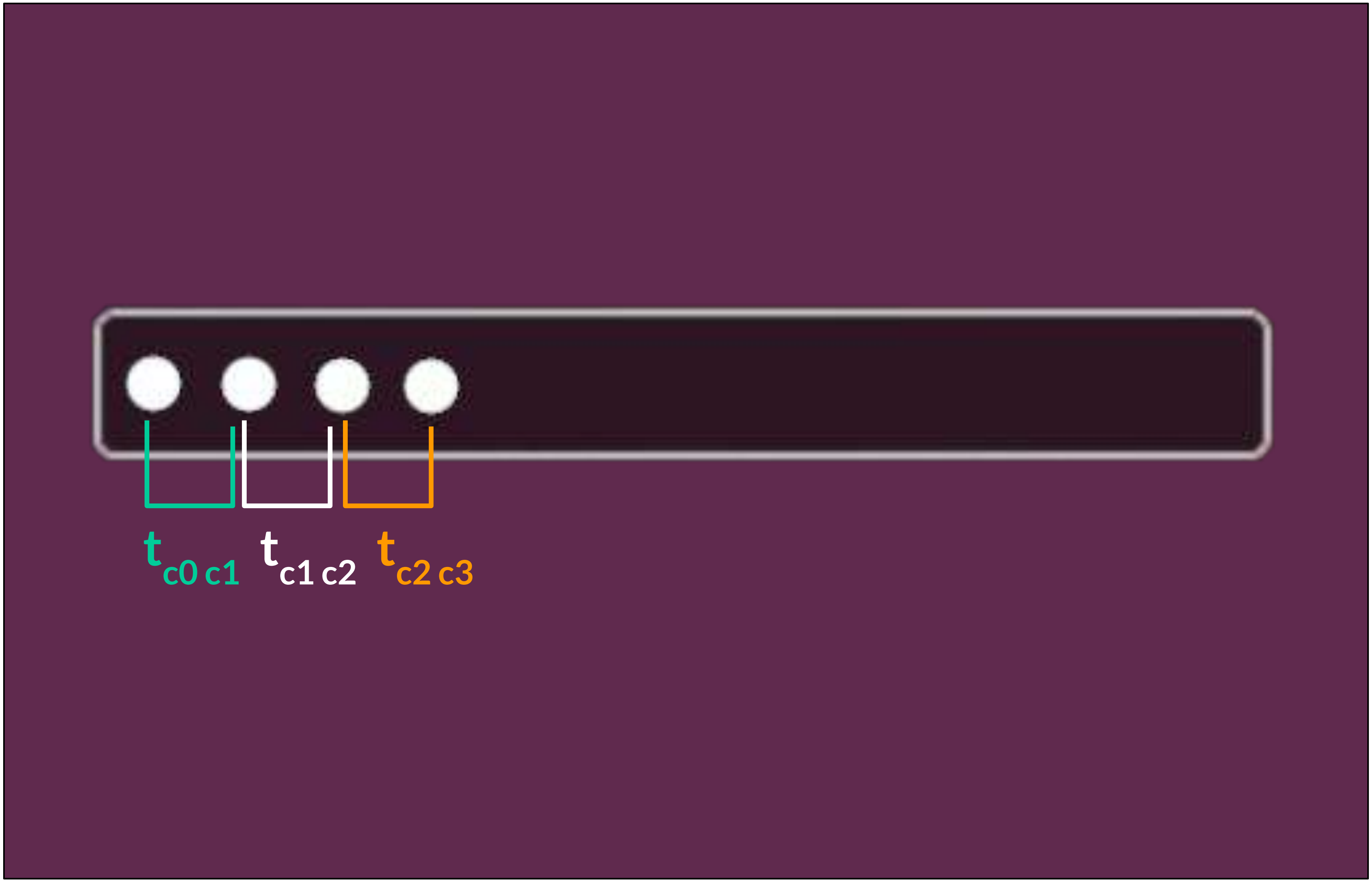
Timing Information Leak - 1





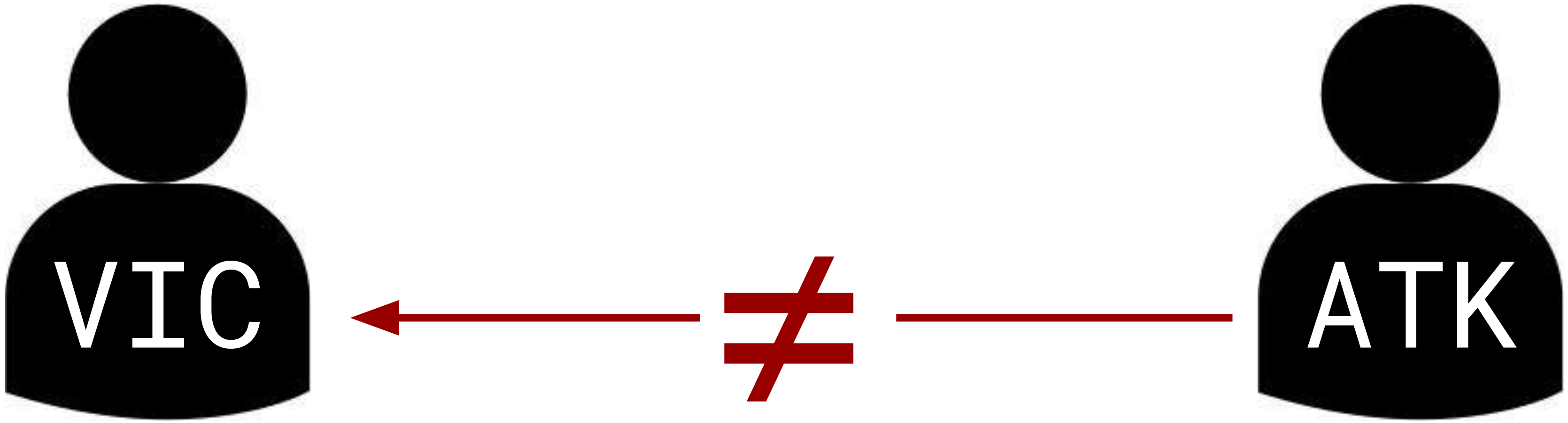


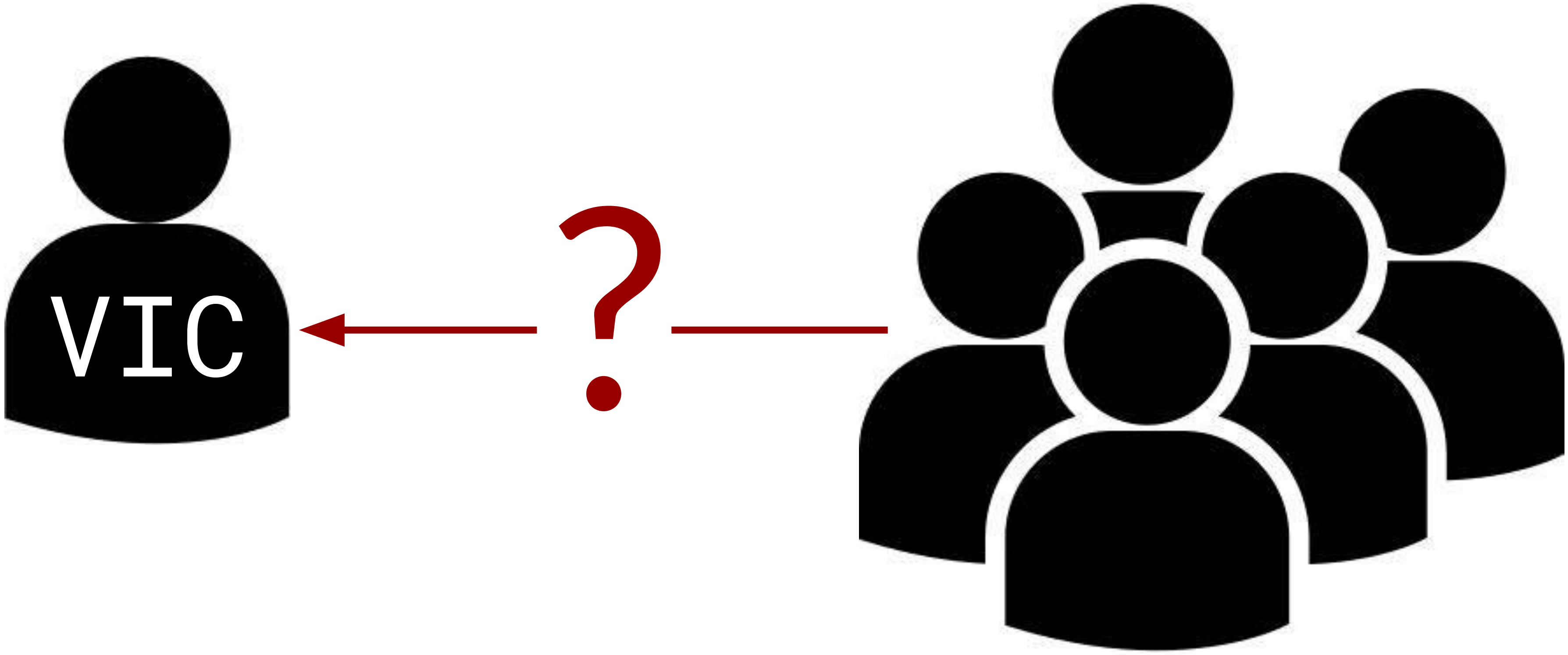



$t_{c0\ c1}$
$t_{c1\ c2}$
$t_{c2\ c3}$

Timing Information Leak - 2

Keypad not visible - but the screen is!


VIC
≠
ATK

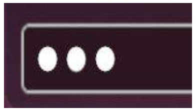

VIC
?

Contributions

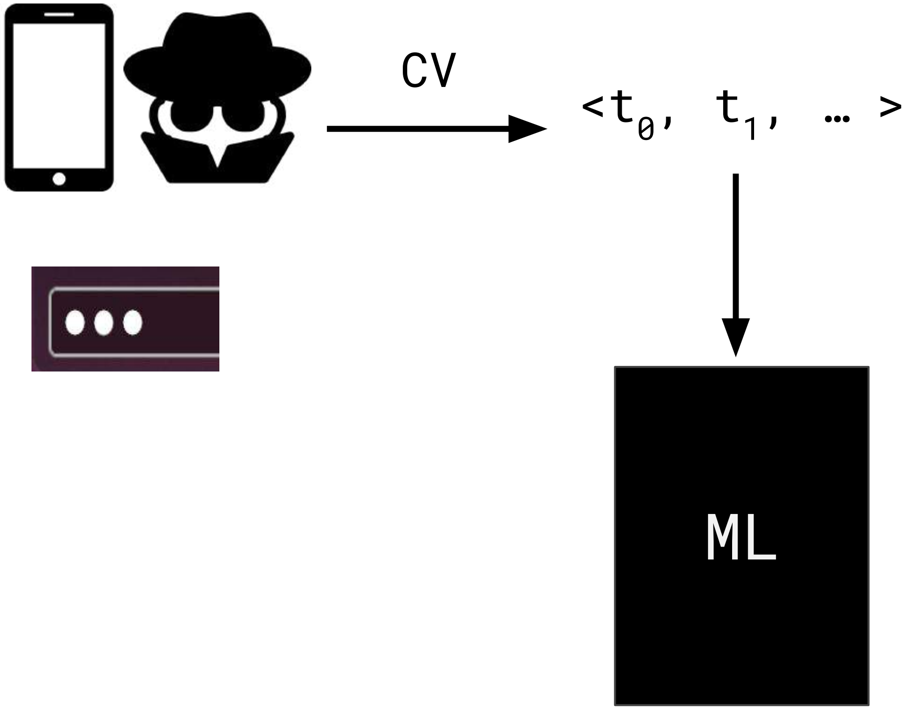
- Quantify information leakage of on-screen keystroke feedback

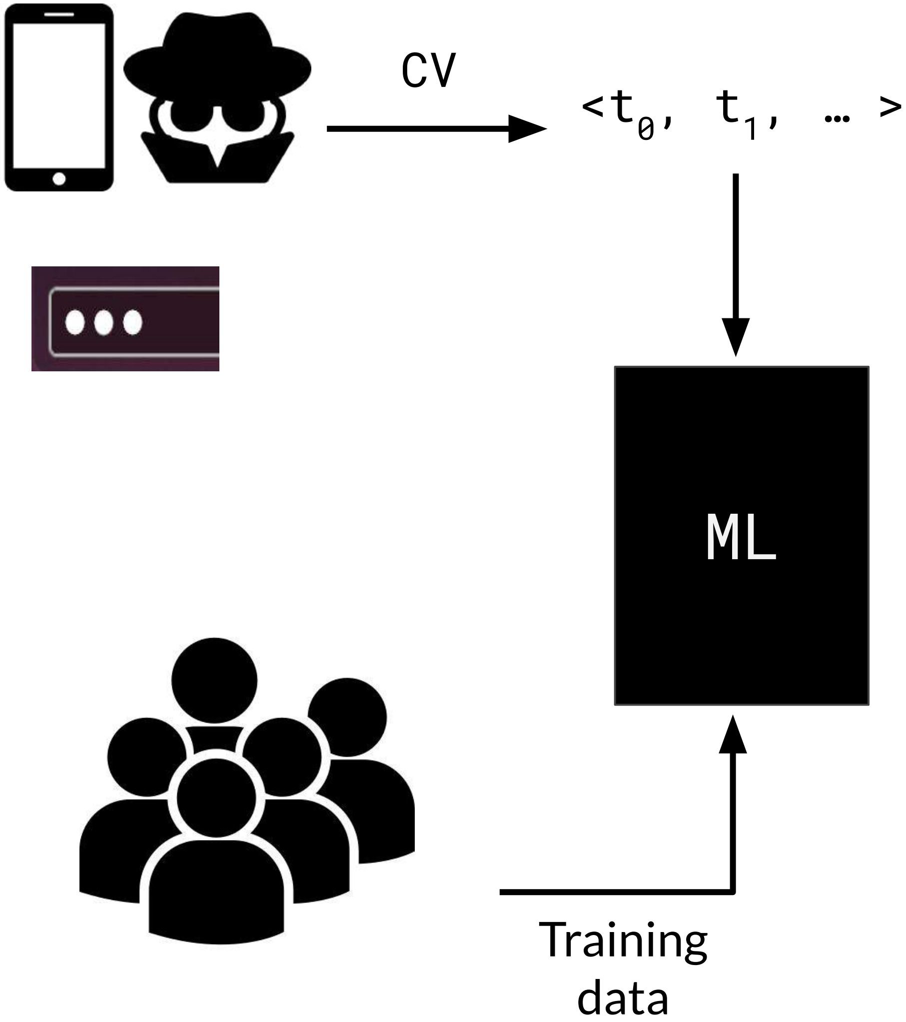
- Novel attack: *SILK-TV*
  - *Uses public datasets only from multiple sources ("population data")*
  - *Machine Learning to guess typed text (passwords and PINs)*

SILK-TV

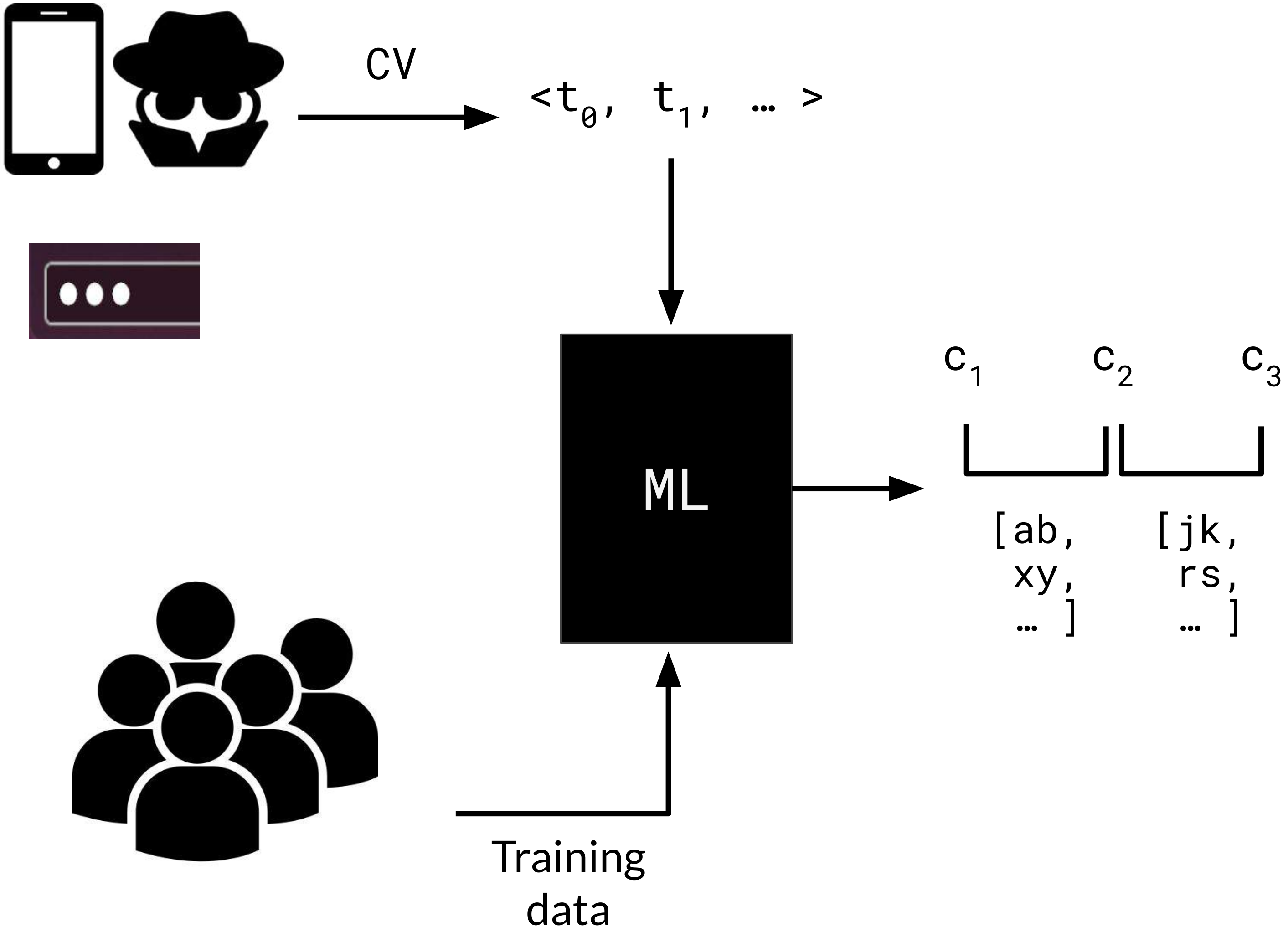


SILK-TV

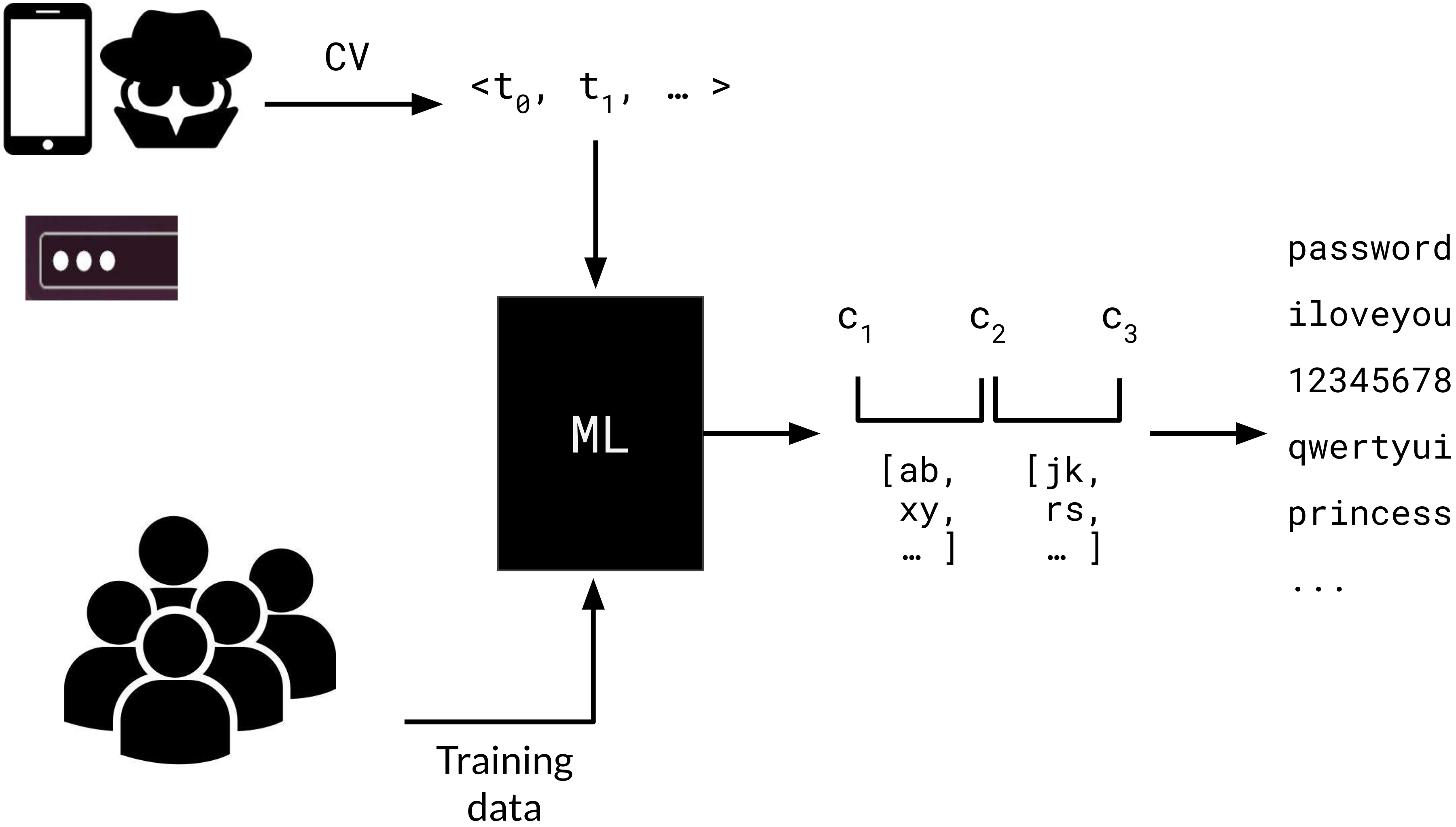



CV
<t0, t1, … >
ML
Training
data

SILK-TV



SILK-TV


CV
<t0, t1, … >
ML
c1 c2 c3
[ab,
xy,
…]
[jk,
rs,
…]
password
iloveyou
12345678
qwertyui
princess
...
Training
data

Data Collection - Passwords

- Data from **projector** and **laptop screen** @ 60Hz
- Recorded with a smartphone
- 62 users - 3 times each pwd - ***touch* typing** on keyboard
- Randomly selected 4 passwords from `rockyou`[1]
  - *123brian, jillie02, lamondre, william1*

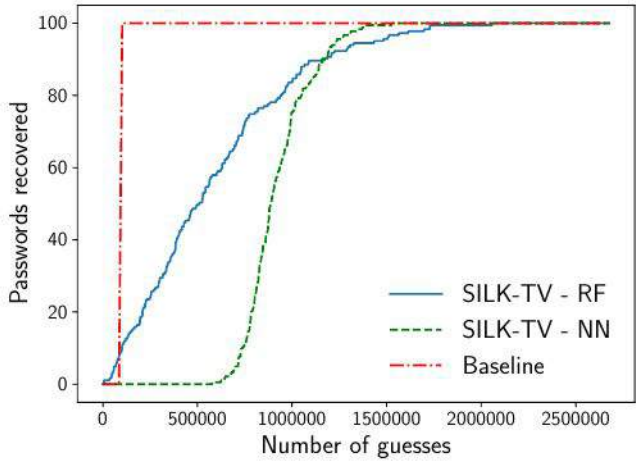
1 - http://downloads.skullsecurity.org/passwords/rockyou.txt.bz2

Evaluation - Passwords

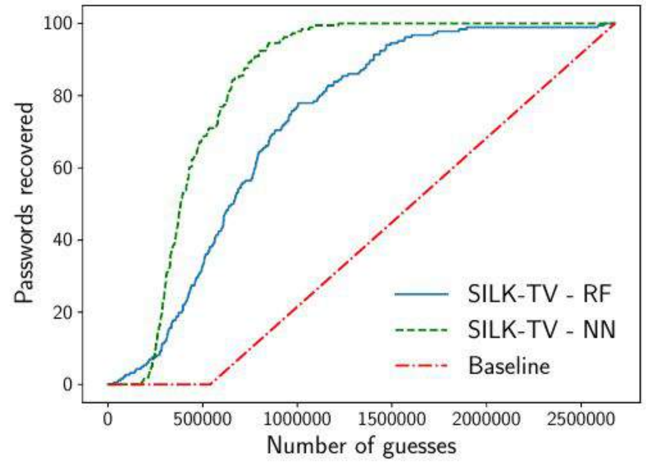
- Baseline: password list sorted by frequency
  - *"Best" strategy for a zero-information attacker*
  - `123brian` - *93,874th*
  - `jillie02` - *1,753,571st*
  - `lamondre` - *397,213rd*
  - `william1` - *187th* ← *very frequent password*

- Evaluation scenarios
  - *"Single shot"*
  - *"Multiple recordings" (e.g., professor at lectures)*

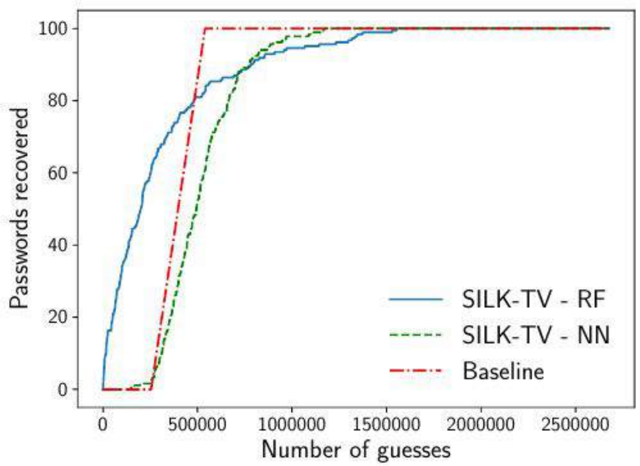
Password - "Single Shot" results



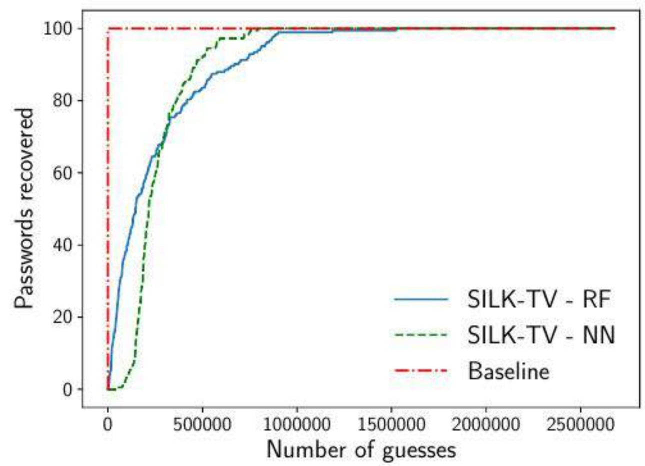
(a) `123brian` (183 auth. attempts).



(b) `jillie02` (186 auth. attempts).

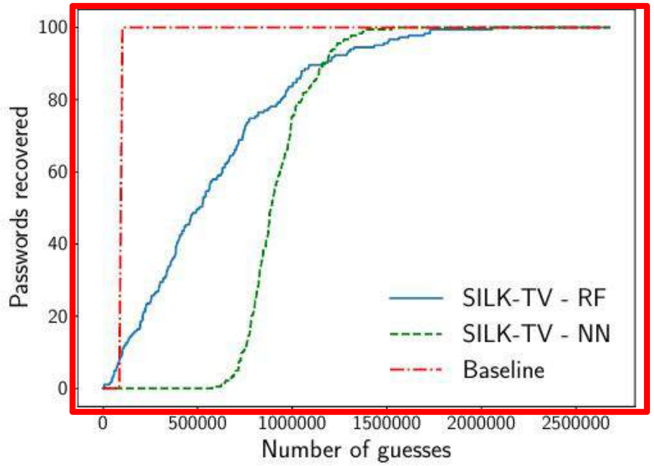


(c) `lamondre` (184 auth. attempts).

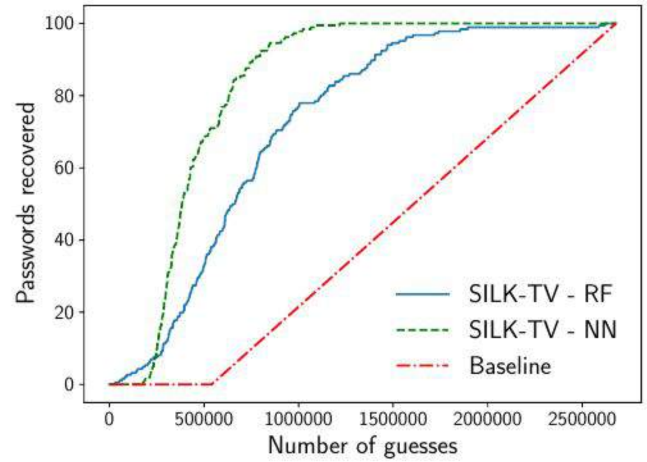


(d) `william1` (183 auth. attempts).

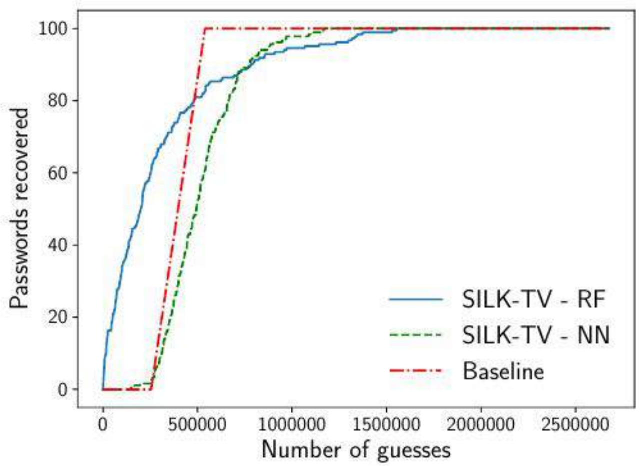
Password - “Single Shot” results



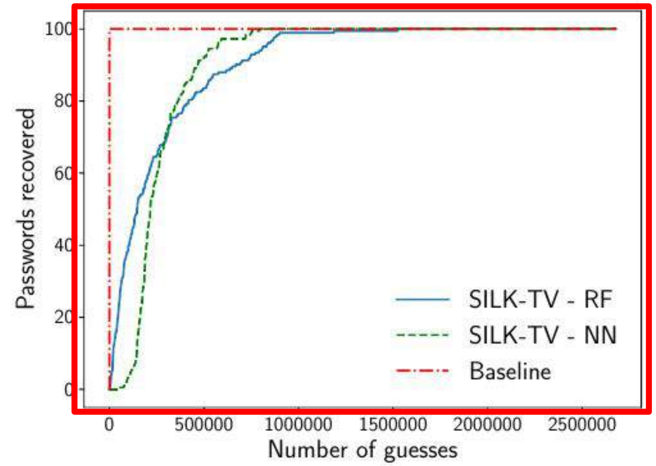
(a) `123brian` (183 auth. attempts).



(b) `jillie02` (186 auth. attempts).

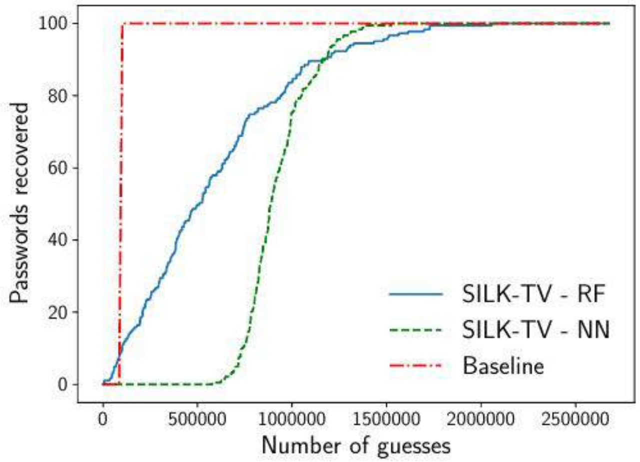


(c) `lamondre` (184 auth. attempts).

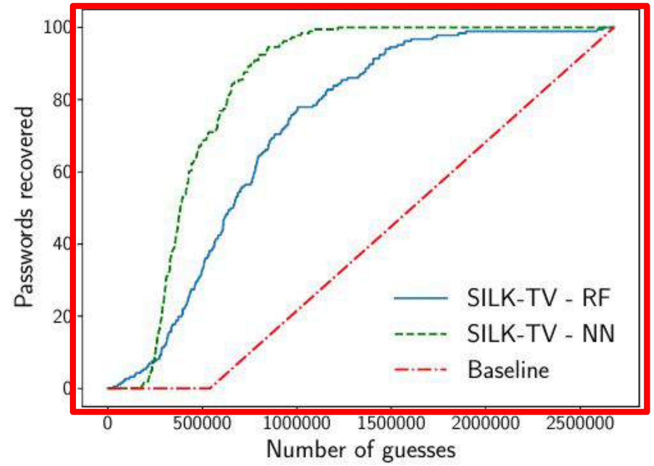


(d) `william1` (183 auth. attempts).

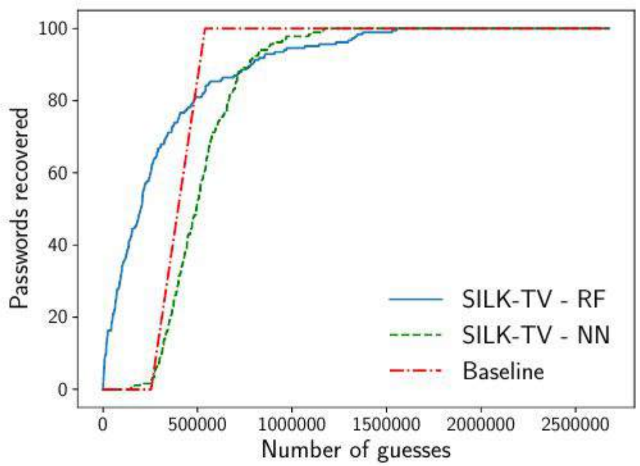
Password - “Single Shot” results



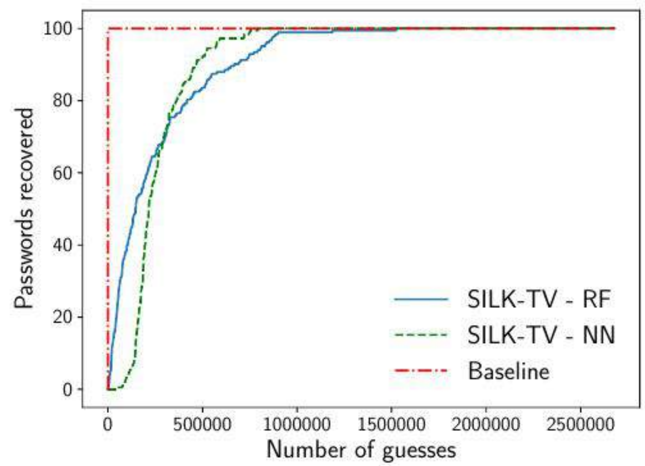
(a) **123brian** (183 auth. attempts).



(b) **jillie02** (186 auth. attempts).



(c) **lamondre** (184 auth. attempts).



(d) **william1** (183 auth. attempts).

Password - “Single Shot” results

	Avg	Stdev	Med	Rnd	<Rnd	Best	<20k	<100k
Random Forest								
123brian	581,743	414,761	508,332	93,874	8.7%	5,535	1.1%	9.3%
jillie02	749,718	448,319	656,754	1,753,571	97.8%	28,962	0.0%	2.7%
lamondre	301,906	334,681	199,344	397,213	75.0%	145	13.0%	33.7%
william1	246,437	264,090	145,966	187	0.5%	68	10.9%	39.9%
Neural Network								
123brian	923,534	165,454	886,802	93,874	0.0%	577,739	0.0%	0.0%
jillie02	456,811	210,512	383,230	1,753,571	100.0%	164,754	0.0%	0.0%
lamondre	517,472	189,355	493,713	397,213	28.8%	148,403	0.0%	0.0%
william1	265,813	140,753	215,840	187	0.0%	45,176	0.0%	3.8%

`Avg, Stdev, Median` *of SILK-TV cracking attempts*

`Rnd` *average baseline cracking attempts*

`<Rnd, Best, <20k, <100k` *highlights of SILK-TV performance*

Password - “Single Shot” results

	Avg	Stdev	Med	Rnd	<Rnd	Best	<20k	<100k
Random Forest								
123brian	581,743	414,761	508,332	93,874	8.7%	5,535	1.1%	9.3%
jillie02	749,718	448,319	656,754	1,753,571	97.8%	28,962	0.0%	2.7%
lamondre	301,906	334,681	199,344	397,213	75.0%	145	13.0%	33.7%
william1	246,437	264,090	145,966	187	0.5%	68	10.9%	39.9%
Neural Network								
123brian	923,534	165,454	886,802	93,874	0.0%	577,739	0.0%	0.0%
jillie02	456,811	210,512	383,230	1,753,571	100.0%	164,754	0.0%	0.0%
lamondre	517,472	189,355	493,713	397,213	28.8%	148,403	0.0%	0.0%
william1	265,813	140,753	215,840	187	0.0%	45,176	0.0%	3.8%

`Avg, Stdev, Median` *of SILK-TV cracking attempts*

`Rnd` *average baseline cracking attempts*

`<Rnd, Best, <20k, <100k` *highlights of SILK-TV performance*

Password - "Single Shot" results

	Avg	Stdev	Med	Rnd	<Rnd	Best	<20k	<100k
Random Forest								
123brian	581,743	414,761	508,332	93,874	8.7%	5,535	1.1%	9.3%
jillie02	749,718	448,319	656,754	1,753,571	97.8%	28,962	0.0%	2.7%
lamondre	301,906	334,681	199,344	397,213	75.0%	145	13.0%	33.7%
william1	246,437	264,090	145,966	187	0.5%	68	10.9%	39.9%
Neural Network								
123brian	923,534	165,454	886,802	93,874	0.0%	577,739	0.0%	0.0%
jillie02	456,811	210,512	383,230	1,753,571	100.0%	164,754	0.0%	0.0%
lamondre	517,472	189,355	493,713	397,213	28.8%	148,403	0.0%	0.0%
william1	265,813	140,753	215,840	187	0.0%	45,176	0.0%	3.8%

`Avg, Stdev, Median` *of SILK-TV cracking attempts*

`Rnd` *average baseline cracking attempts*

`<Rnd, Best, <20k, <100k` *highlights of SILK-TV performance*

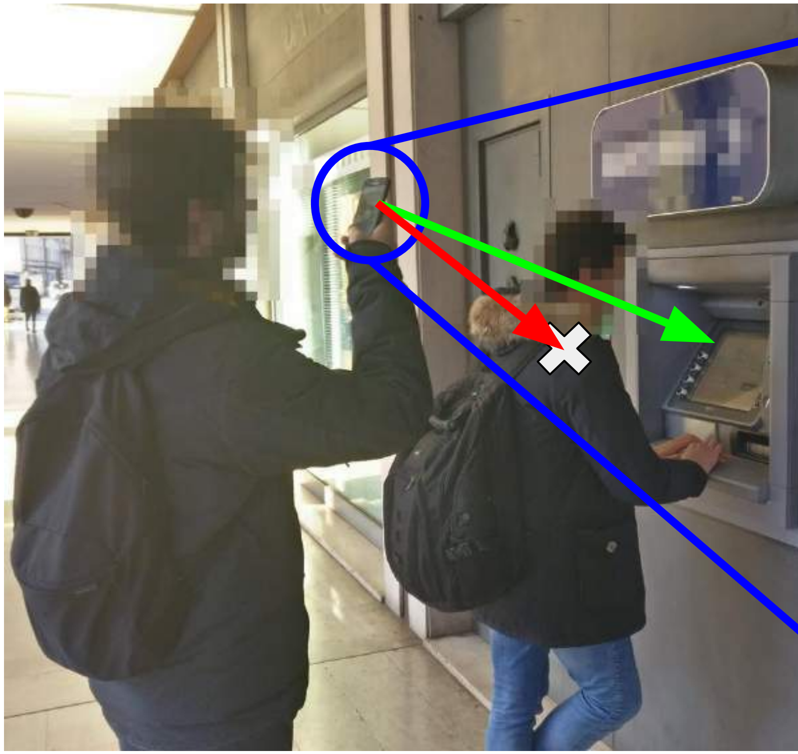
Conclusions

- Timing information from videos is **accurate**
- Password masking leak timing → useful information
  - *Reduces number of attempts*
  - *More useful on **uncommon** passwords!*

PIN Salabim

PIN Salabim

PIN Salabim



Keypad not visible - but the screen is!

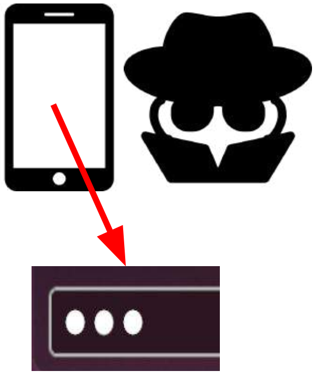
PILOT
Password and PIN Information Leakage from Obfuscated Typing Videos

Kiran Balagani, Matteo Cardaioli, Mauro Conti, Paolo Gasti, Martin Georgiev, Tristan Gurtler, Daniele Lain, Charissa Miller, Kendall Molas, Nikita Samarin, Eugen Saraci, Gene Tsudik, and Lynn Wu

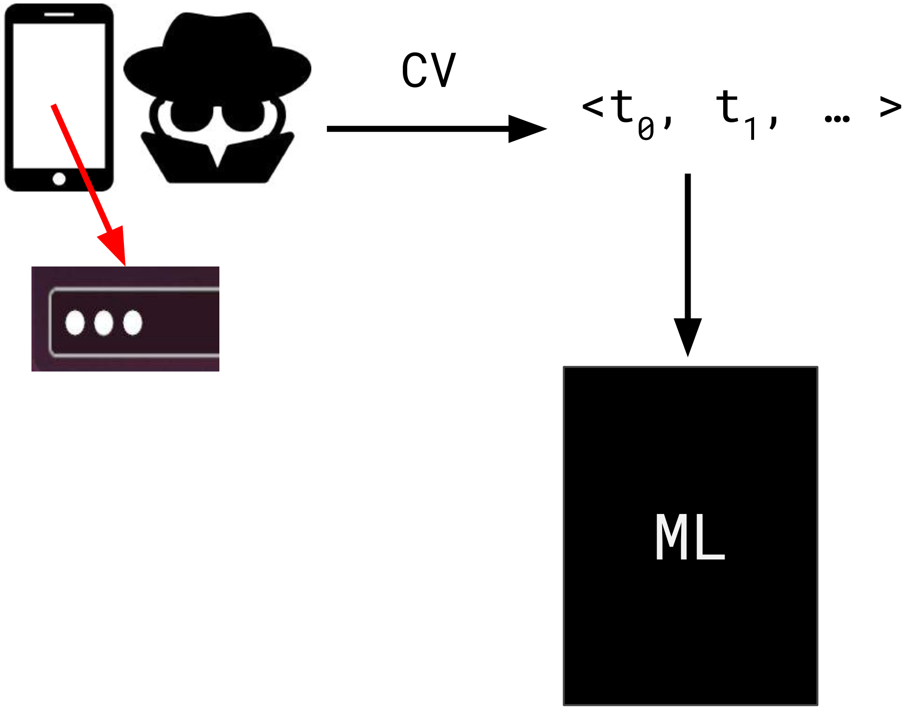
In Journal of Computer Security 2019

Università degli Studi di Padova
New York Institute of Technology
The University of California Irvine
ETH zürich

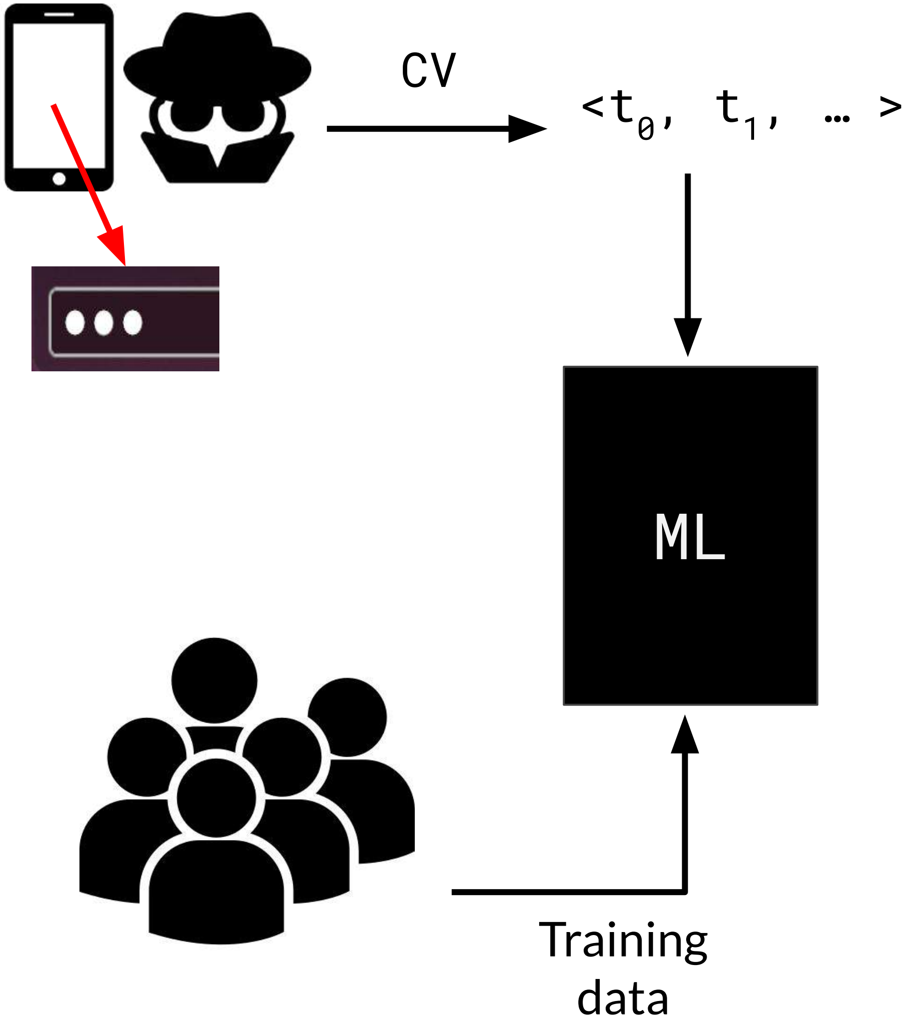
PILOT



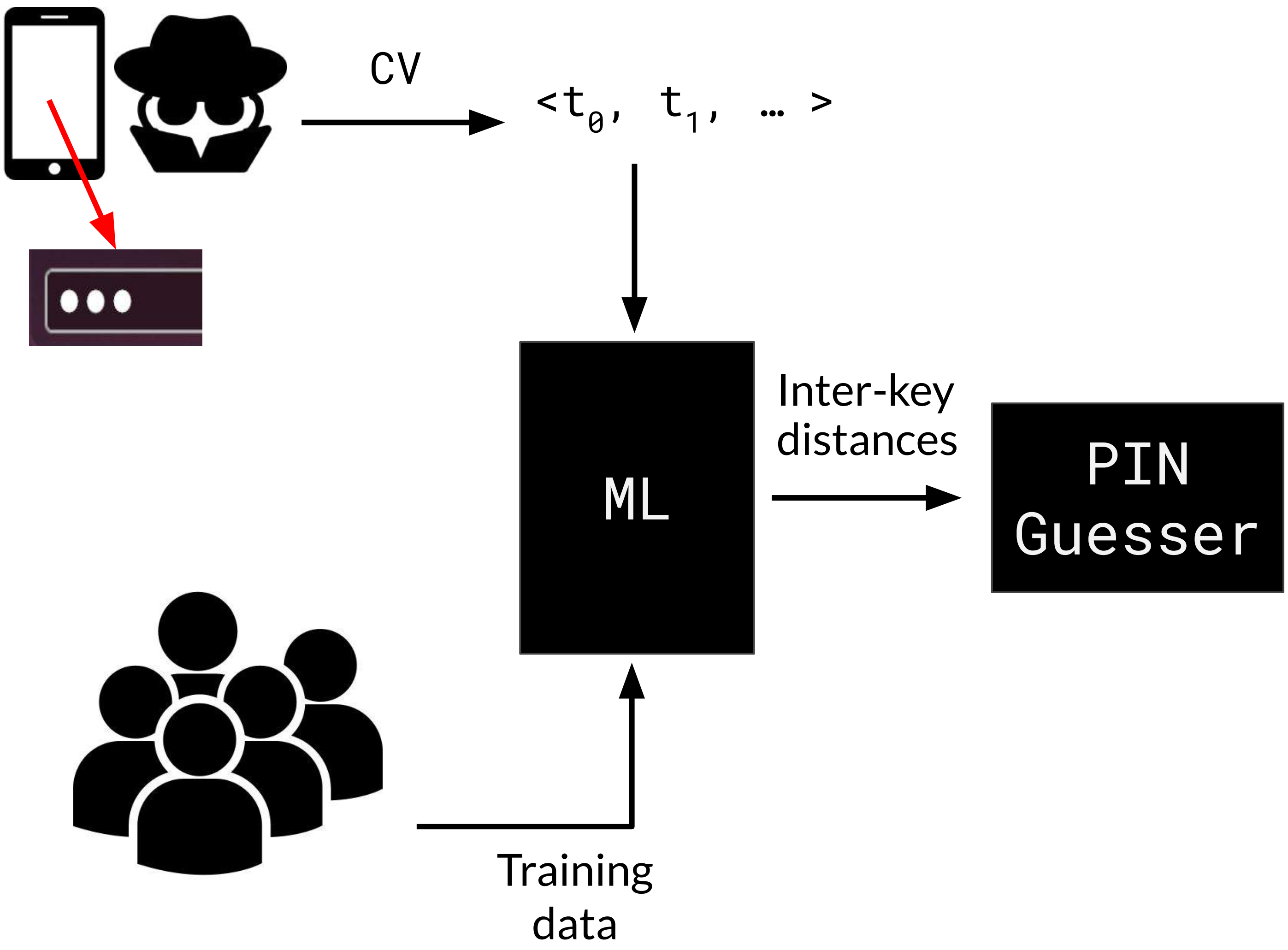
PILOT



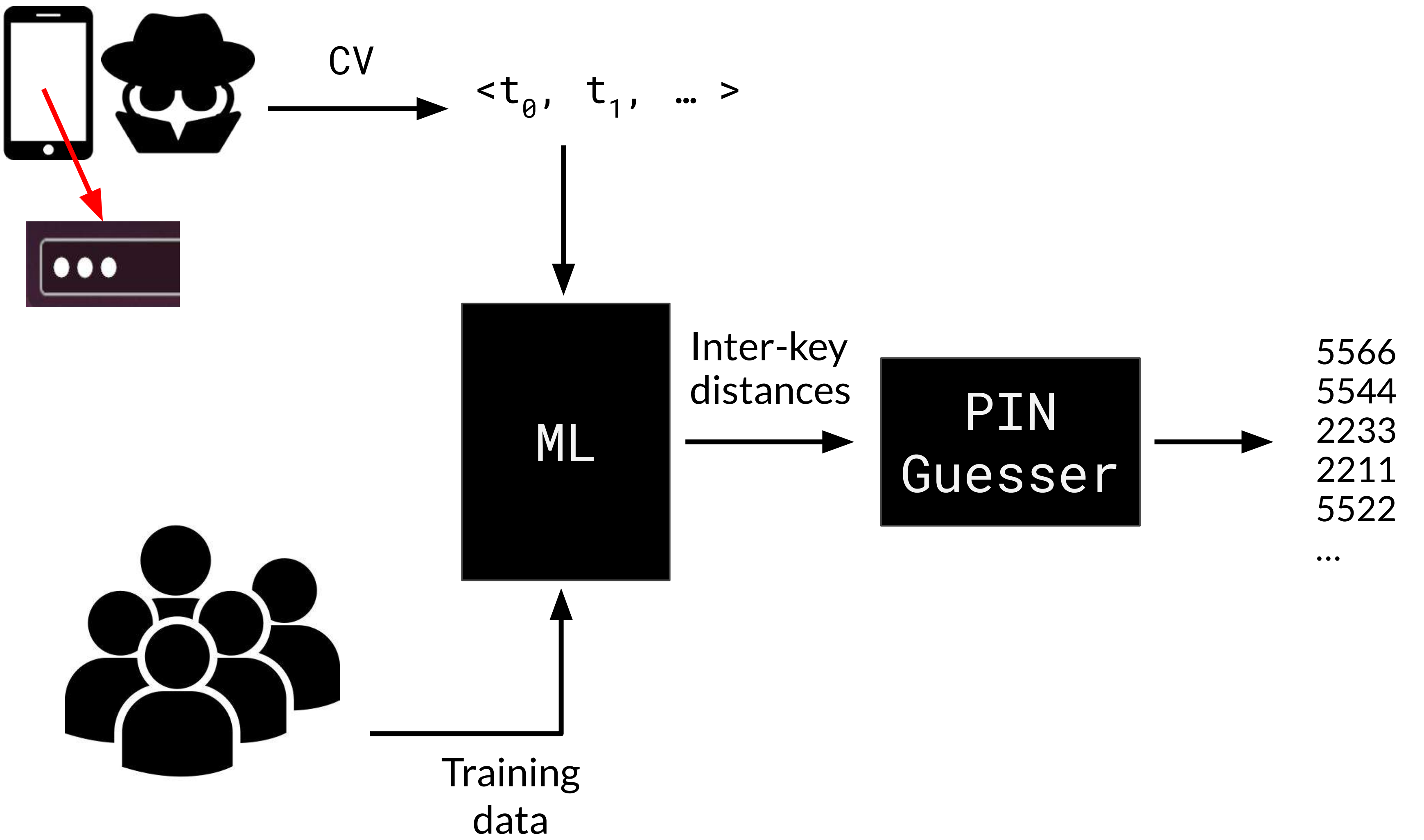
PILOT



PILOT



PILOT



CV
<t0, t1, … >
ML
Inter-key distances
PIN Guesser
5566
5544
2233
2211
5522
...
Training data

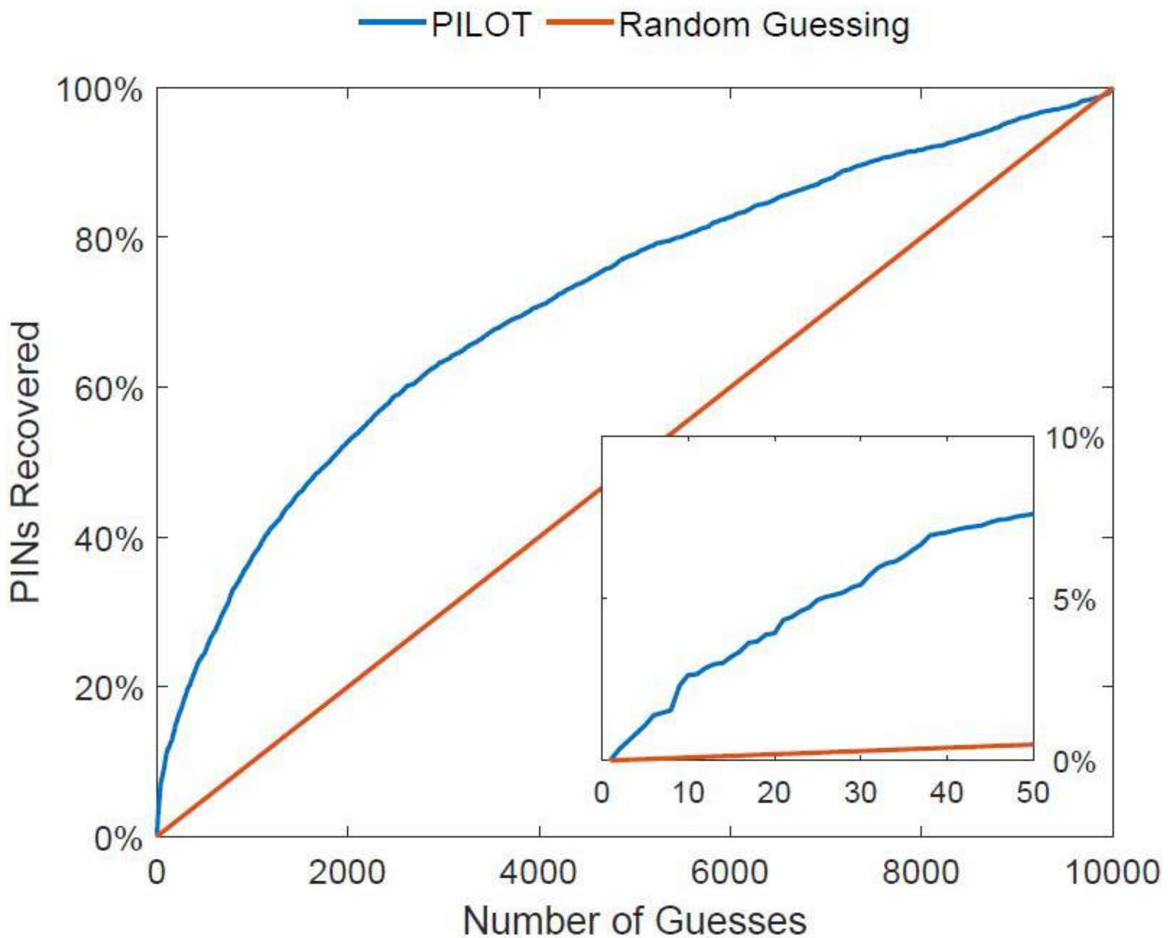
SPRITZ Security & Privacy Research Group

Università degli Studi di Padova

GFT

Percentage of PINs recovered with PILOT vs Random Guessing

- 4 digit PIN (USA ATM card)



Your PIN Sounds Good!
On The Feasibility of PIN Inference Through Audio Leakage

Matteo Cardaioli, Mauro Conti, Kiran Balagani, and Paolo Gasti

IEEE Transactions on Information Forensics and Security 2019 (Submitted)
https://arxiv.org/abs/1905.08742

Neither keypad nor screen are visible

SPRITZ Security & Privacy Research Group

Università degli Studi di Padova

GFT

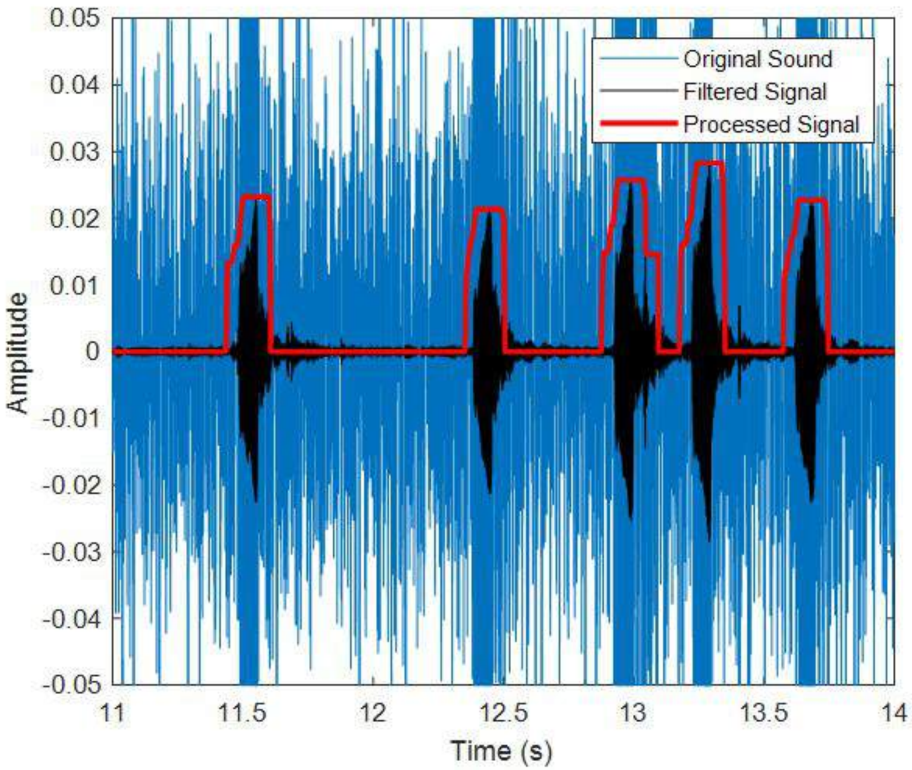
Inter-keystroke timing identification through sound analysis

- **Signal filtering**

  To extract feedback sound characteristic frequency

- **Signal processing**

  To remove residual noise and to identify time distance between peaks



Adversarial **additional knowledge** about the user or the PIN

- Knowledge of **typing behavior**

  Hunt-and-peck *vs.* ***touch typing***

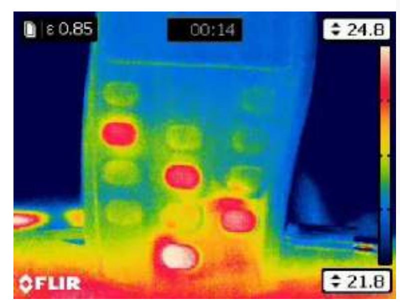
- Knowledge of **a digit**

  Adversary ***knows one digit*** *of the PIN*

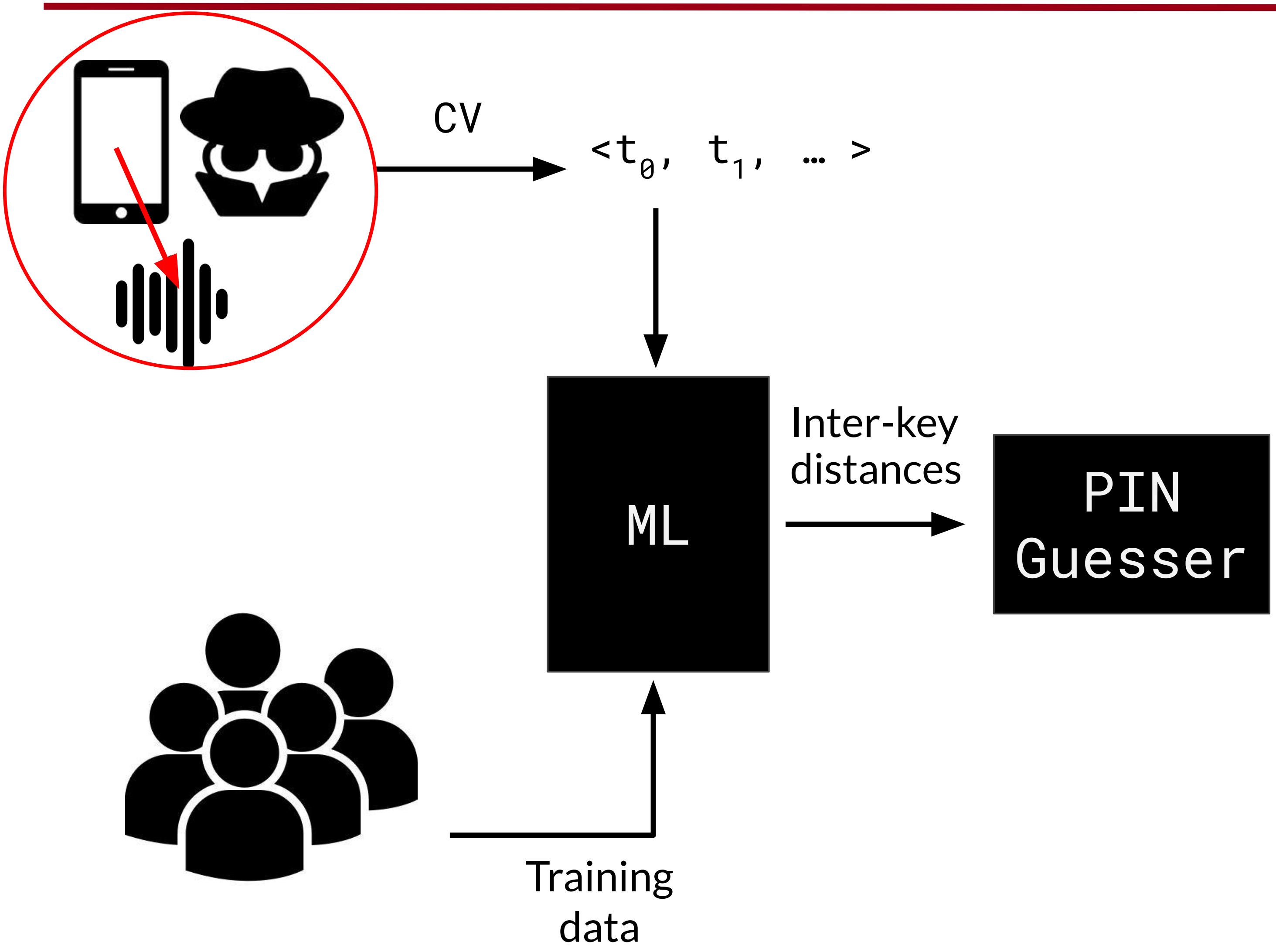
- **Heatmap**

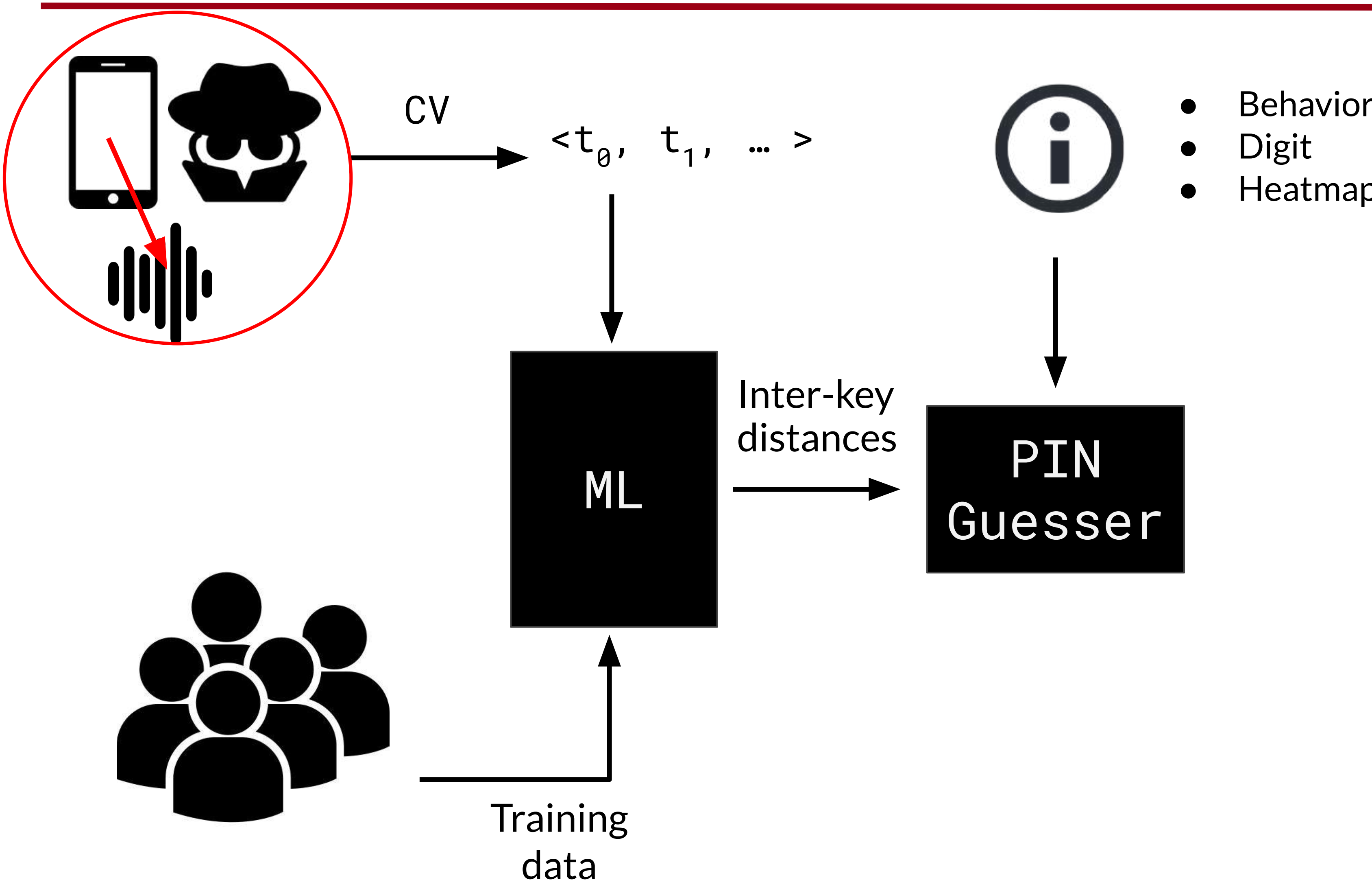
  Adversary performs a ***thermal attack***

  - *Better on plastic and rubber*
    Not so good on metal

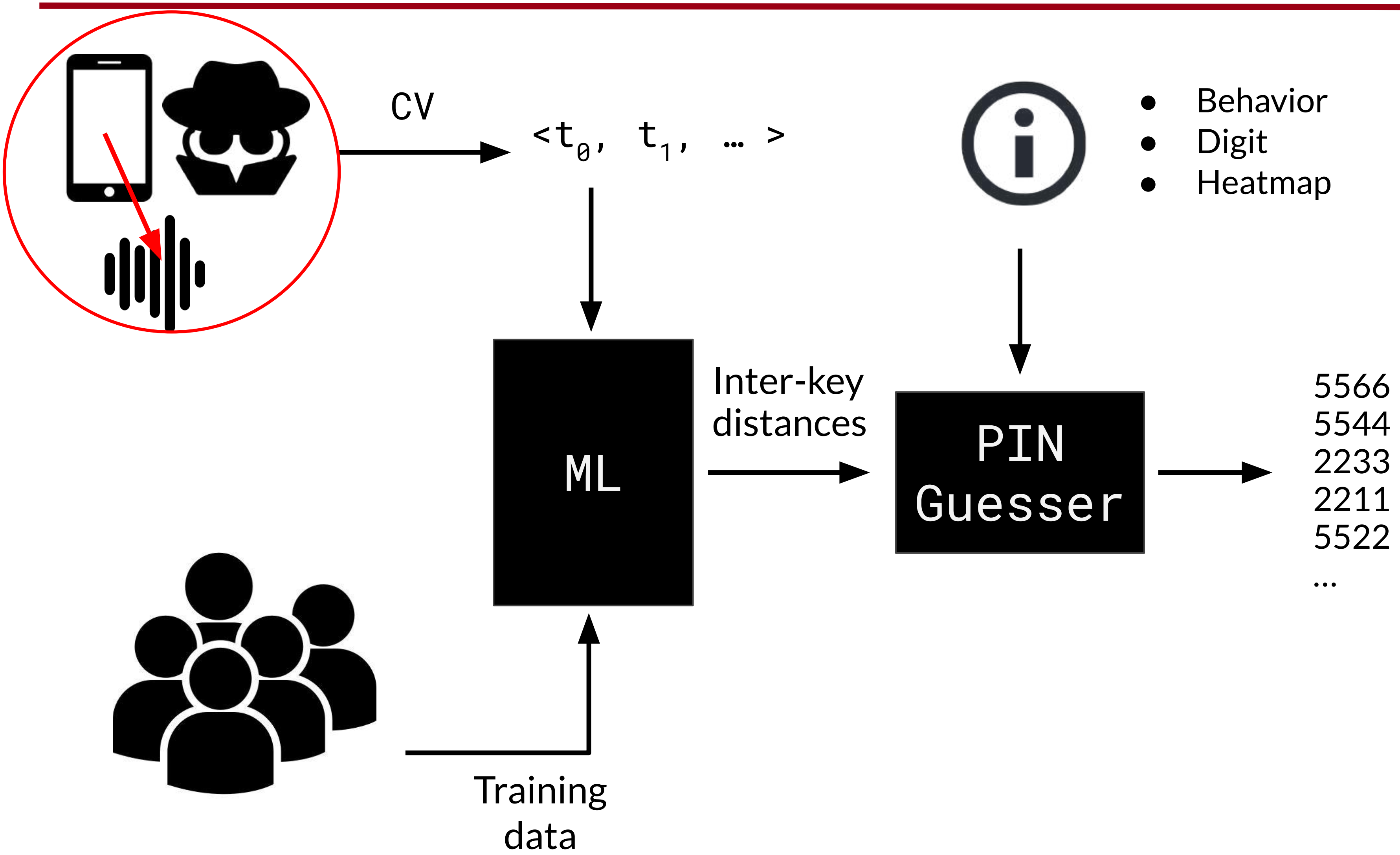


Your PIN Sounds Good!



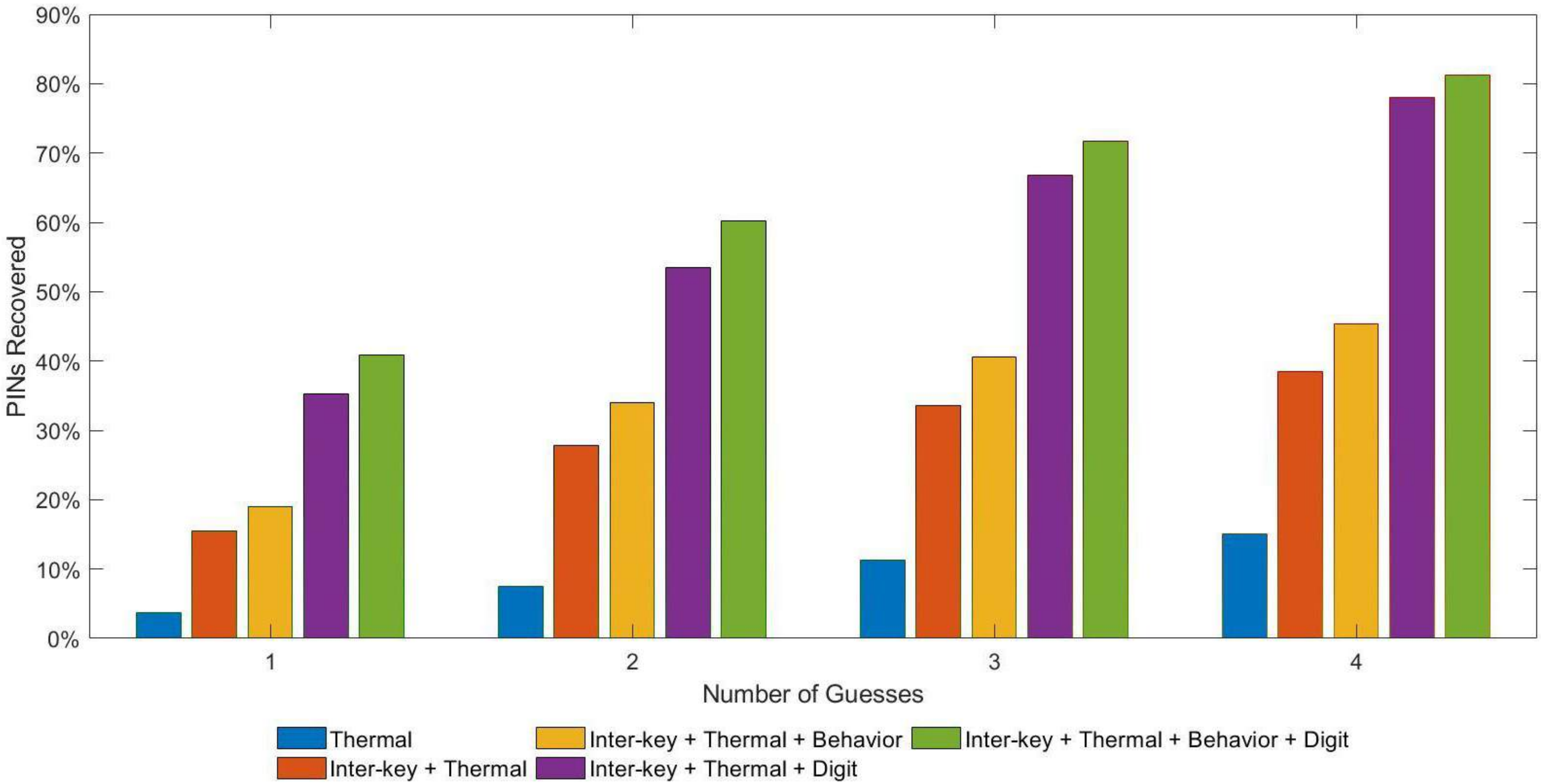

CV
<t0, t1, … >
Behavior
Digit
Heatmap
ML
Inter-key distances
PIN Guesser
Training data

Your PIN Sounds Good!

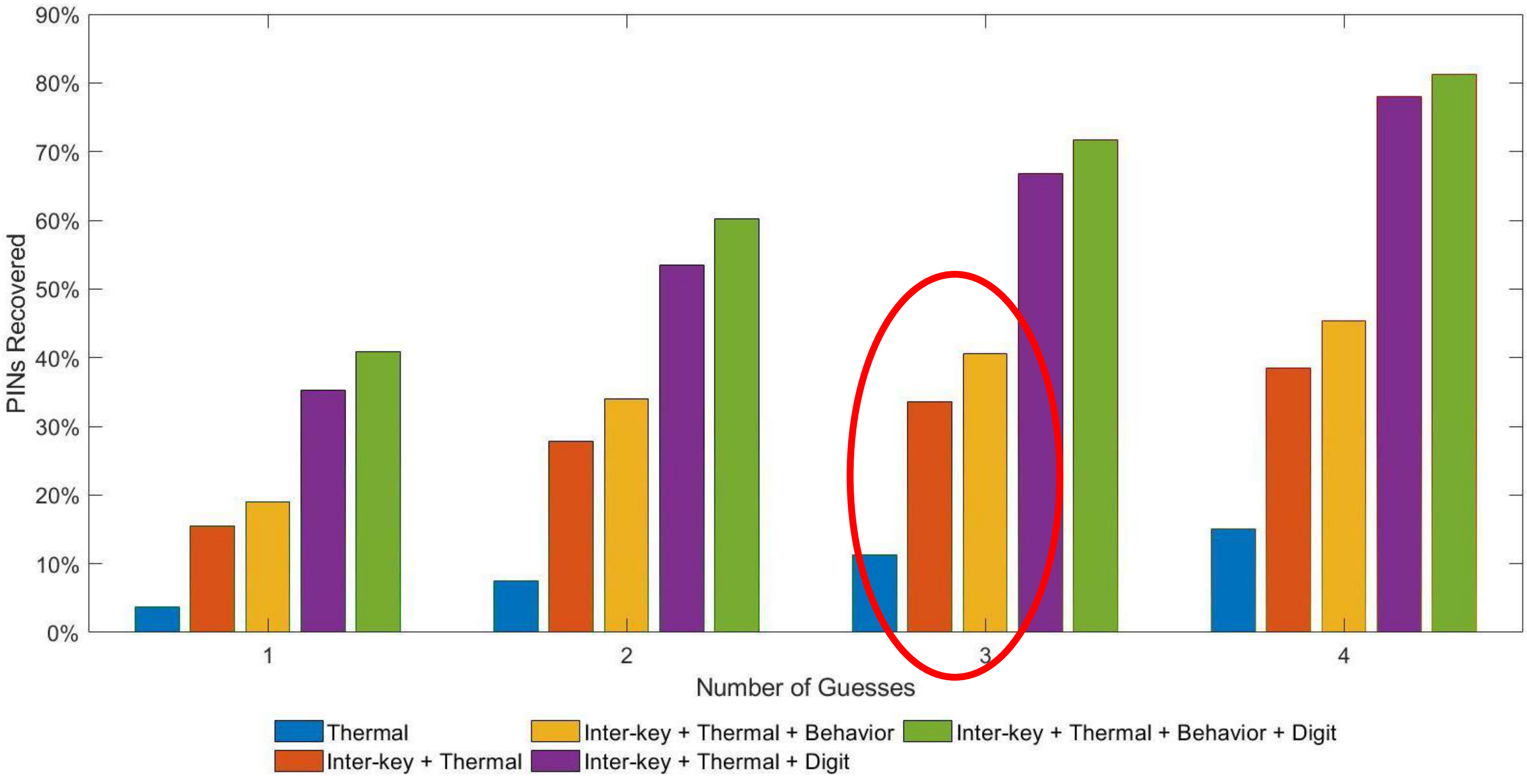


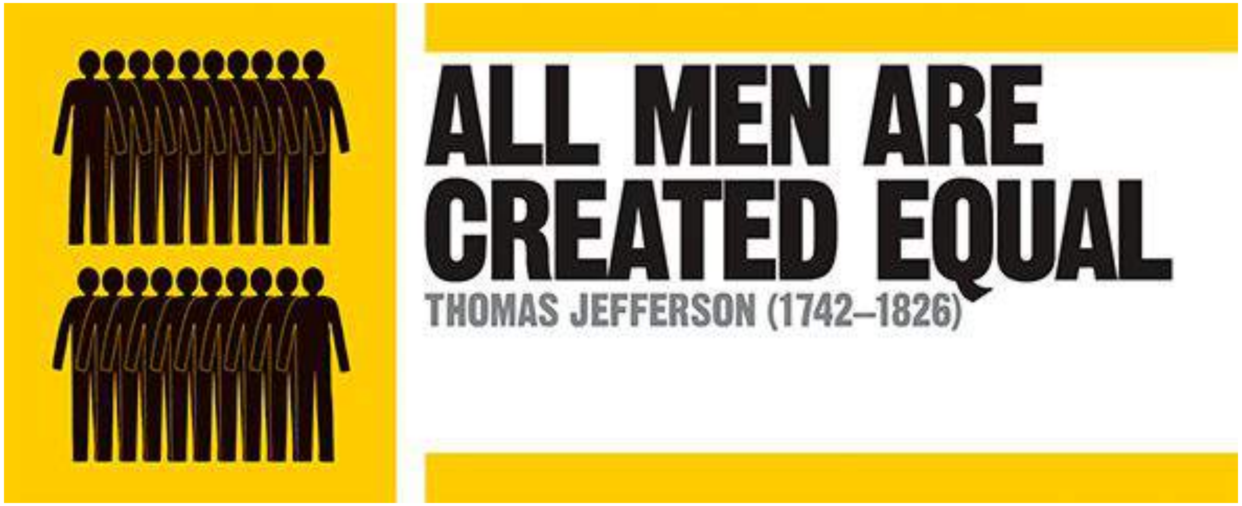
Your PIN Sounds Good!

% PINs recovered: inter-keystroke timing + other informations

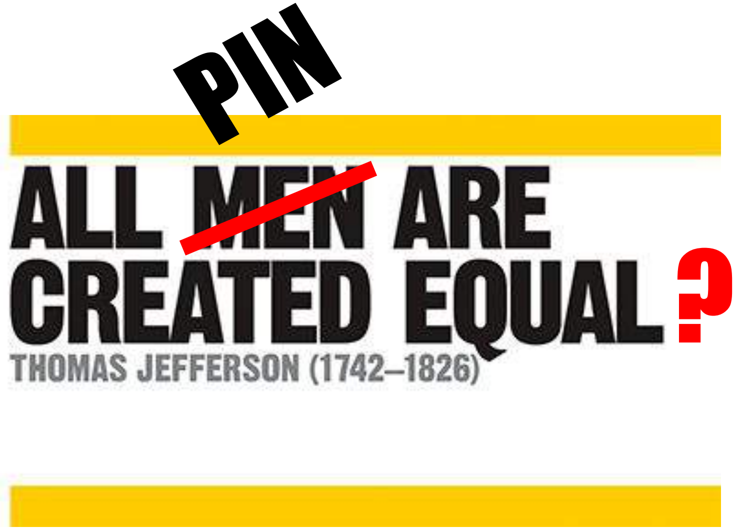


% PINs recovered: inter-keystroke timing + other informations

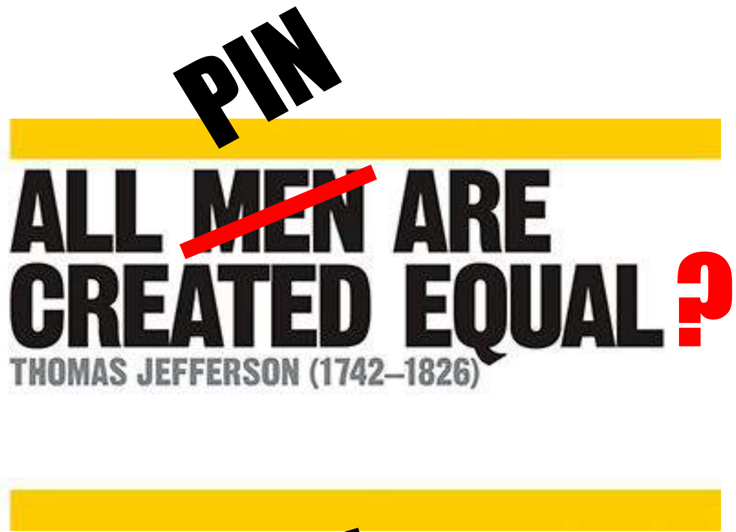



ALL MEN ARE CREATED EQUAL
THOMAS JEFFERSON (1742–1826)

Your PIN Sounds Good!

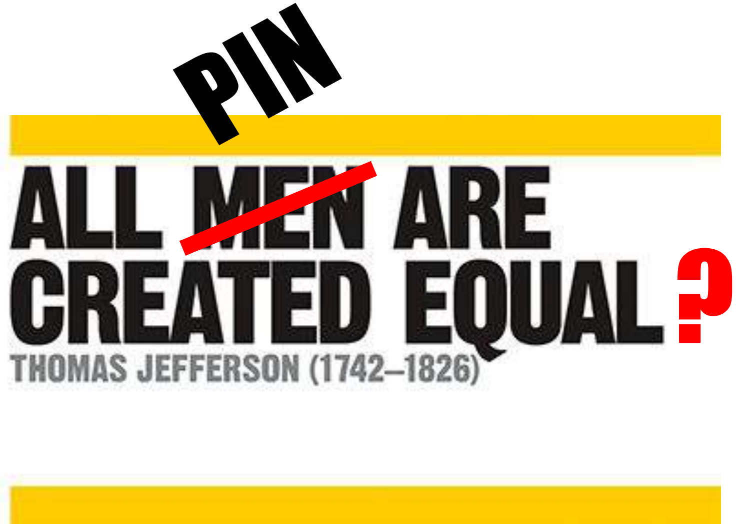


Your PIN Sounds Good!



User Chosen

Your PIN Sounds Good!



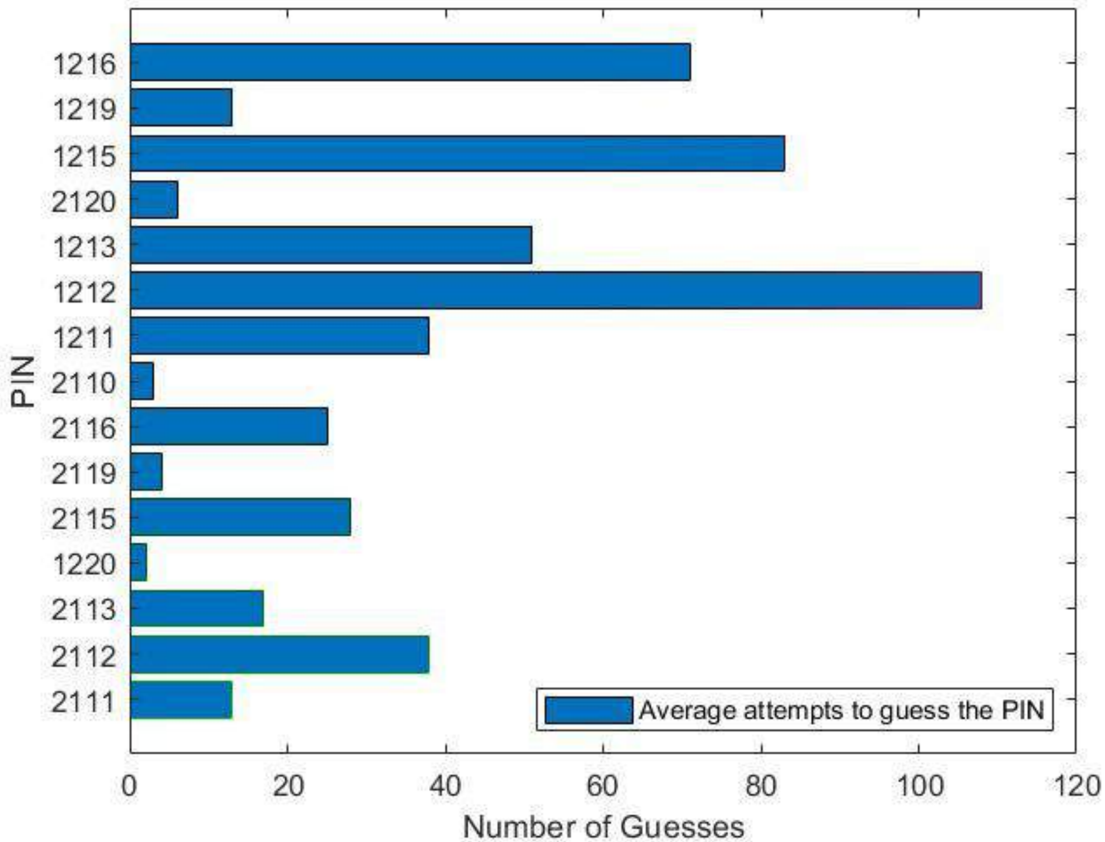
User Chosen

Random

Your PIN Sounds Good!

Not all PINs are born the same

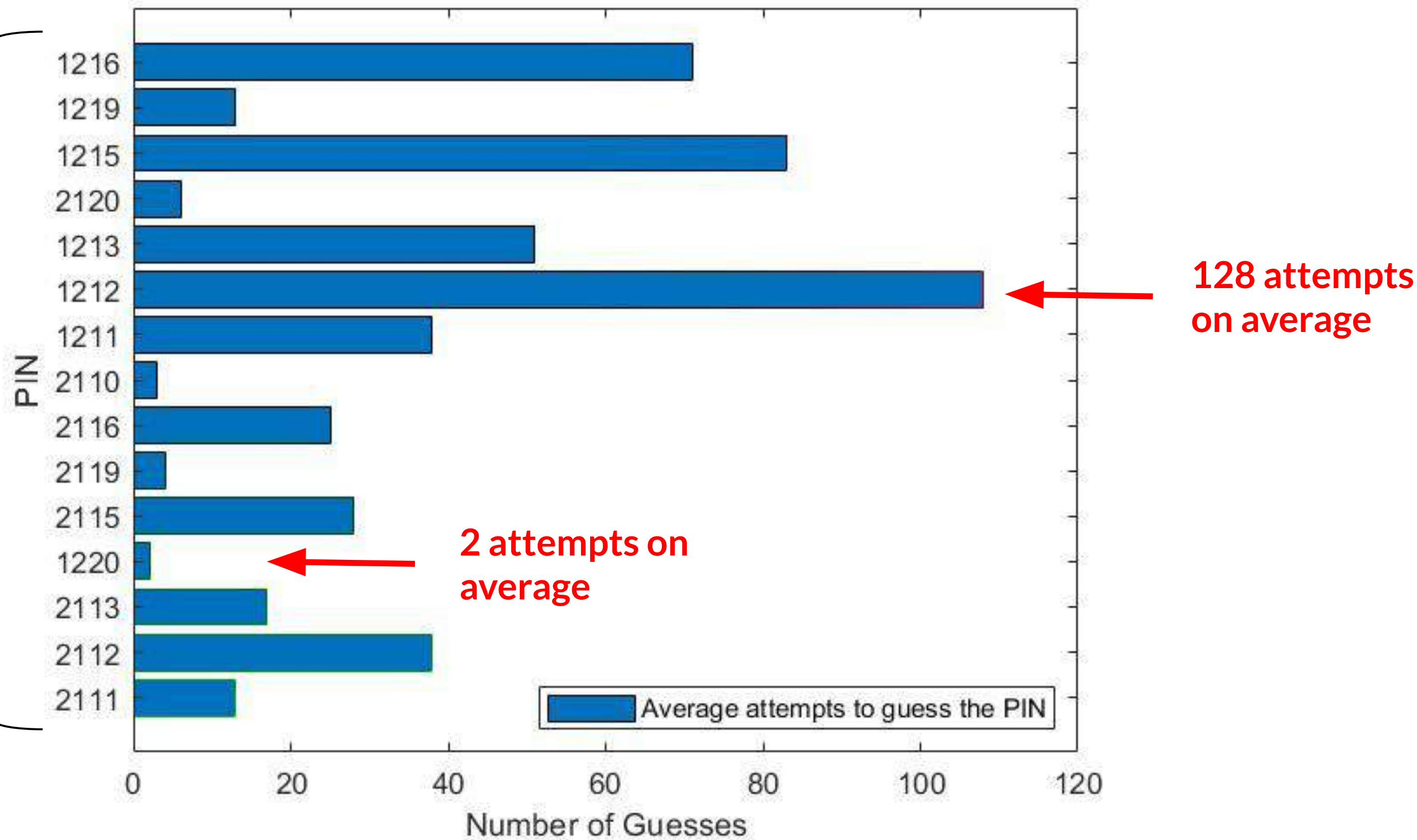
Knowing ***inter-key distance only***



Not all PINs are born the same

*Knowing **inter-key distance only*** → *PINs probability distribution is **no longer uniform***

Showing just a subset of PINs



DEMO time!

Outline

- Covert and Side Channels 101
- Network Traffic Analysis
  - *As a side channel: app and sensitive data inference*
- Energy Consumption
  - *As a side channel: user and app inference*
  - *As a covert channel: data exfiltration*
- Device Movement
  - *As a side channel: smartphone user authentication*
  - *Attacks against biometric authentication*
- Keystroke Timing
  - *As a side channel: text typed on keyboards*
- **Acoustic Emanations**
  - ***As a side channel: text typed on keyboards***

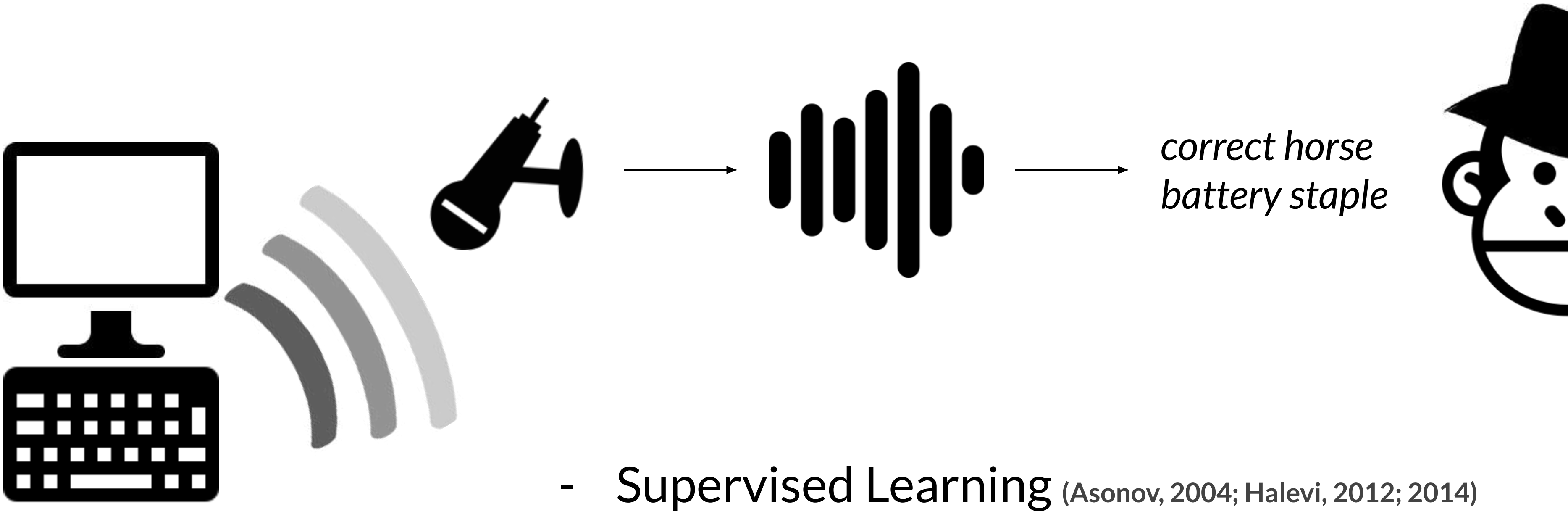
A. Compagno, M. Conti, D. Lain, G. Tsudik

Don't Skype & Type! Acoustic Eavesdropping in Voice-over-IP.

In ACM SIGSAC AsiaCCS 2017

Presented at Black Hat USA 2017

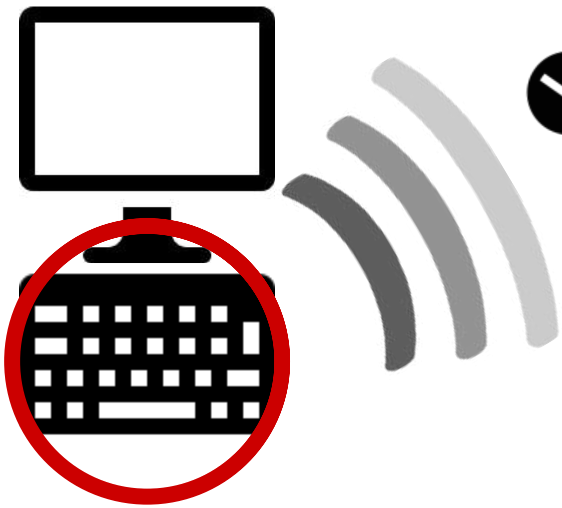
Keyboard Acoustic Eavesdropping

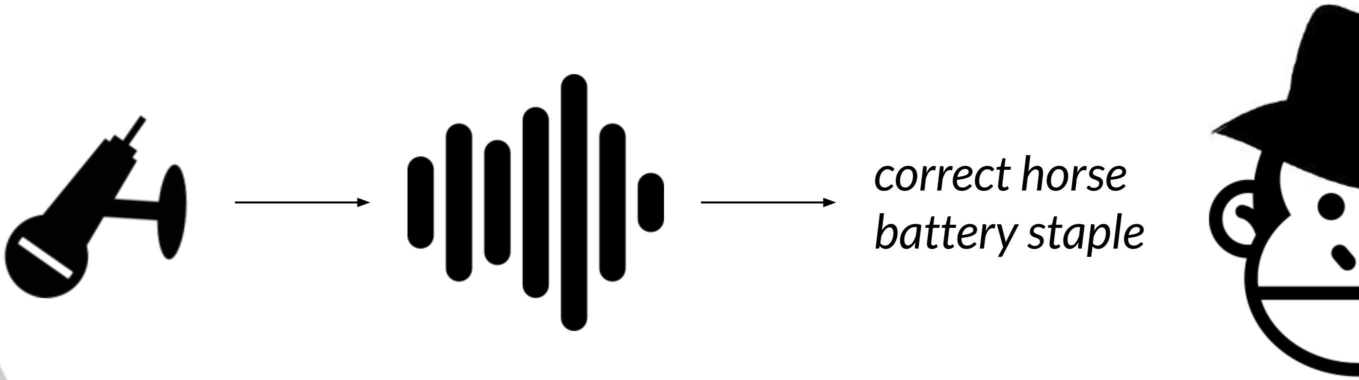


- **Supervised Learning** (Asonov, 2004; Halevi, 2012; 2014)
  Less input assumptions, more specific

- **Unsupervised Learning** (Berger, 2006; Zhuang, 2009)
  More input assumptions, more general

Keyboard Acoustic Eavesdropping





1 - How to get either:
- A lot of data
- Some labeled data

Keyboard Acoustic Eavesdropping

1 - How to get either:
- A lot of data
- Some labeled data

2 - How to place a compromised microphone close to my victim?

Motivation

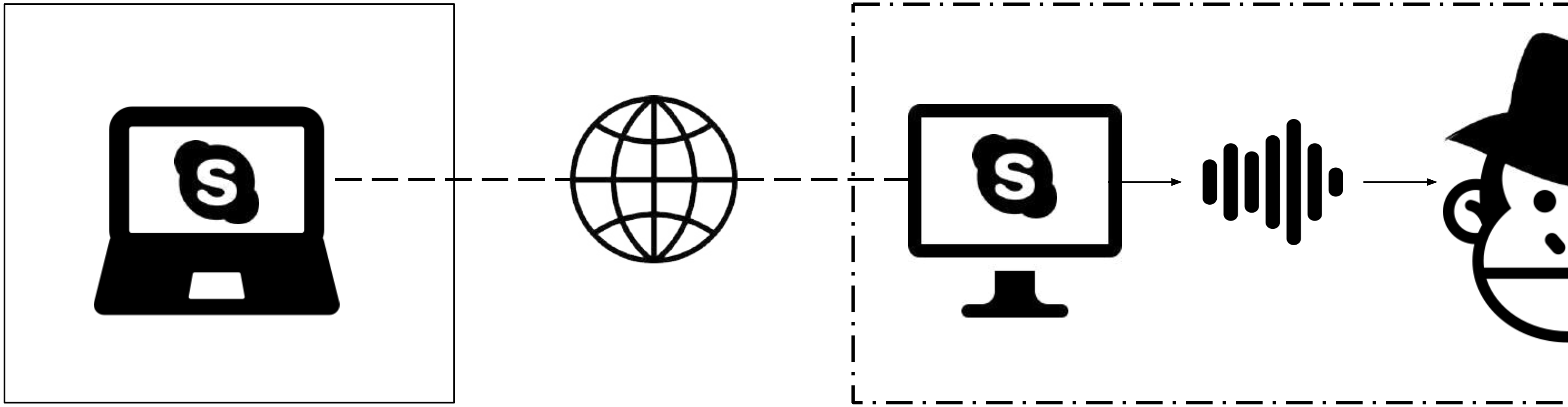
VoIP → one of the most used software: in academia, industry, at home

People type private stuff during Skype calls - it happens!

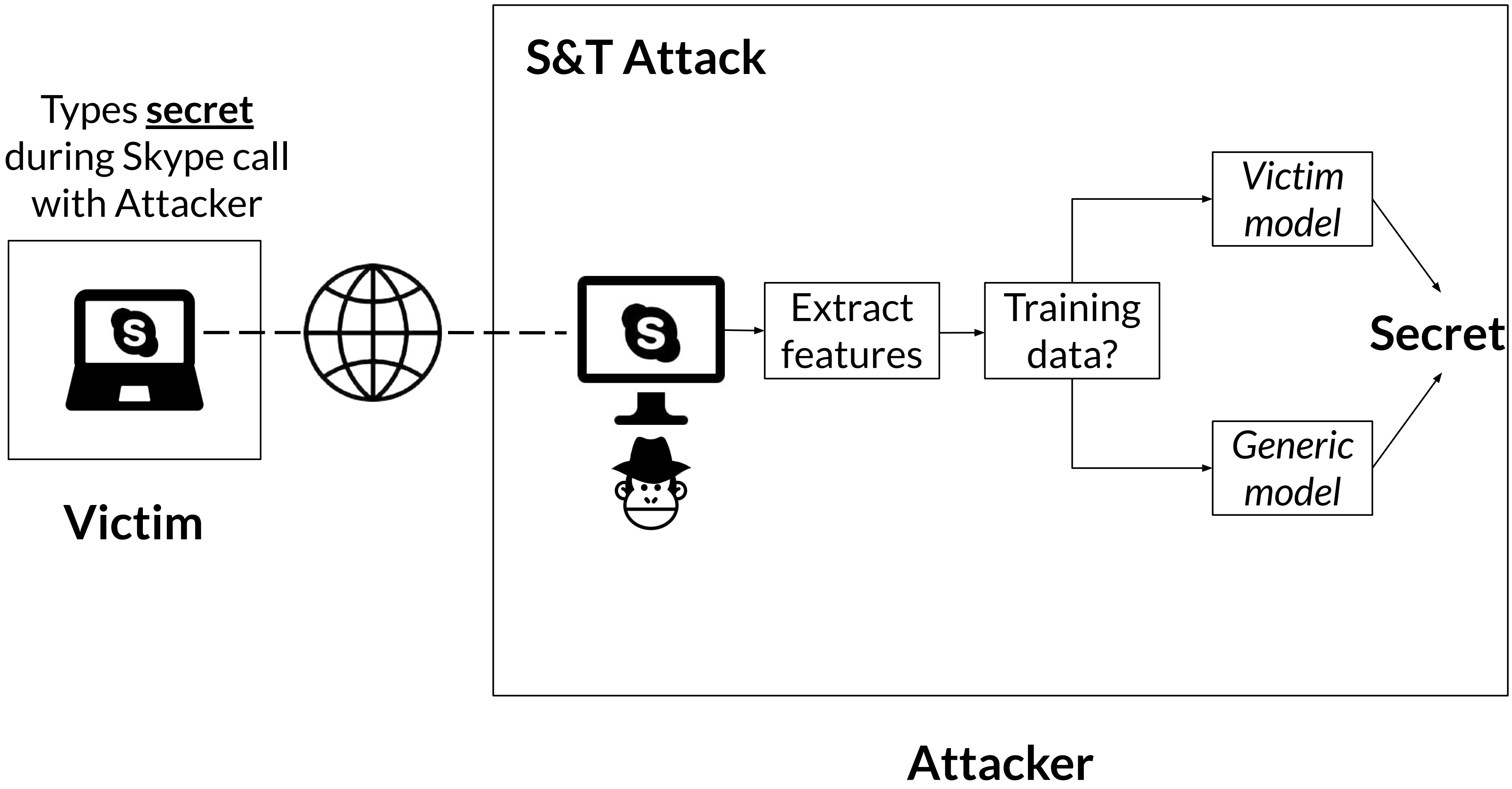
- *Login to websites*
- *Write a sensitive email*
- *Take notes*

We hear the keys' noise and use it to understand typed text

- *Victim is willingly giving us access to his microphone*



Skype&Type Attack



S&T - Tools

- Data windowing and segmentation

  To extract sound samples

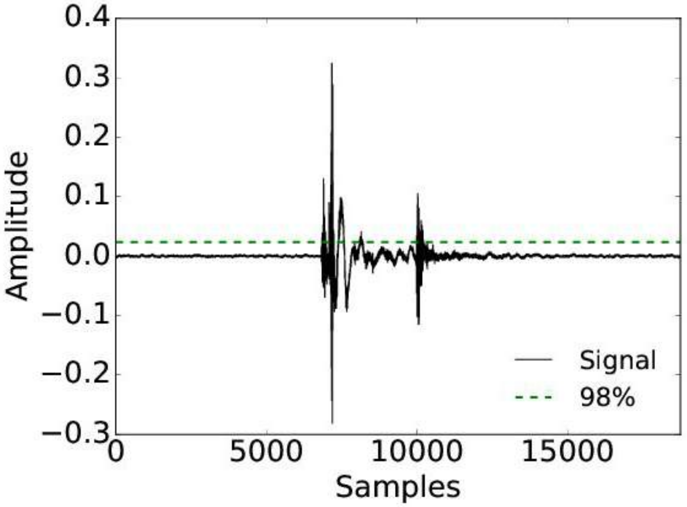
- Mel frequency cepstral coefficients

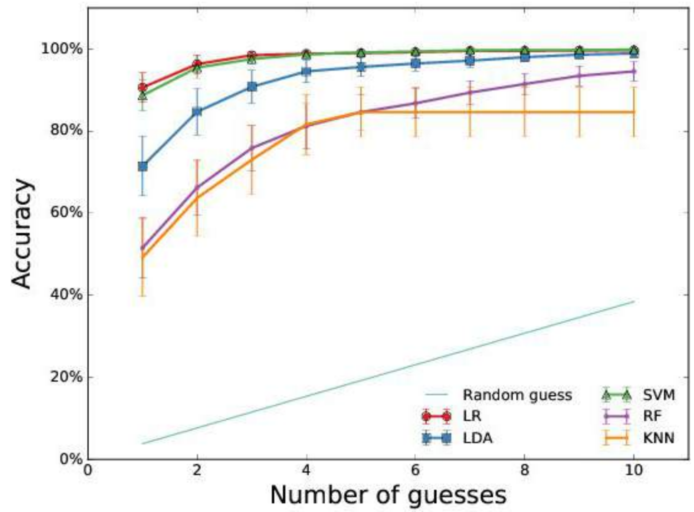
  Best performing and robust

- Supervised learning paradigm

  Target text can be possibly:
  - *Short (no clustering)*
  - *Random (no dictionary)*

- Logistic Regression classifier





Data Collection

- Try S&T in many scenarios
  - *With **5** different users over **Skype** (Google Hangouts also vulnerable)*
  - *Using **3** different common laptops: Macbook Pro, Lenovo, Toshiba*
  - *With **2** typing styles: single finger, and natural "touch" typing*

- Evaluate top-n accuracy of character recognition

  as a function of the number of guesses, focus on top-1 and top-5 accuracy

- Against a "dumb" random guess

  Might be a random password -- we can not use "smarter" approaches

Attack Scenarios

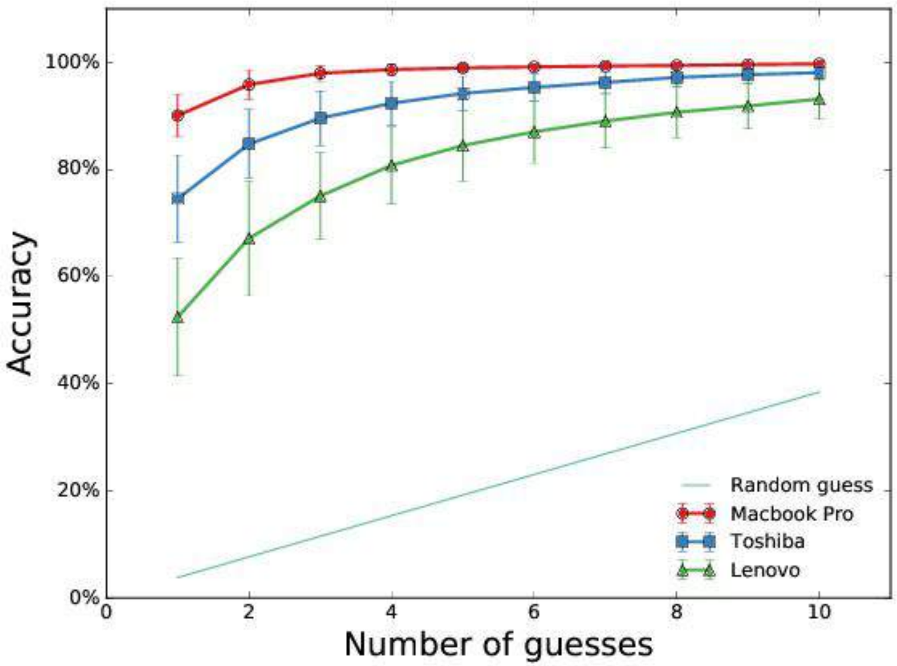
Evaluate the attack on two realistic scenarios

- Complete Profiling Scenario (Asonov, 2004; Halevi, 2012; 2014)
  - *Profiled the user on his laptop → specific training set*
  - *Ground truth disclosure, e.g., a short chat message*

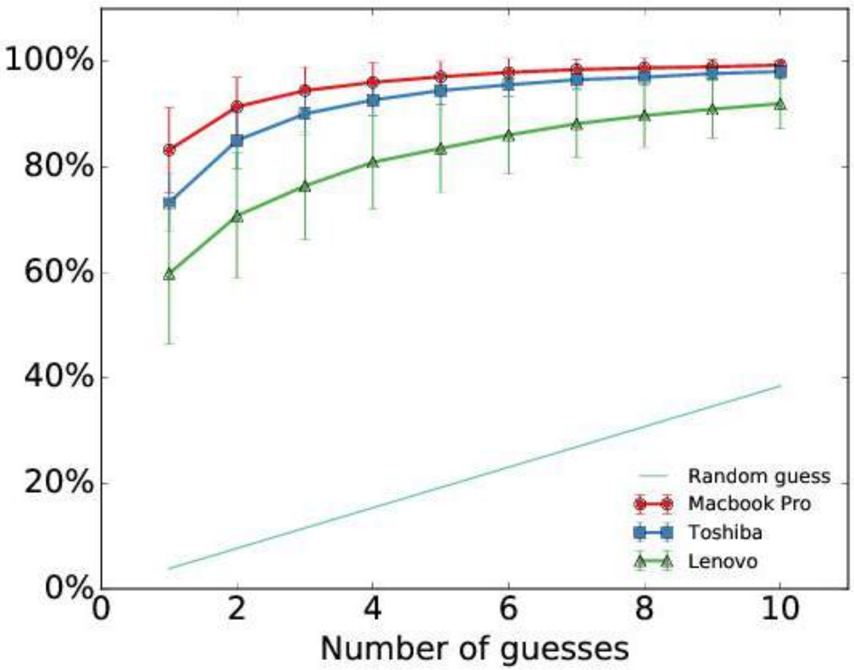
- Model Profiling Scenario
  - *Profiled a laptop of the same model on some users*
  - ***Victim is/can be unknown!***

Complete Profiling

Training set with the data the user disclosed



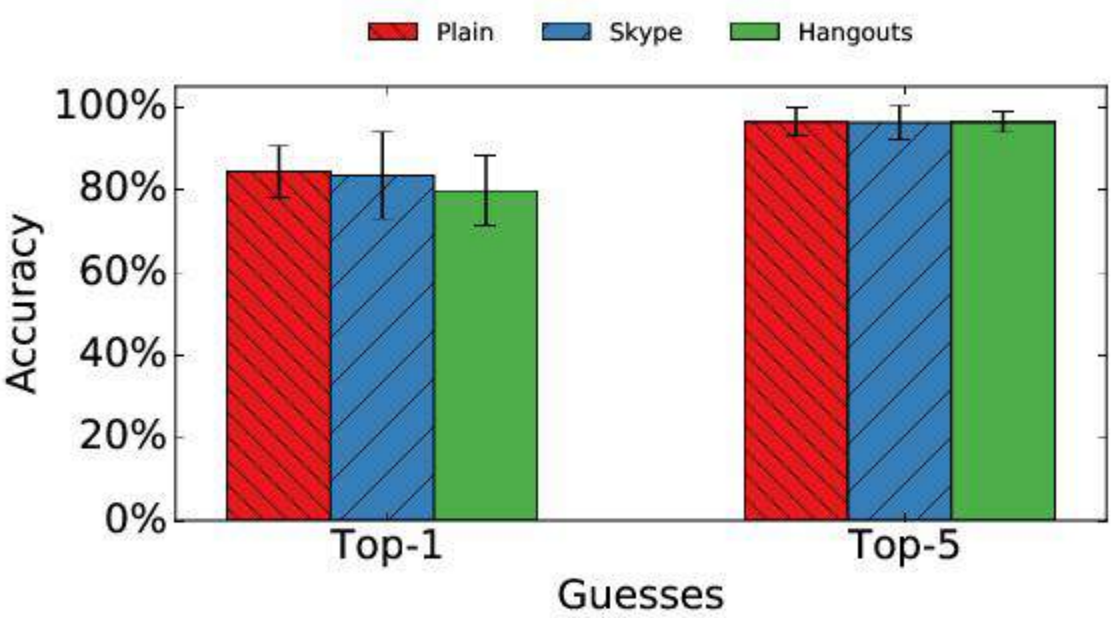
Hunt&Peck typing, unfiltered data



Touch typing, Skype filtered data

Complete Profiling

Is only Skype vulnerable to our attack?



No! It looks like a common problem for VoIP software

Model Profiling Setup

On the *Model Profiling* Scenario, the victim can be unknown

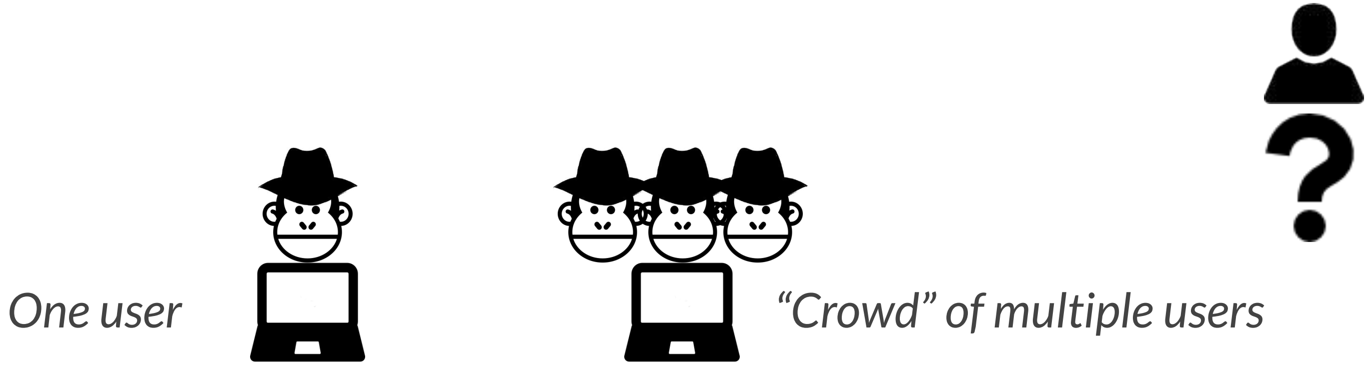
Someone the attacker does not know personally

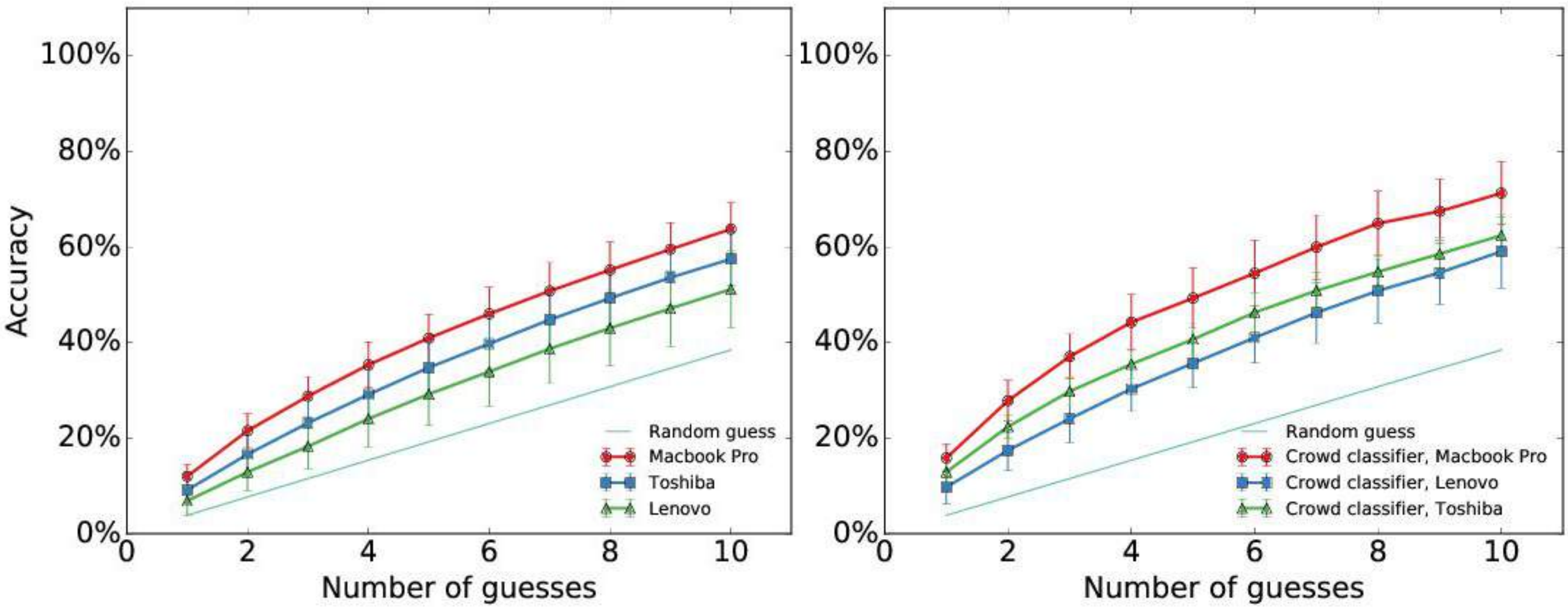
First need to understand the laptop of the victim

→ match it with a database of model signatures

- Guess correctly **93%** of the times if the model is known
- Statistical measures if the model is unknown

Model Profiling





Summing Up Our Results

- Recognize a single character
  - *Complete Profiling: 90%+ accuracy*
  - *Model Profiling: 40%+ accuracy*

- Recognize a single word
  - *Complete Profiling: 98% correct letters*
  - *Model Profiling: 50% correct letters*

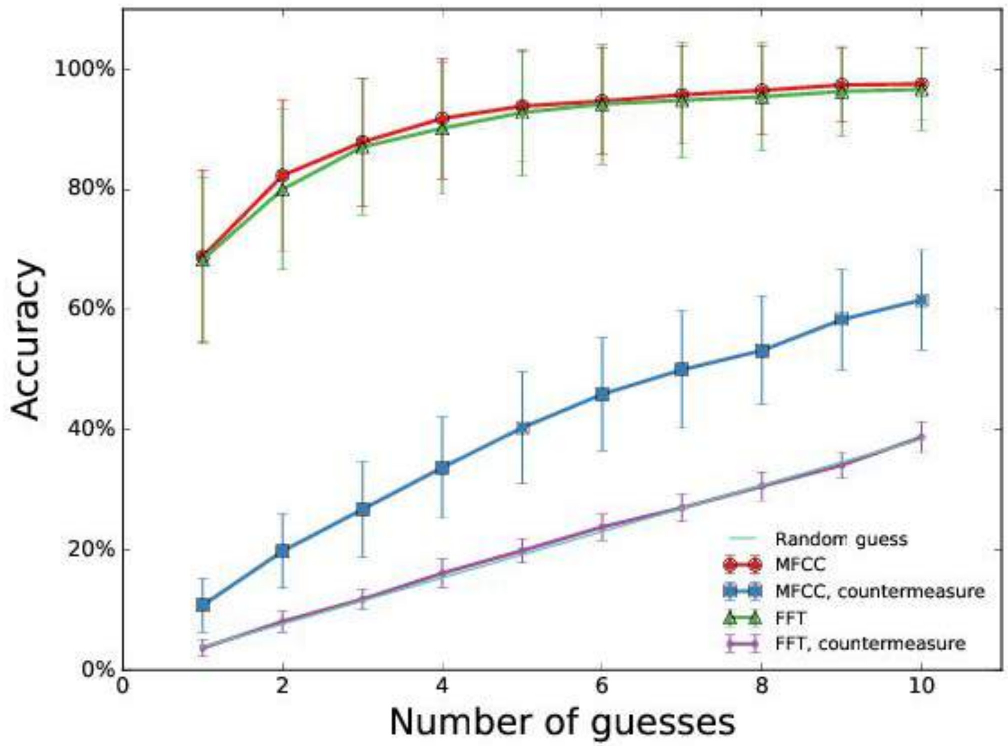
- Recognize a random password
  - *Improves 1-5 orders of magnitude time needed to guess the password*
  - *From 50 days to 42 seconds on a domestic PC*

Countermeasures

- <u>Don't Skype & Type</u>

- Remove volume when we detect a keypress sound
  - *Impacts voice, greatly degrades call quality*

- Disrupt spectral features with random equalization
  - *Assess impact on voice, real time feasibility*



Conclusions & Future Work

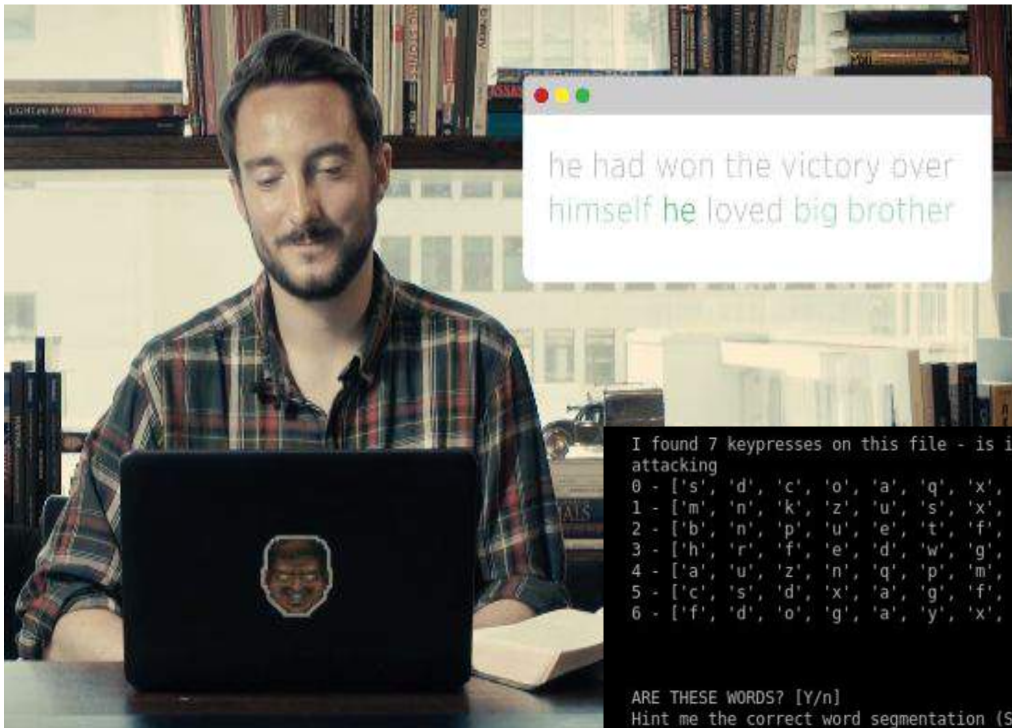
- VoIP Keyboard acoustic eavesdropping a serious threat

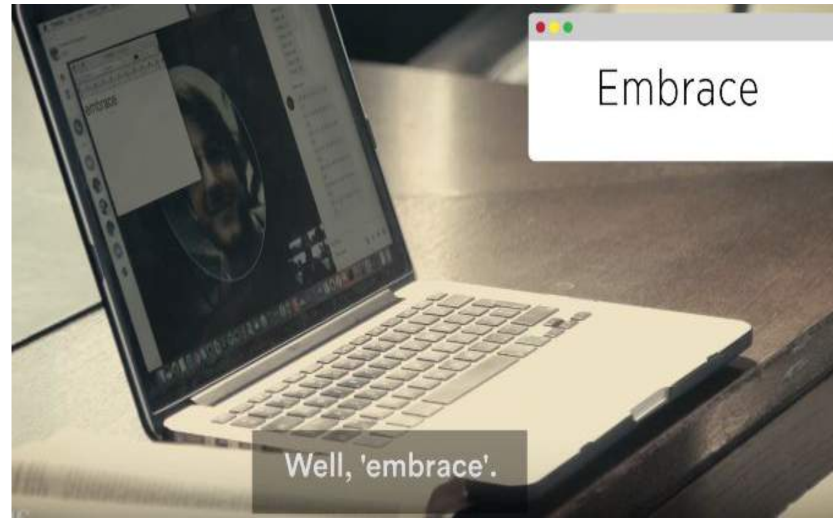
- Feasible and accurate:
  - *Realistic attack scenarios*
  - *91.71% on **Complete Profiling** scenario*
    - *Halevi (2012; 2014): 85.78%*
  - *41.89% on **Model Profiling** scenario*
    - *Novel attack vs. unknown victims*
  - *Robust to degradation and to voice*

- Future work:
  - *Try more users and different keyboards, and on more VoIP software*
  - *Try to attack another user in the same room*
  - *Analyze and improve the countermeasures*

Does it **really** work?

vs Forbes, 1984 & the Bible





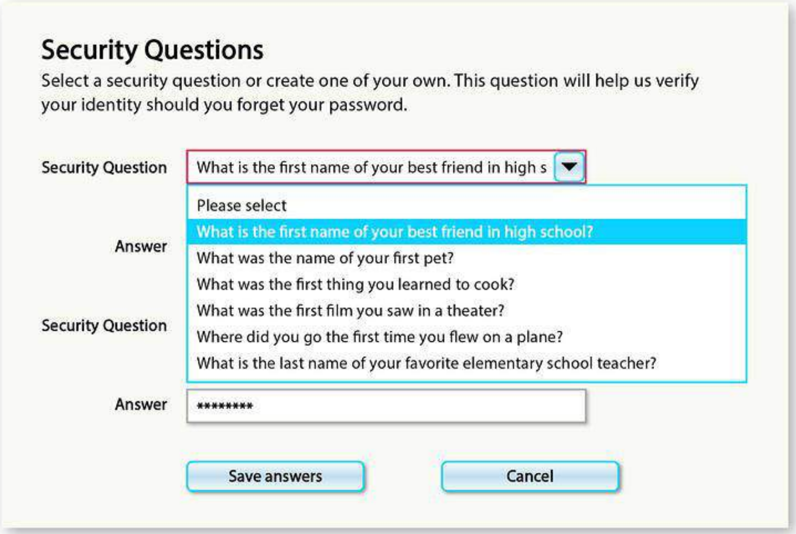

```
I found 7 keypresses on this file - is it correct? [Y/n]
attacking
0 - ['s', 'd', 'c', 'o', 'a', 'q', 'x', 'f', 'g']
1 - ['m', 'n', 'k', 'z', 'u', 's', 'x', 'i', 'a']
2 - ['b', 'n', 'p', 'u', 'e', 't', 'f', 's', 'v']
3 - ['h', 'r', 'f', 'e', 'd', 'w', 'g', 'p', 'c']
4 - ['a', 'u', 'z', 'n', 'q', 'p', 'm', 'c', 's']
5 - ['c', 's', 'd', 'x', 'a', 'g', 'f', 'k', 'z']
6 - ['f', 'd', 'o', 'g', 'a', 'y', 'x', 'h', 'c']



ARE THESE WORDS? [Y/n]
Hint me the correct word segmentation (Suggested spaces in []):
[('embrace', 21), ('surface', 26), ('conduct', 28), ('disease', 29), ('attract', 30), ('courage', 31), ('fantasy', 32), ('contact', 3
3), ('intense', 33), ('library', 33), ('silence', 33), ('already', 34), ('average', 34), ('defense', 34), ('impress', 34), ('subject'
, 34), ('suppose', 34), ('discuss', 35), ('expense', 35), ('offense', 36), ('science', 36), ('storage', 36), ('absence', 37), ('stoma
ch', 37), ('finance', 38), ('operate', 38), ('overall', 38), ('suspect', 38), ('century', 39), ('funding', 39)]
```

Forbes Credits: https://www.forbes.com/sites/thomasbrewster/2017/07/06/skype-and-type-attack-steals-passwords

Thank you!

Questions?

(if you do not have one, please find some suggestions below)

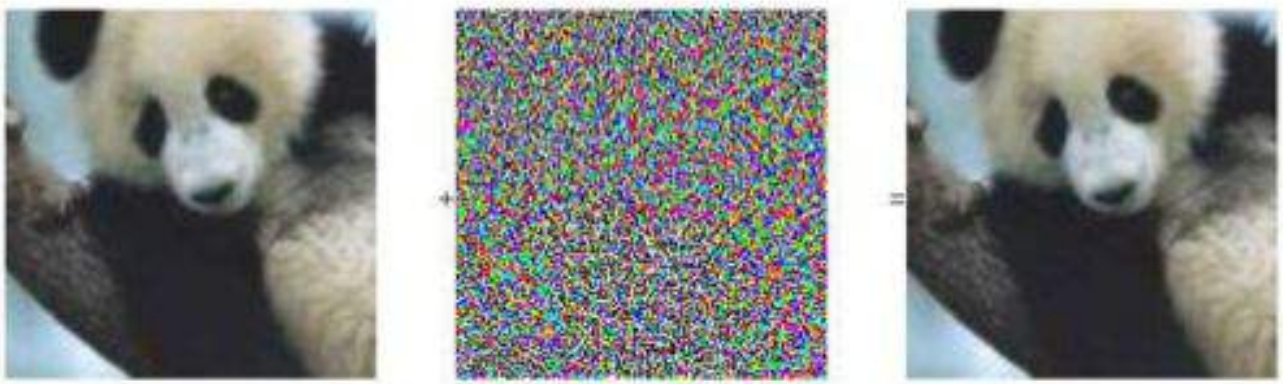


This is the END!

Backup Slides
after this point… ;-)

Adversarial Machine Learning

Adversarial Examples (Deep Learning/CNNs)



Original image classified as a panda with 60% confidence.

Tiny adversarial perturbation.

Imperceptibly modified image, classified as a gibbon with 99% confidence.

http://www.kdnuggets.com/2015/07/deep-learning-adversarial-examples-misconceptions.html

http://karpathy.github.io/2015/03/30/breaking-convnets/

Classification Accuracy

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

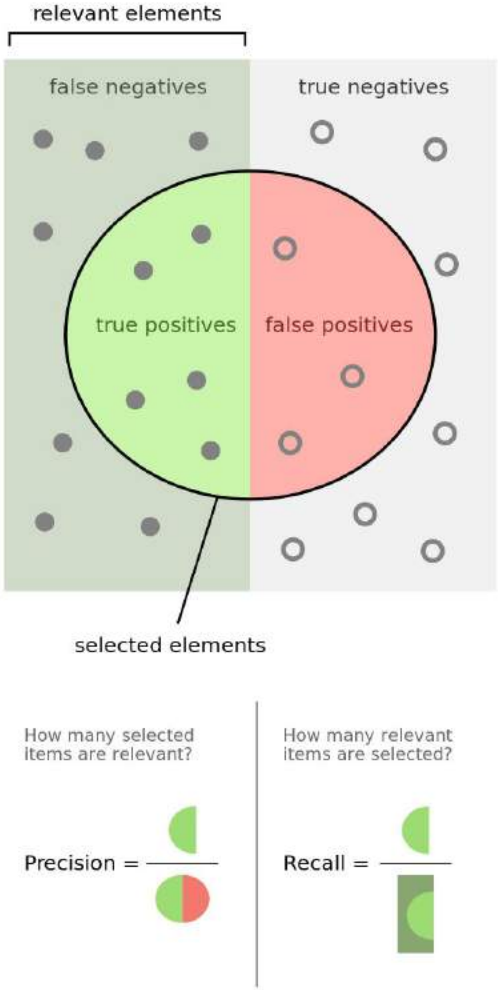
It works well only if there are equal number of samples belonging to each class.

For example, consider that there are 98% samples of class A and 2% samples of class B in our training set. Then our model can easily get **98% training accuracy** by simply predicting every training sample belonging to class A.

When the same model is tested on a test set with 60% samples of class A and 40% samples of class B, then the **test accuracy would drop down to 60%.** Classification Accuracy is great, but gives us the false sense of achieving high accuracy.

SPRITZ Security & Privacy Research Group
Università degli Studi di Padova

- Precision, Recall, and F-measure



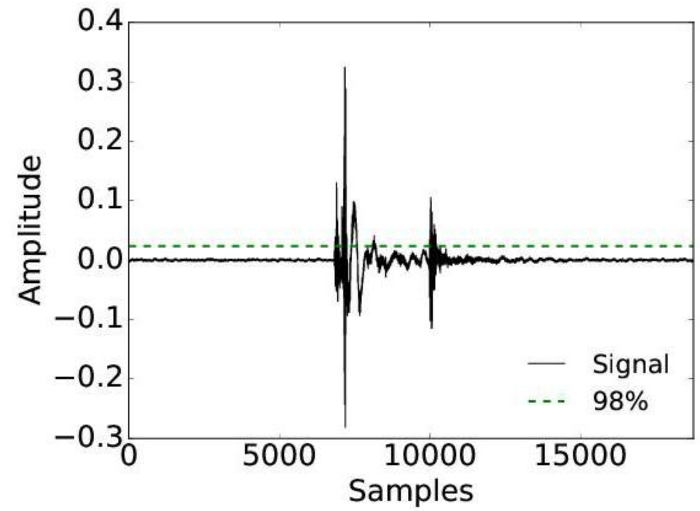
$$F_1 = 2 \cdot \frac{1}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

$$F_\beta = \frac{(1+\beta^2) \cdot \mathrm{true\ positive}}{(1+\beta^2) \cdot \mathrm{true\ positive} + \beta^2 \cdot \mathrm{false\ negative} + \mathrm{false\ positive}}$$

Attack - Data Processing

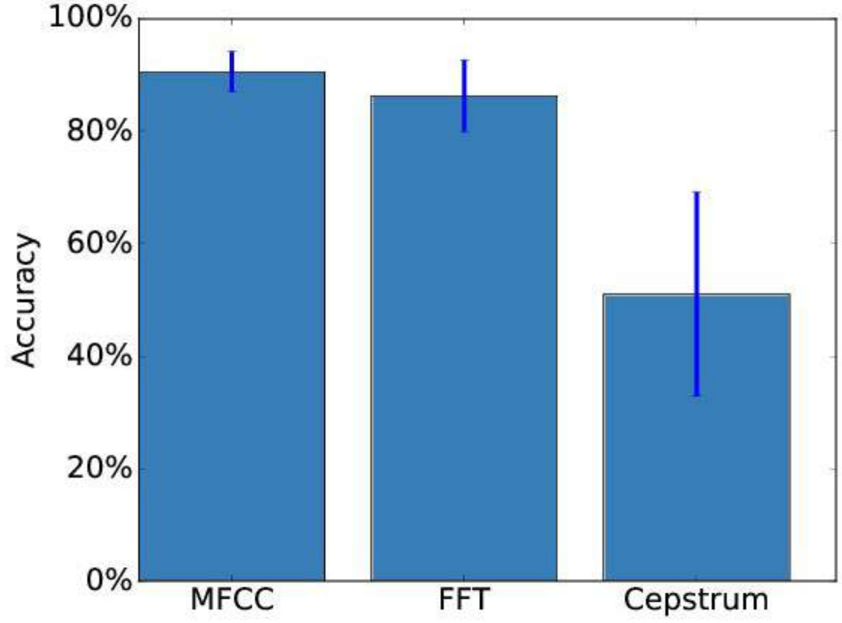
- Data windowing and segmentation

  To extract sound samples



- Feature extraction: *mel frequency cepstral coefficients*

  Selected with a preliminary experiment



Novelty - Attack Scenarios

SPRITZ Security & Privacy Research Group
Università degli Studi di Padova

Evaluate the attack on three different realistic scenarios

- **Complete Profiling Scenario** (Asonov, 2004; Halevi, 2012; 2014)
    - *Profiled the user on his laptop → specific training set*
    - *Ground truth disclosure, e.g., a short chat message*

- **User Profiling Scenario**
    - *Profiled the user on a different laptop*
    - *Social engineering techniques*

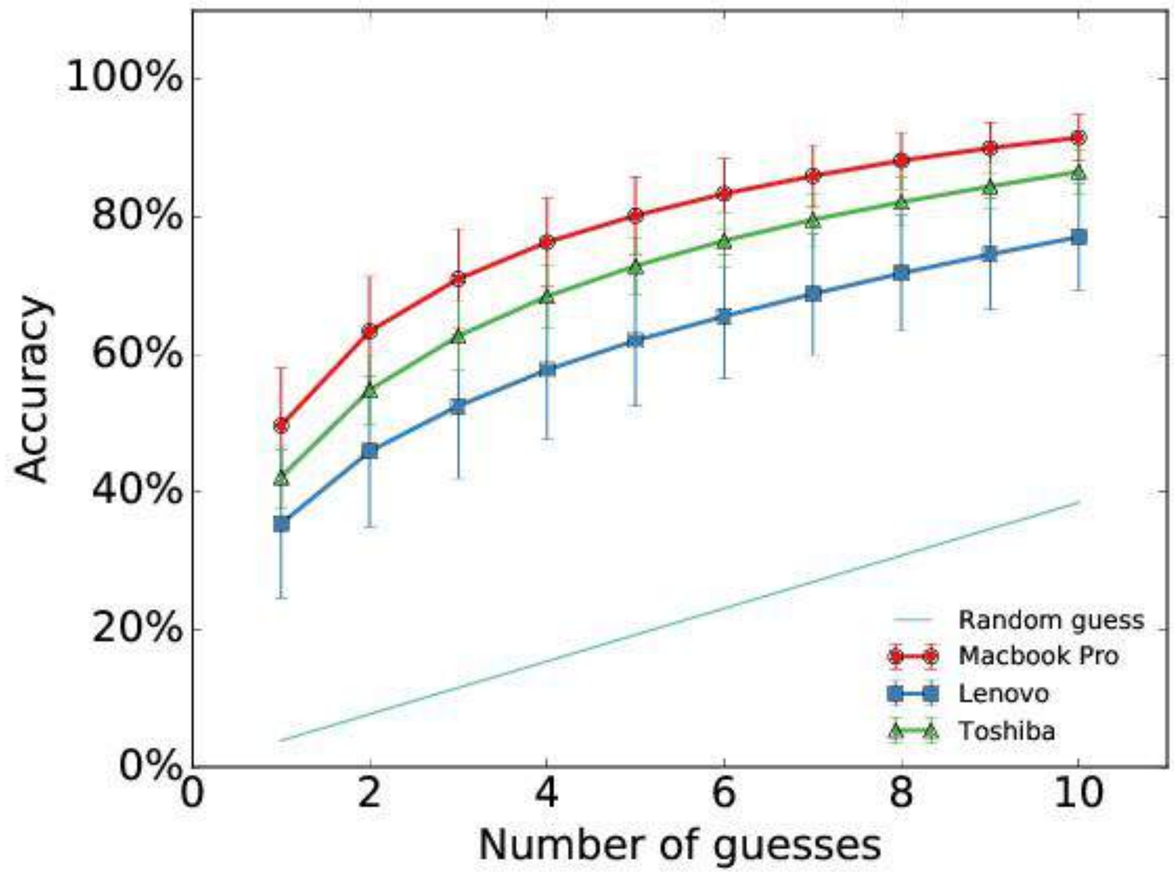
- **Model Profiling Scenario**
    - *Profiled a laptop of the same model on some users*
    - *The victim can be unknown*

Evaluation - Small Training Set

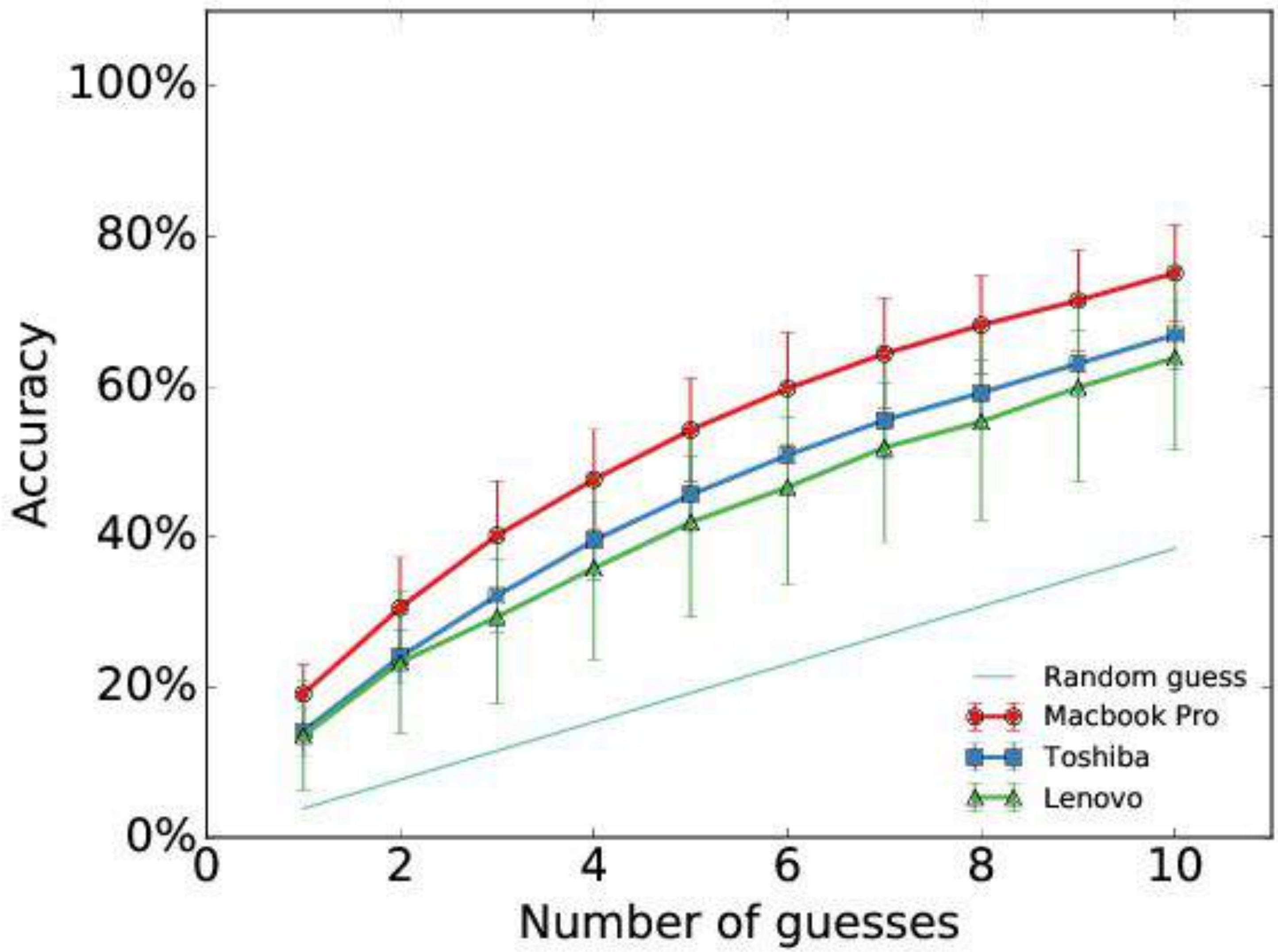
10 samples/character aren't your typical chat message

***Training set** with realistic letter frequencies*
***Test** against random password*

Character	*# Samples*
E	10
A	9
R	7
J	1
Z	1



Evaluation - User Profiling



Password Cracking

The goal was to crack the victim's <u>random</u> password

→ We need bruteforce techniques

Random password of 10 lowercase letters

- $\log_2(26^{10}) = 47$ bits of entropy

On the Complete Profiling Scenario (high accuracy)

- $\log_2(5^{10}) = 23.22$ bits of entropy

On the other scenarios - entropy is not meaningful

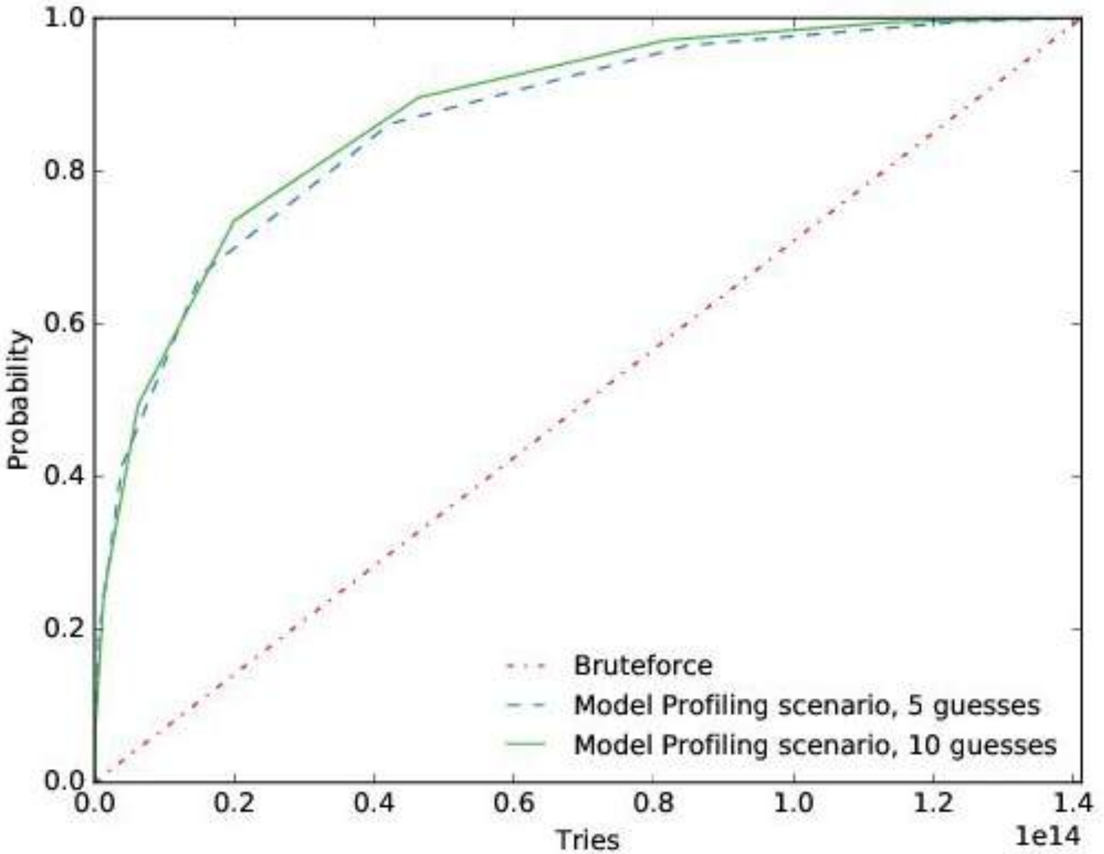
SPRITZ Security & Privacy Research Group

Università degli Studi di Padova

Model Profiling Scenario → improved bruteforce

Take into account character probabilities

Evaluate the reduction of the average number of trials



Features

Fast Fourier Transform coefficients

$$S(f(t)) = 20\log_{10}\left(|\mathcal{F}(f(t))|\right)$$

$f(t) = \text{signal}$
$\mathcal{F} = \text{Discrete Fourier Transform function}$

Cepstrum coefficients

$$C(f(t)) = \left|\mathcal{F}^{-1}(S(f(t)))\right|^2$$

Mel frequency cepstral coefficients

$$MFC(f(t)) = DCT\left(\log_{10}\left(mel\{|\mathcal{F}(f(t))|\}\right)\right)$$

$$mel(f) = 2595\log_{10}\left(1 + \frac{f}{700}\right)$$

$DCT = \text{Discrete Cosine Transform}$

Side and Covert Channels: *the Dr. Jekyll and Mr Hyde of Modern Technologies*

Mauro Conti

2020 WiseML @ WiSec

July 13 2020

SPRITZ
Security & Privacy
Research Group

Università
degli Studi
di Padova