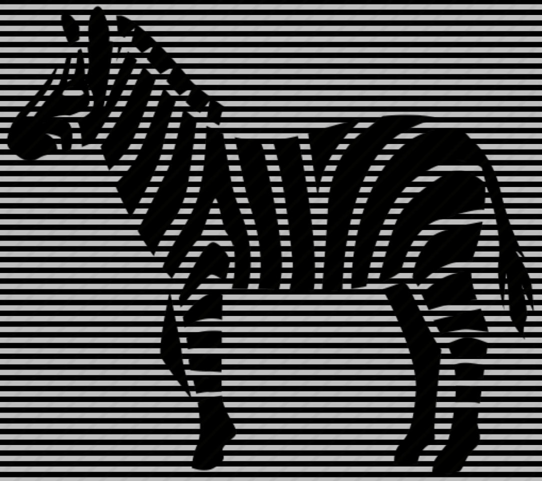
Synesthesia



Daniel Genkin, University of Michigan
genkin@umich.edu

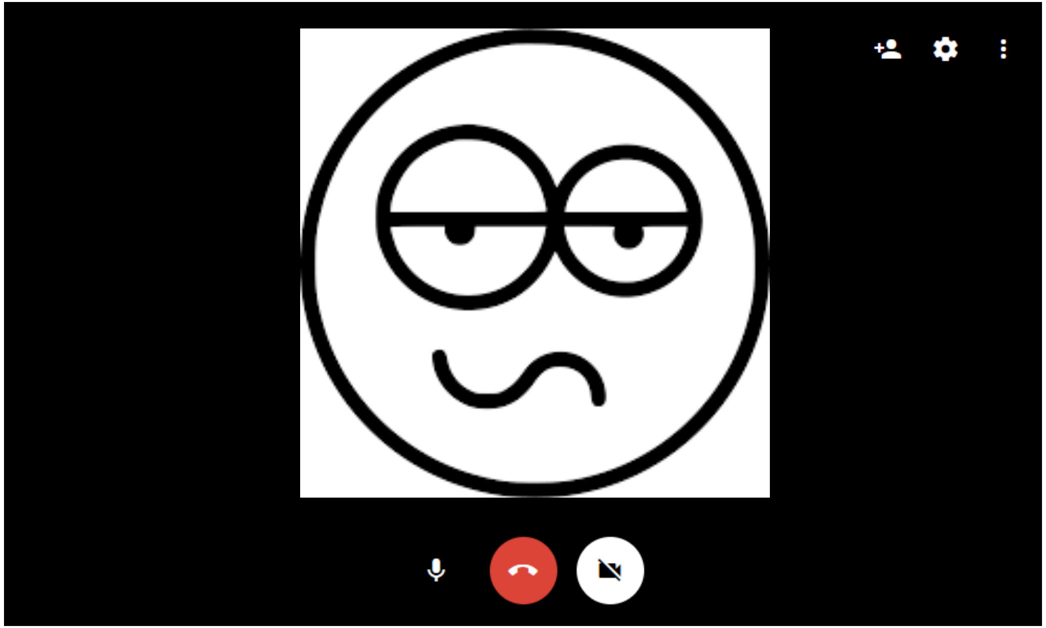
Mihir Pattani, University of Pennsilvania
mihirpattani14@gmail.com

Roei Schuster, Tel Aviv University and Cornell Tech
rs864@cornell.edu

Eran Tromer, Tel Aviv University and Columbia University
tromer@tau.ac.il

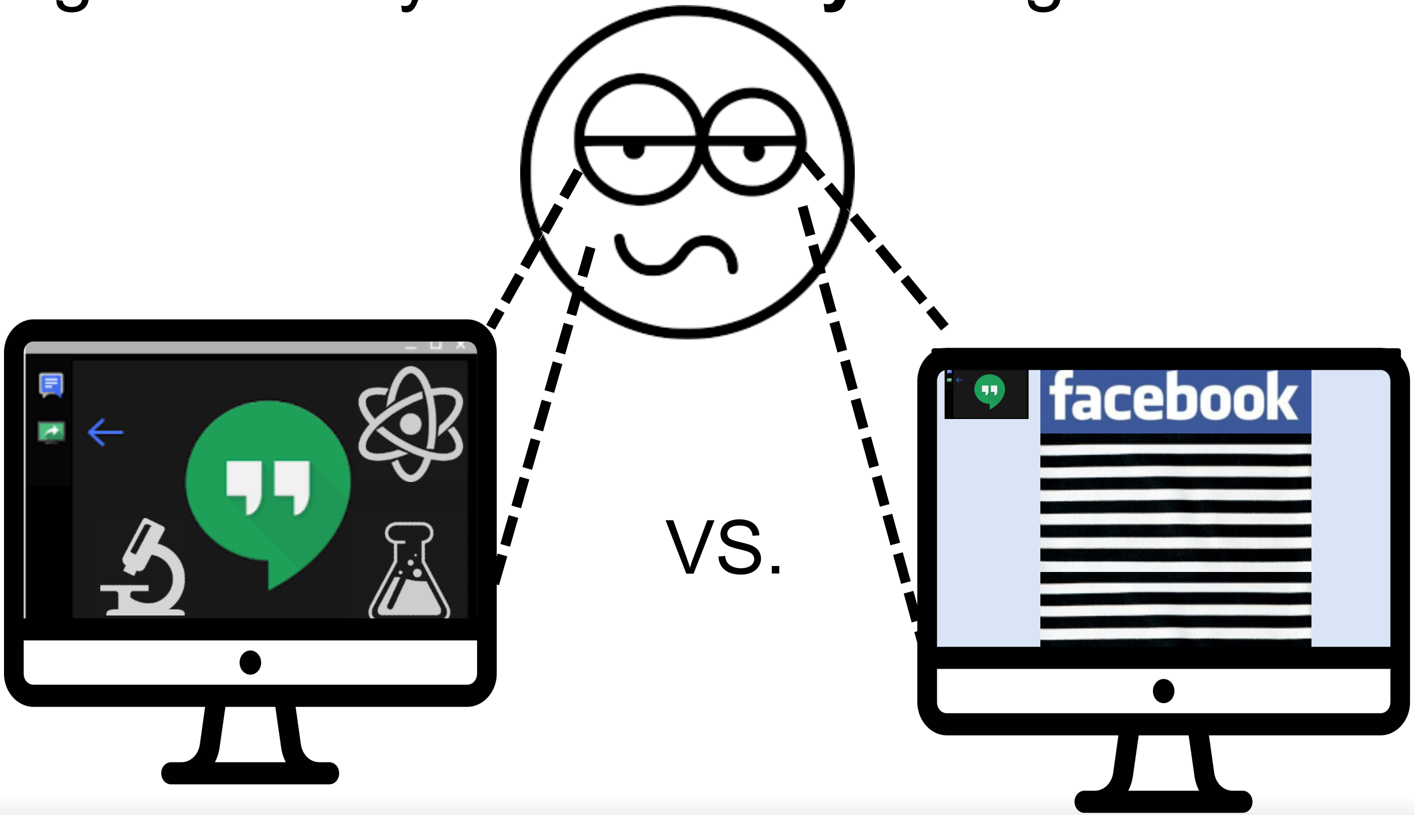
The problem

- Many colleagues appear blandly disengaged during crucial video-conference calls

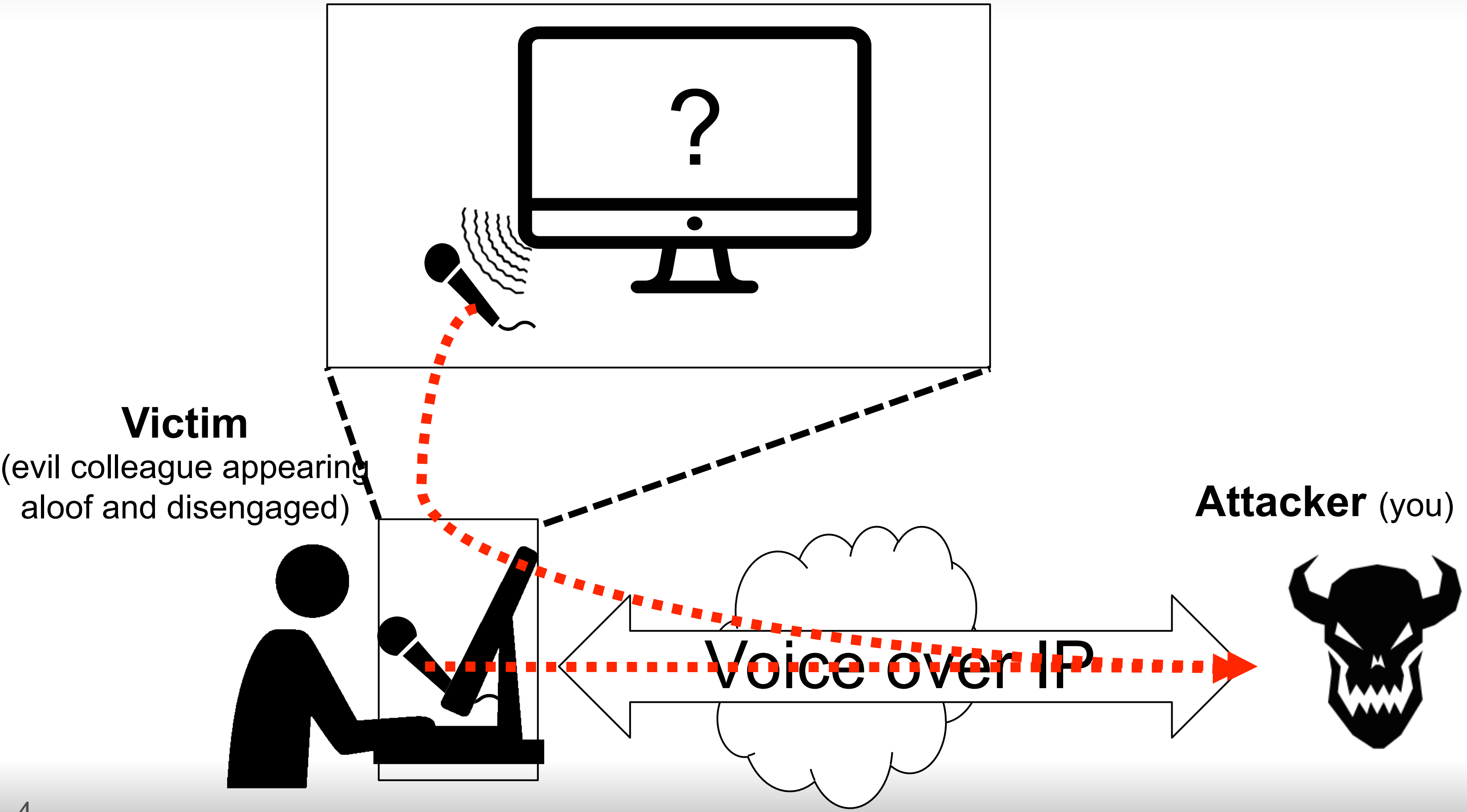


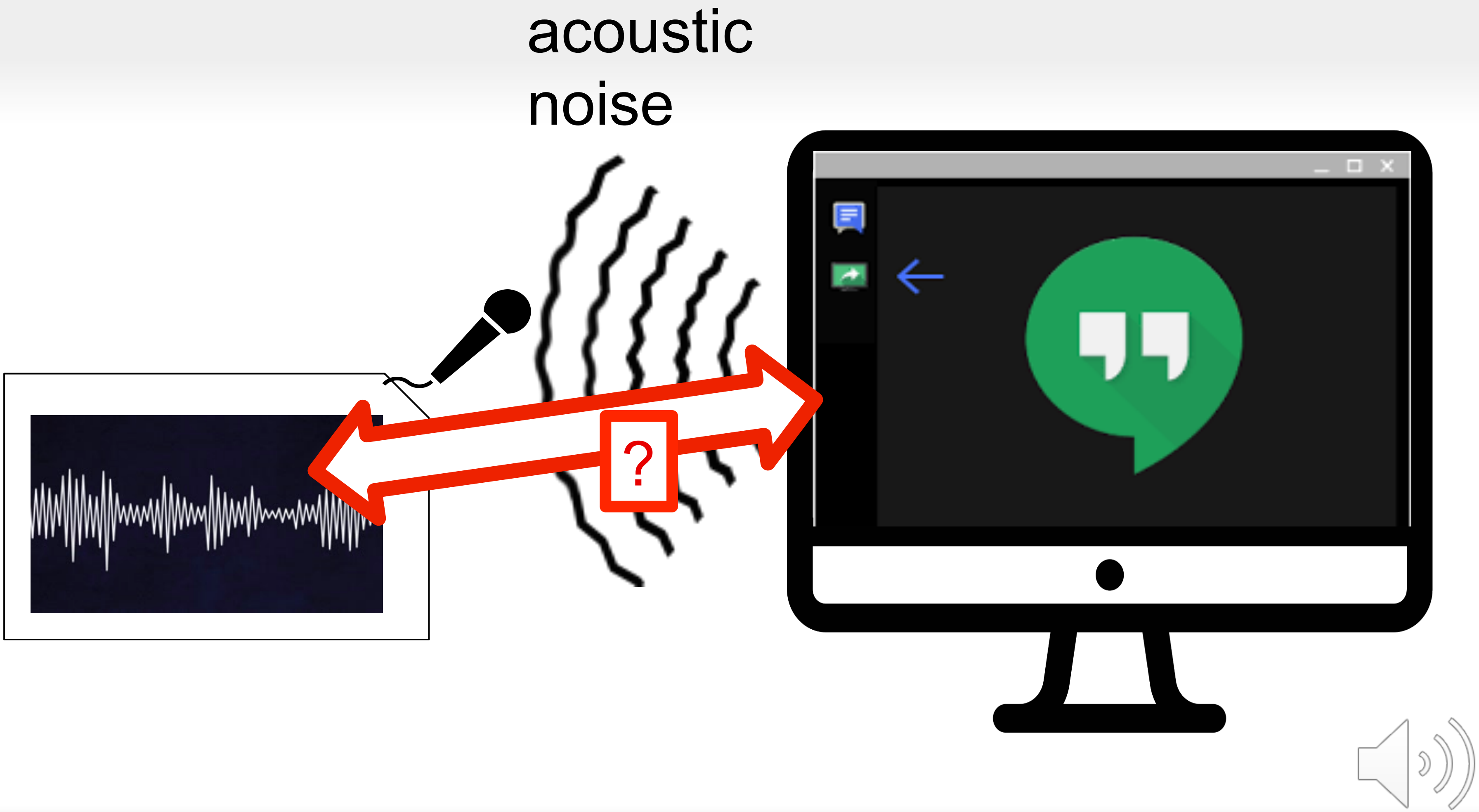
The challenge

- Telling what they are **actually** doing…



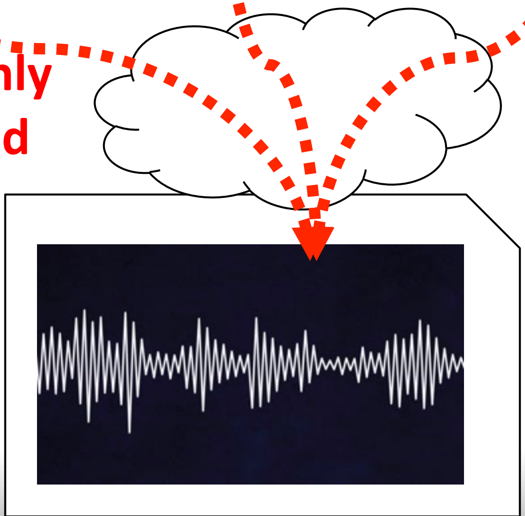
Idea: “hear” the screen




acoustic
noise
?

Acoustic leakage from screens is dangerous

Microphones are ubiquitous



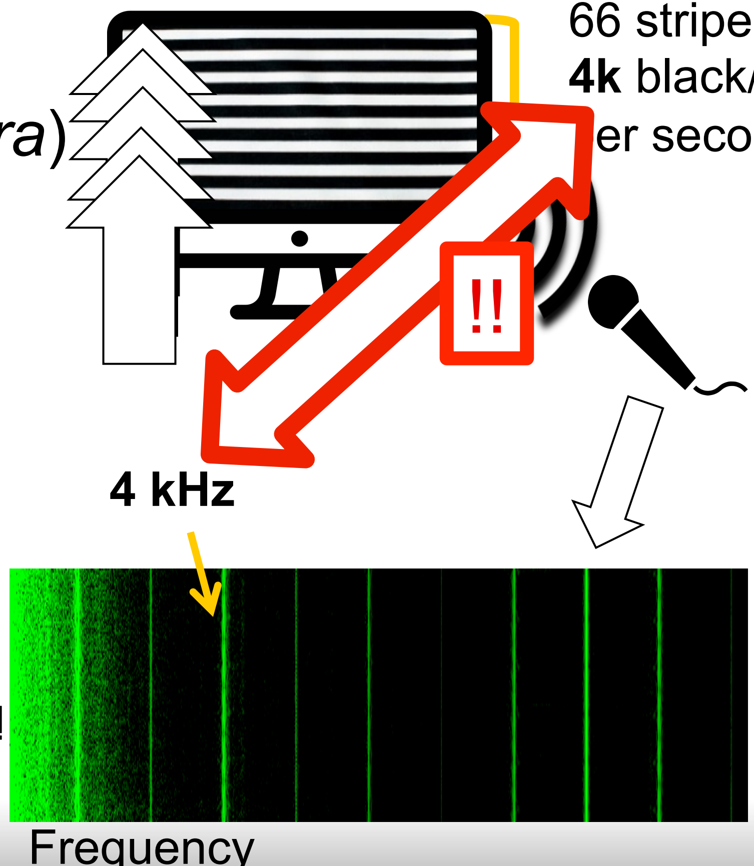
Acoustic leakage highly available compared to electromagnetic leakage

[Eck’85][Kuh’04]

monly
tored
ing
on-screen
content?

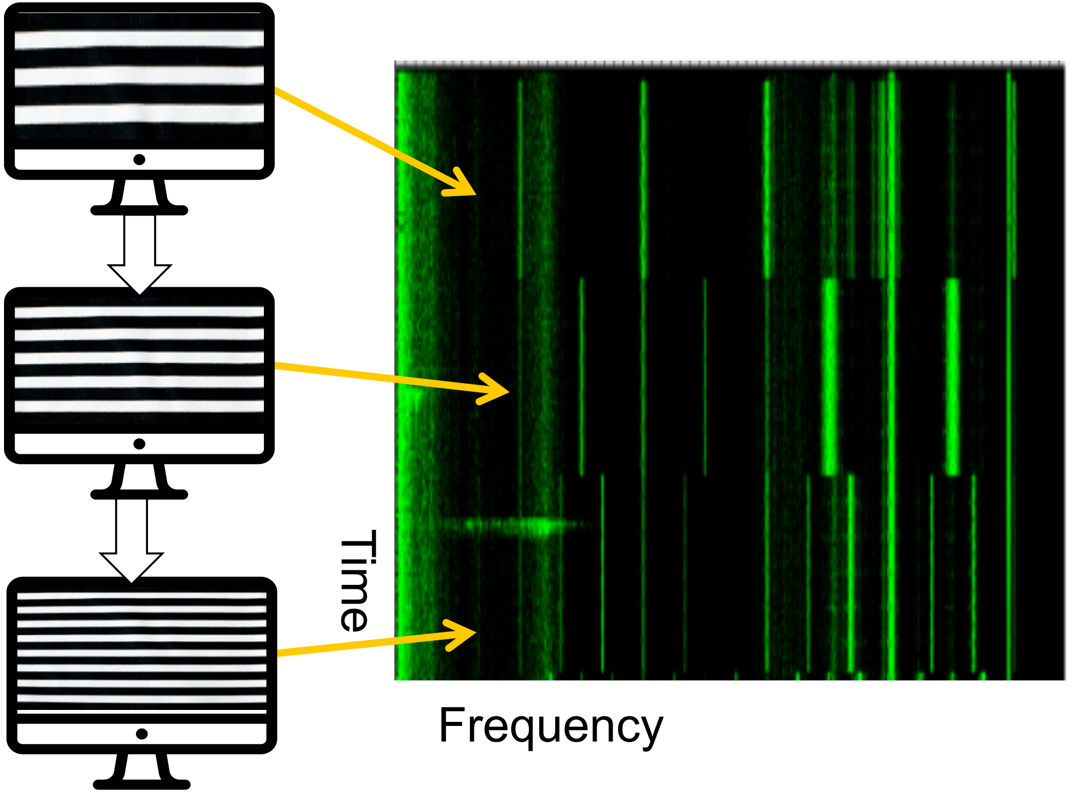
Detecting leakage: “see a Zebra”

pixel color transitions (*Zebra*)

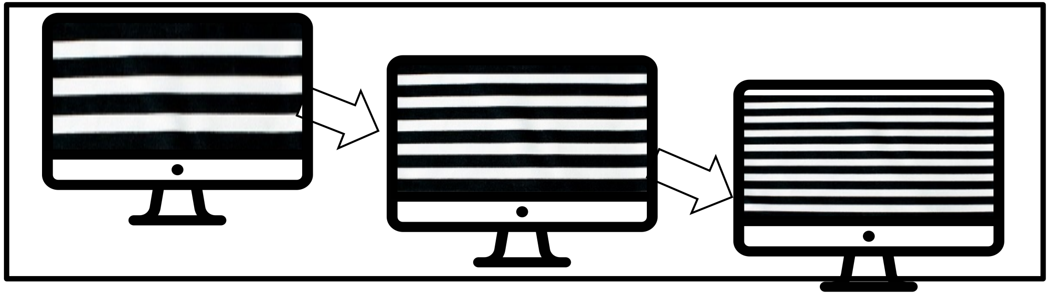


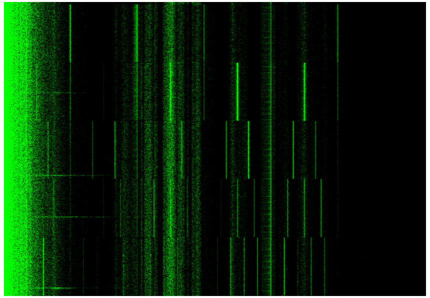
66 stripes x 60 refresh per second =
4k black/white transitions
er second

Changing stripe width

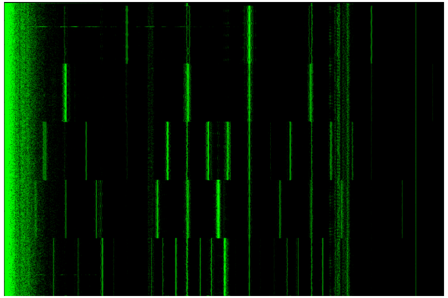


Leakage pattern consistent across makes/models

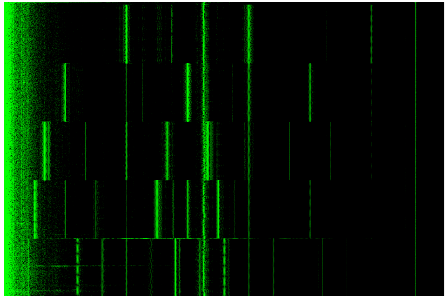




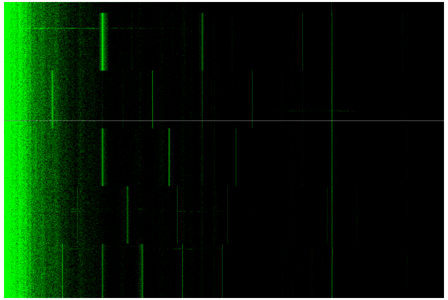
SAMSUNG 920NW



hp ZR30w

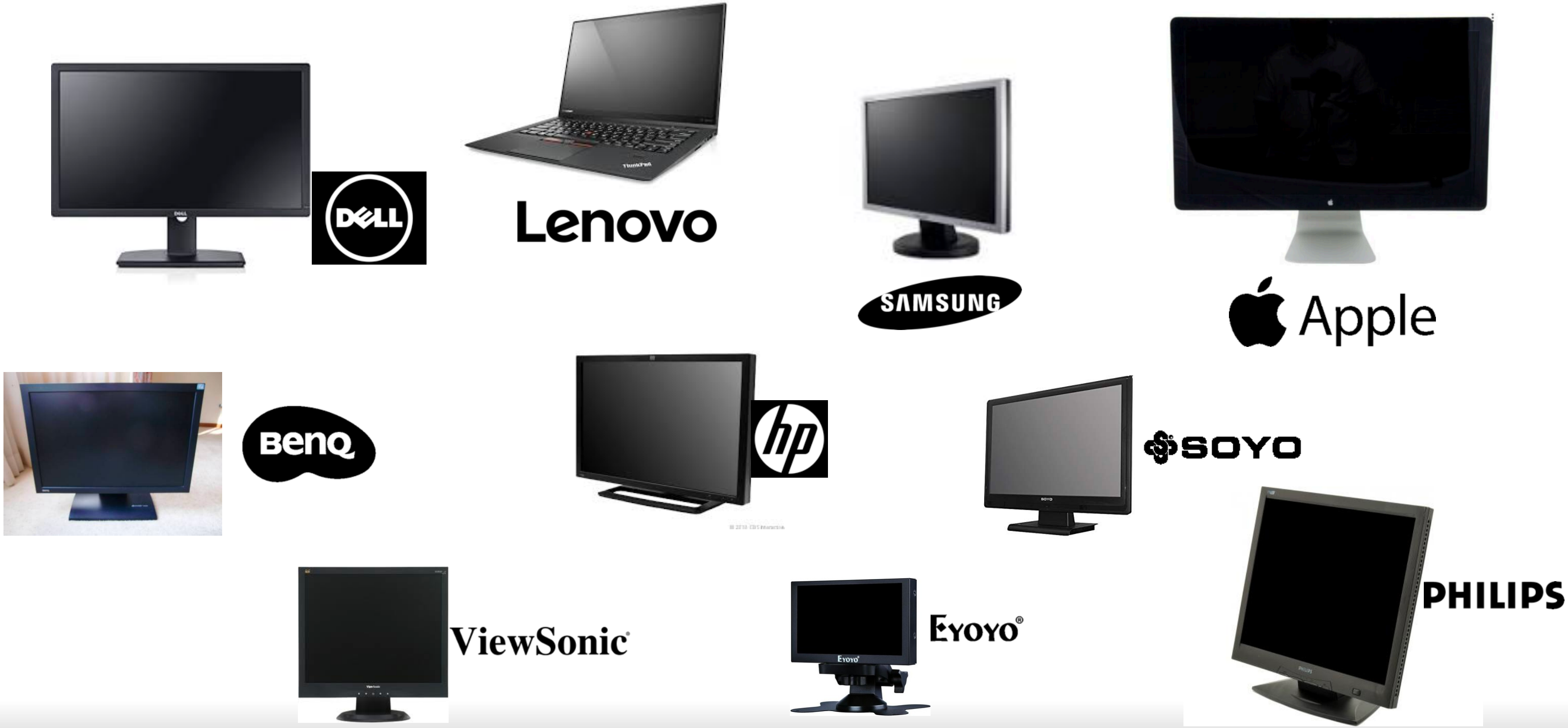


DELL U3011t

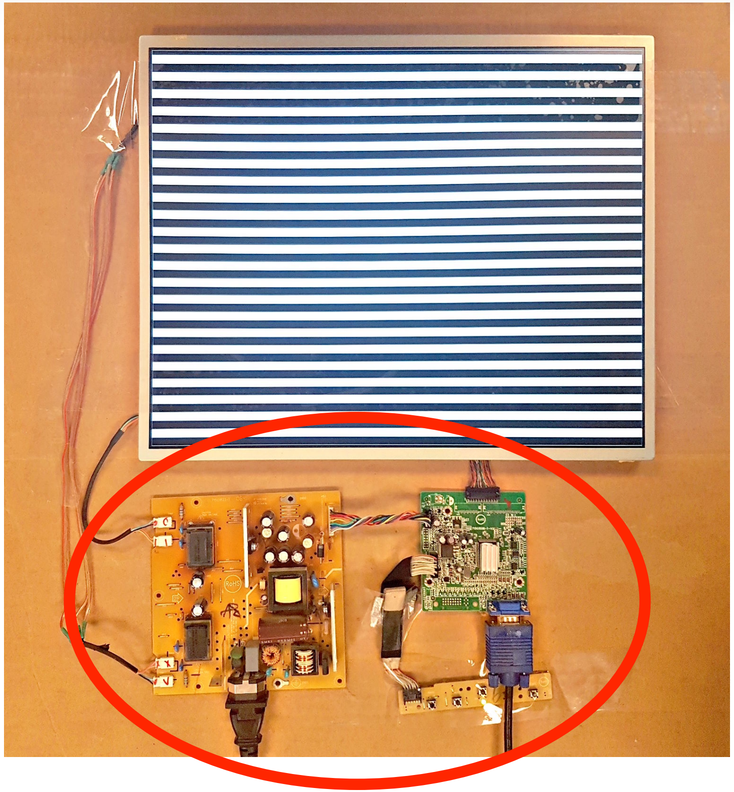


PHILIPS 170S4

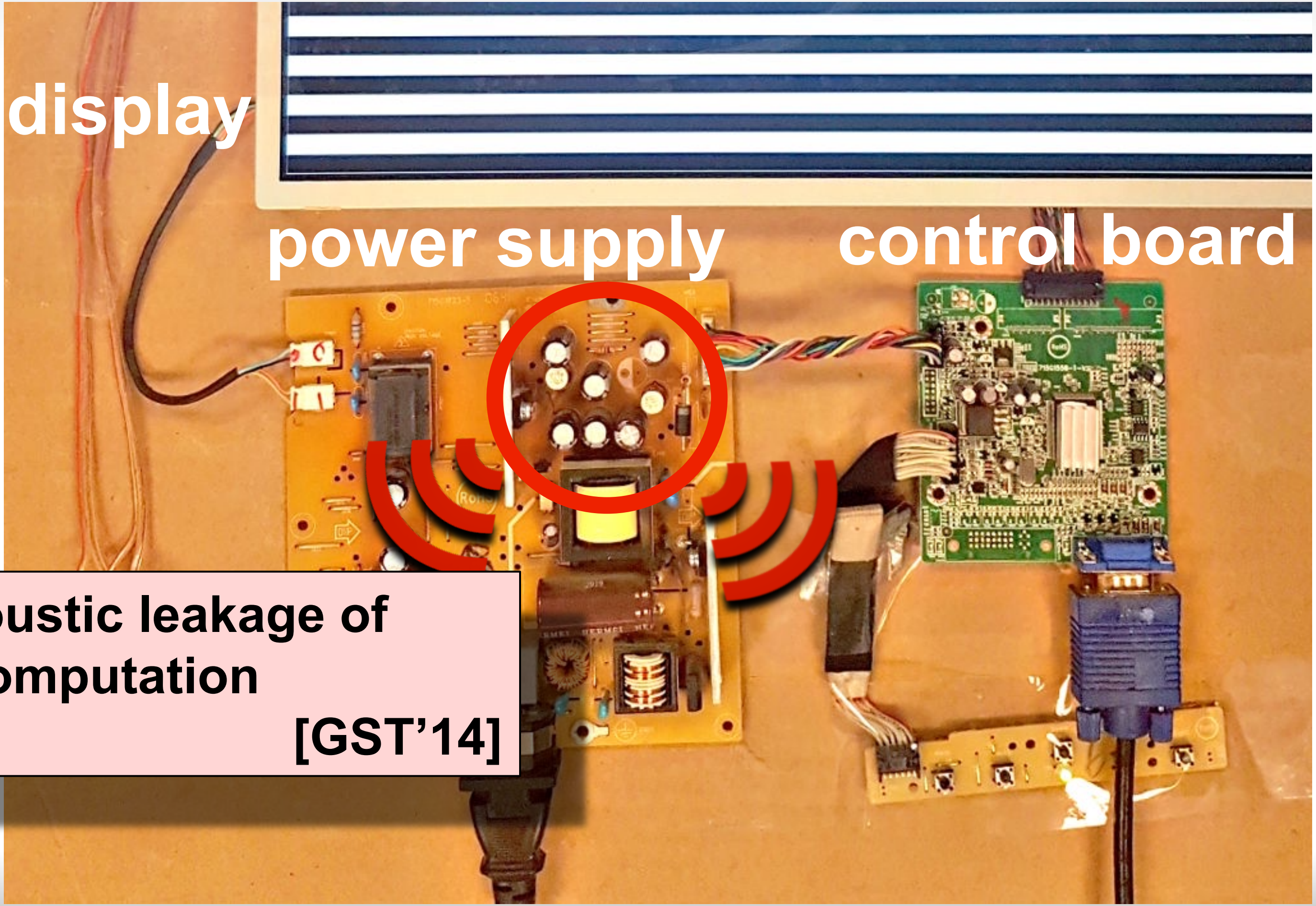
Leakage pattern consistent across **many** makes/ models



Whence acoustic leakage?

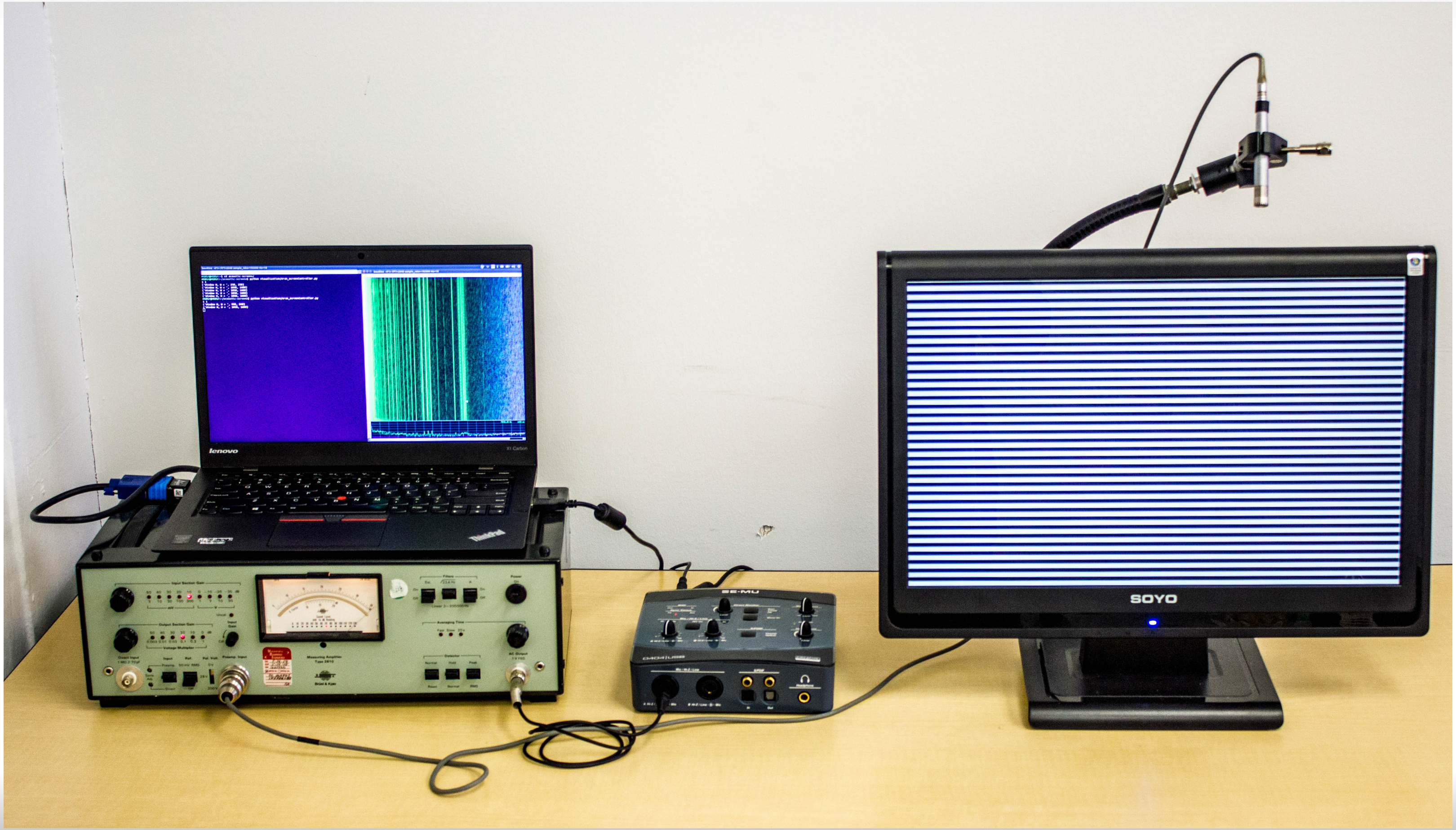


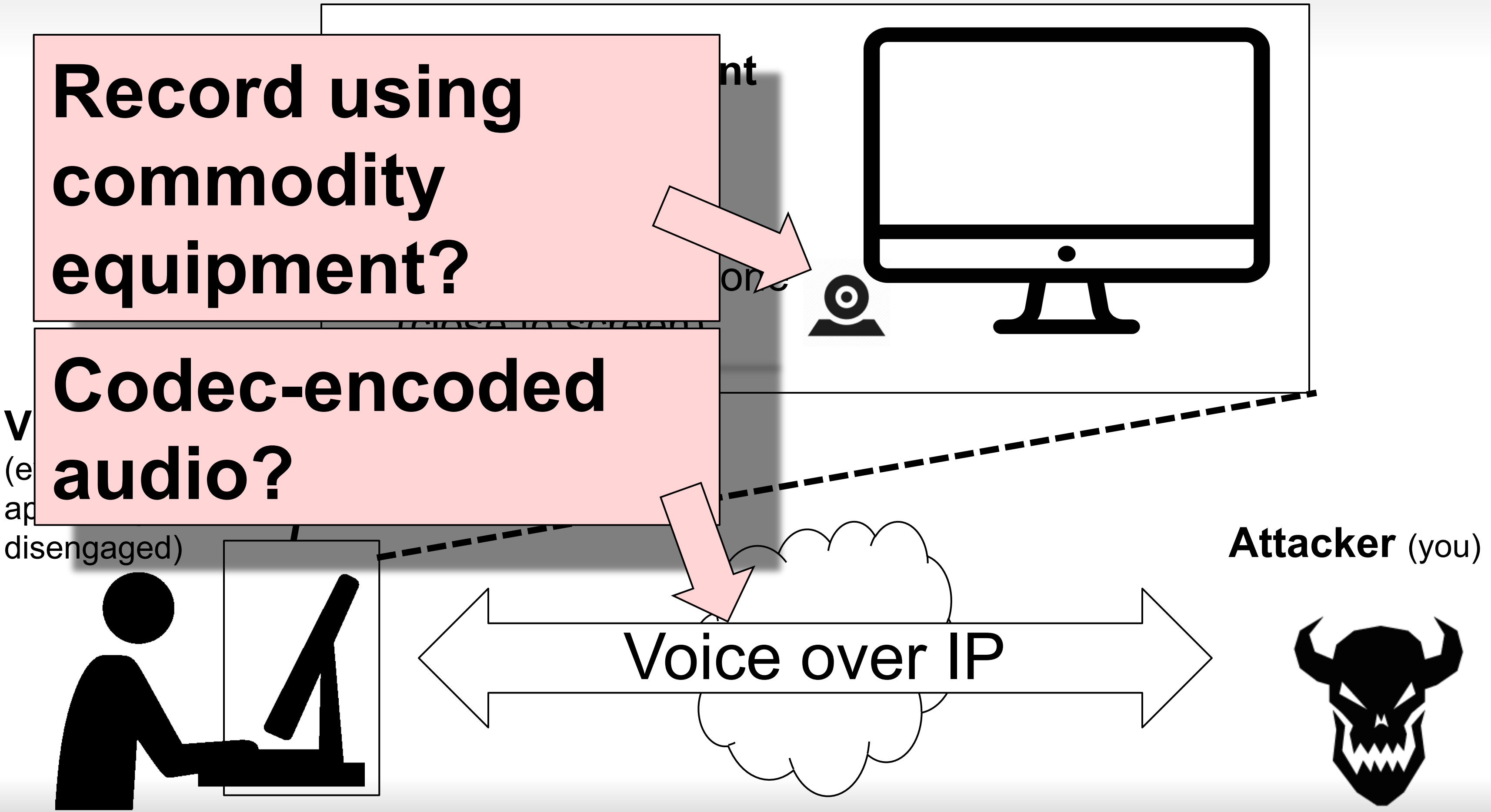
Whence acoustic leakage?



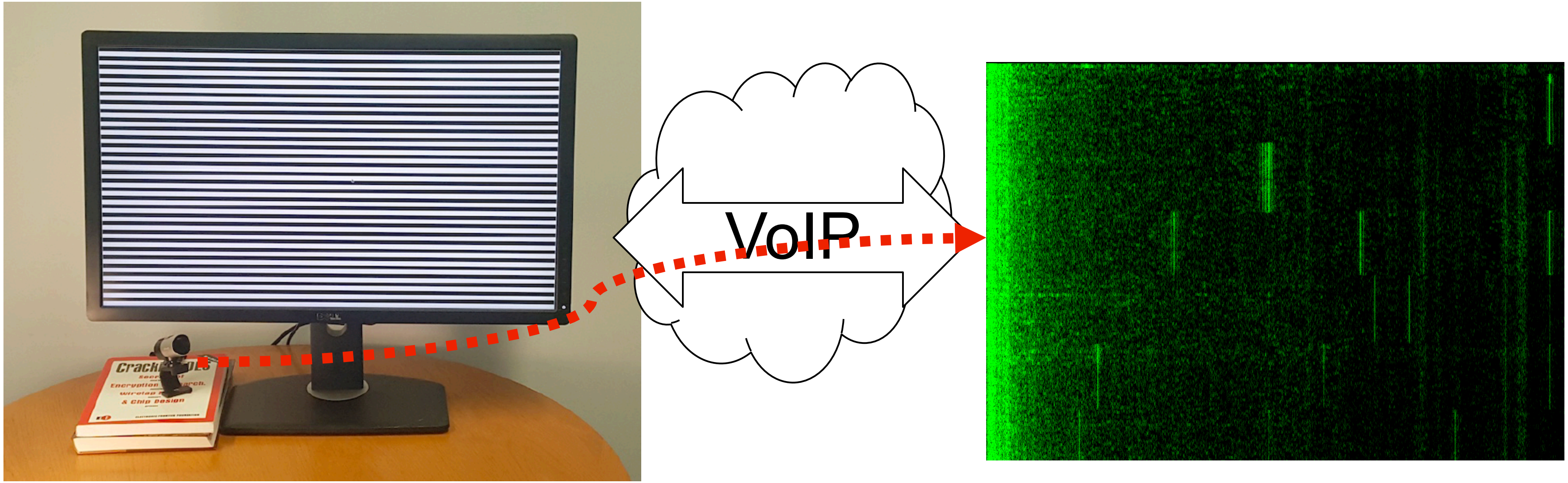
vs. acoustic leakage of CPU computation
[GST'14]

So far: lab conditions

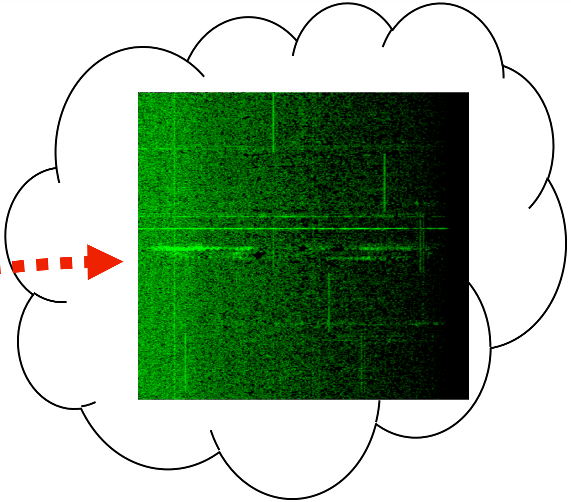



Record using commodity equipment?
Codec-encoded audio?
disengaged)
Voice over IP
Attacker (you)

Codec-encoded VoIP (Google Hangouts)

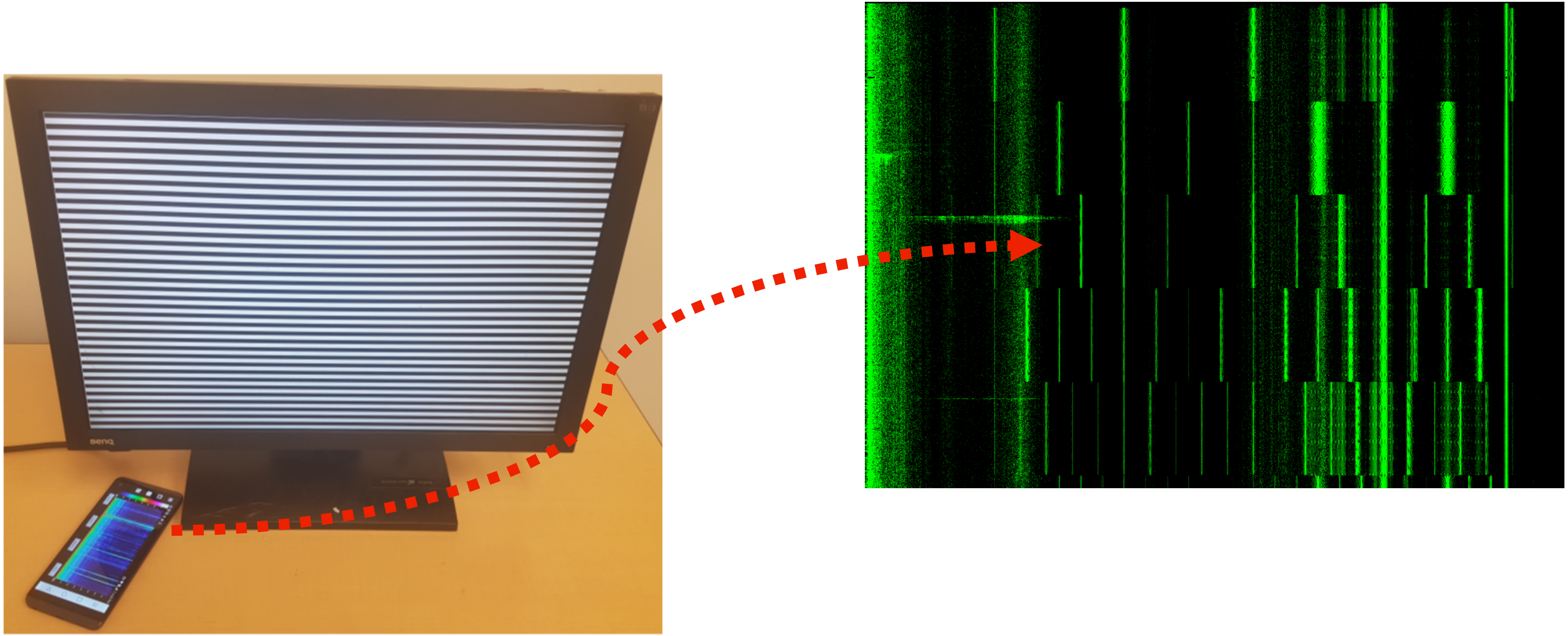


Recordings uploaded to the cloud

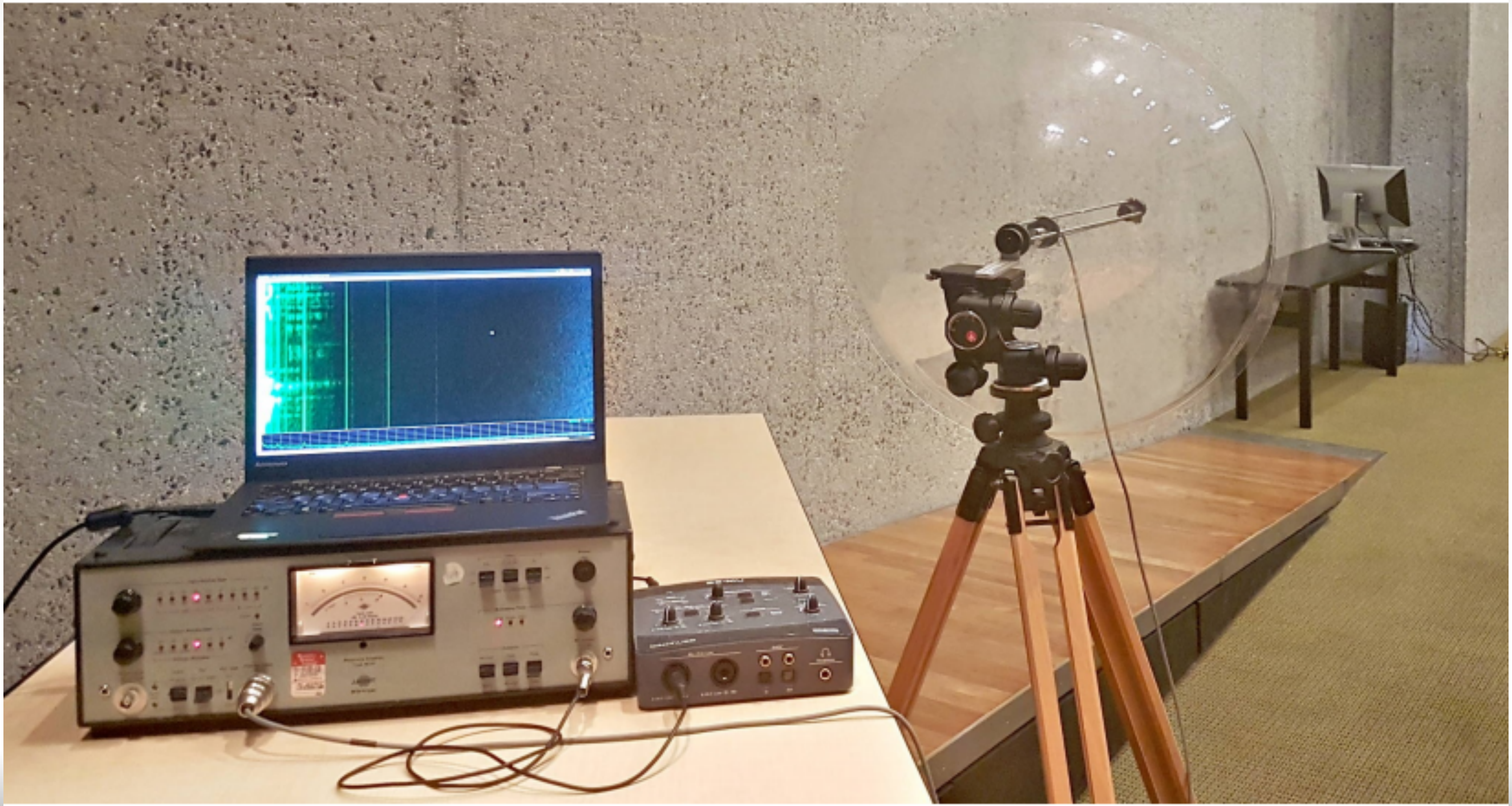


Leakage still detectible in
cloud-archived recordings!

Smart phone

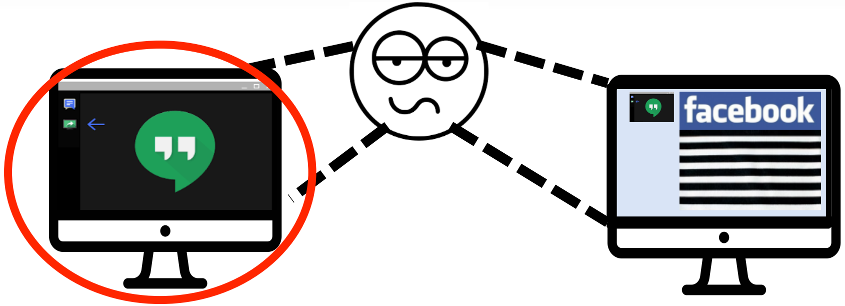


Attack at a distance (using a parabolic dish)

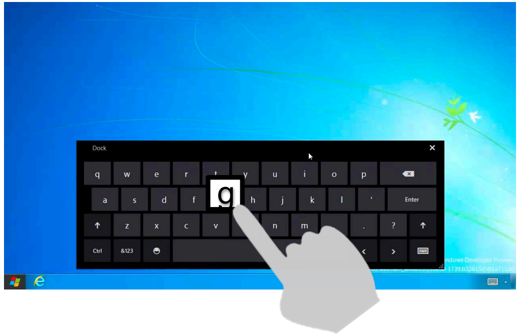


What can an attacker do?

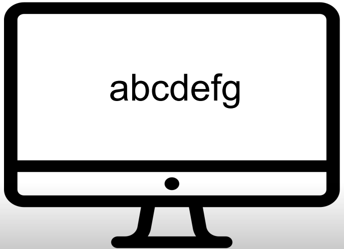
- Activity/website distinguishing



- On-screen keyboard snooping



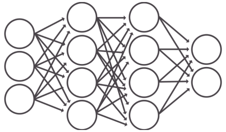
- Text extraction



How?

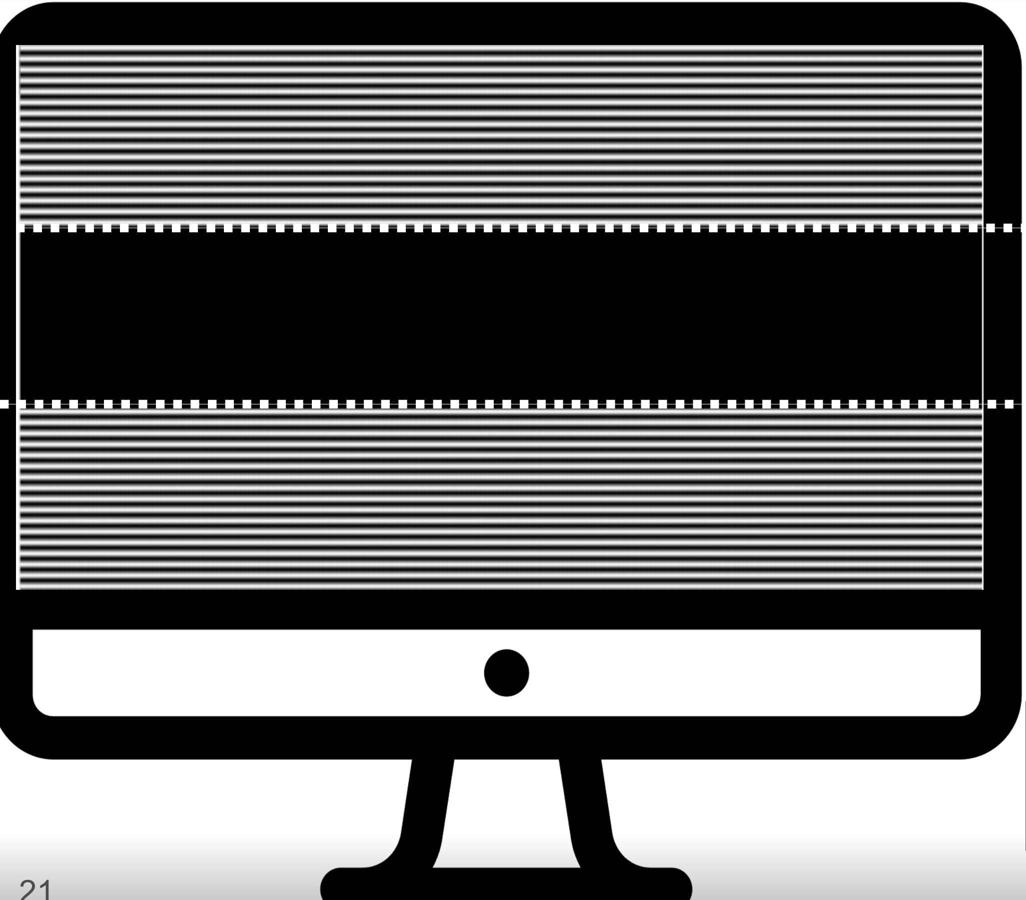
1. denoising

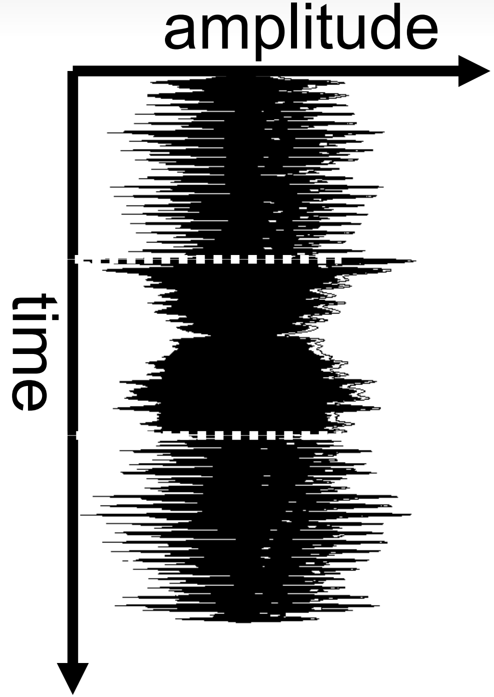
2. ML-based attacks



- Website distinguishing
- On-screen keyboard snoop
- Text extraction

Observation (1): amplitude modulation





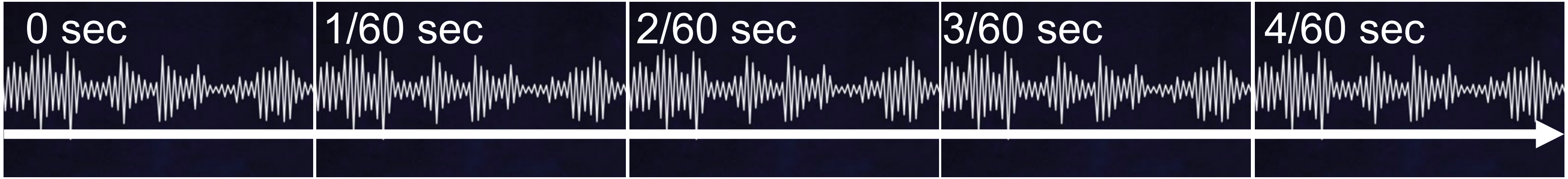
pixel line intensity
modulated on 32 kHz carrier

Observation (2): signal redundancy

- Screen refreshes every ~1/60 seconds

  ➔ the signal is extremely redundant!

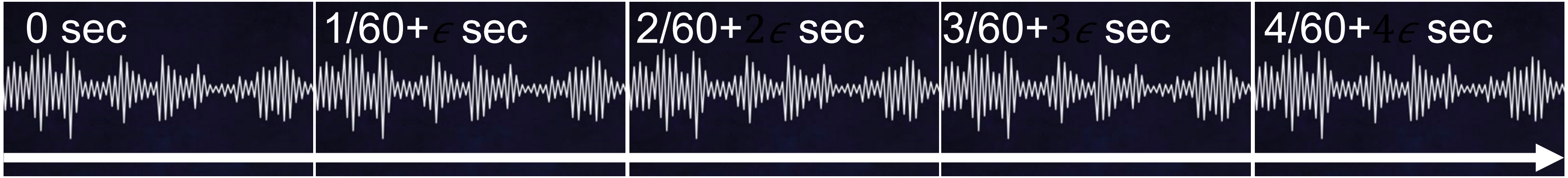
- Chop and average?



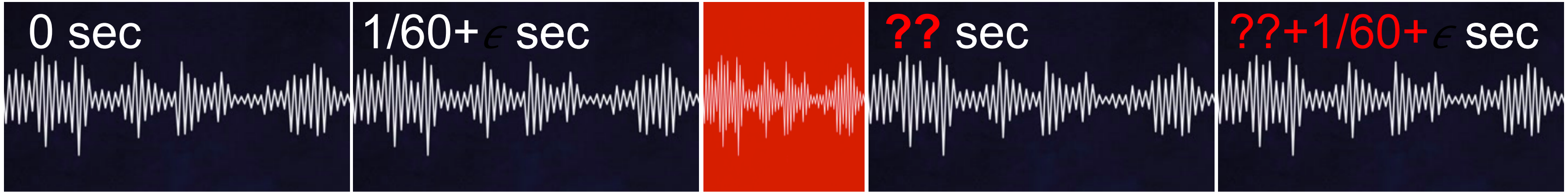
Average: high SNR!

Leveraging redundancy: challenges

- Drift



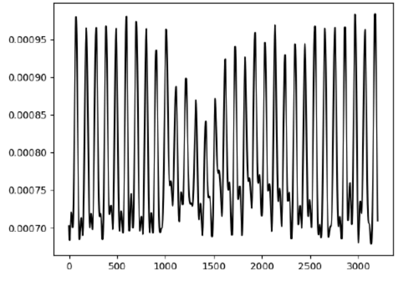
- Jitter (+anomalous refresh cycles)

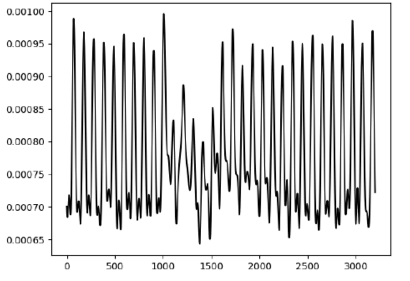


Leveraging redundancy: our approach

- Naïve approaches do not work
- High-level idea:
  - Choose a “master” chop that correlates well with its consecutive one
  - Extract chops chronologically, starting with the master
  - Automatically account for minor drift on-the-fly using a correlation test
  - If correlation becomes very low (indicating jitter encountered), re-synchronize with master chop via correlation analysis

Our approach



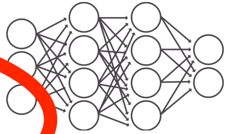


Ground truth

How?

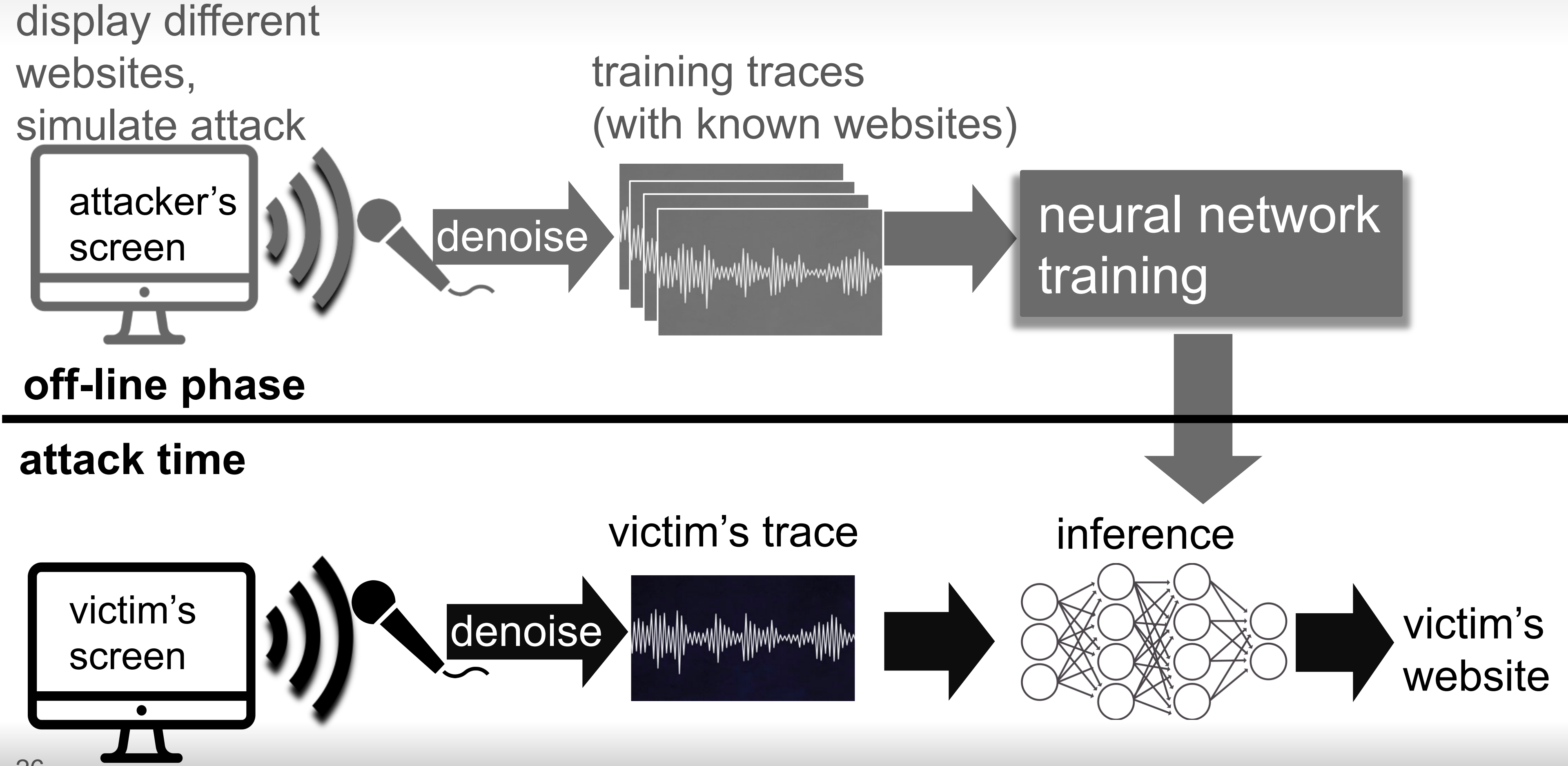
1. denoising

2. ML-based attacks



- Website distinguishing
- On-screen keyboard snoop
- Text extraction

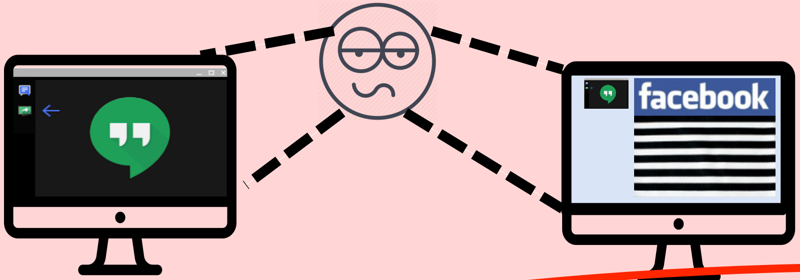
ML-based attacker: website distinguishing



Website distinguishing: results

attacker	accuracy	websites	traces per website
		97	100x5s
		97	100x5s
		97	100x5s
	**99.4%**	10 sites + Hangouts window	300x6s

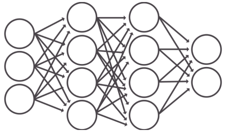
video-chat window vs. surfing the Web



How?

1. denoising

2. ML-based attacks



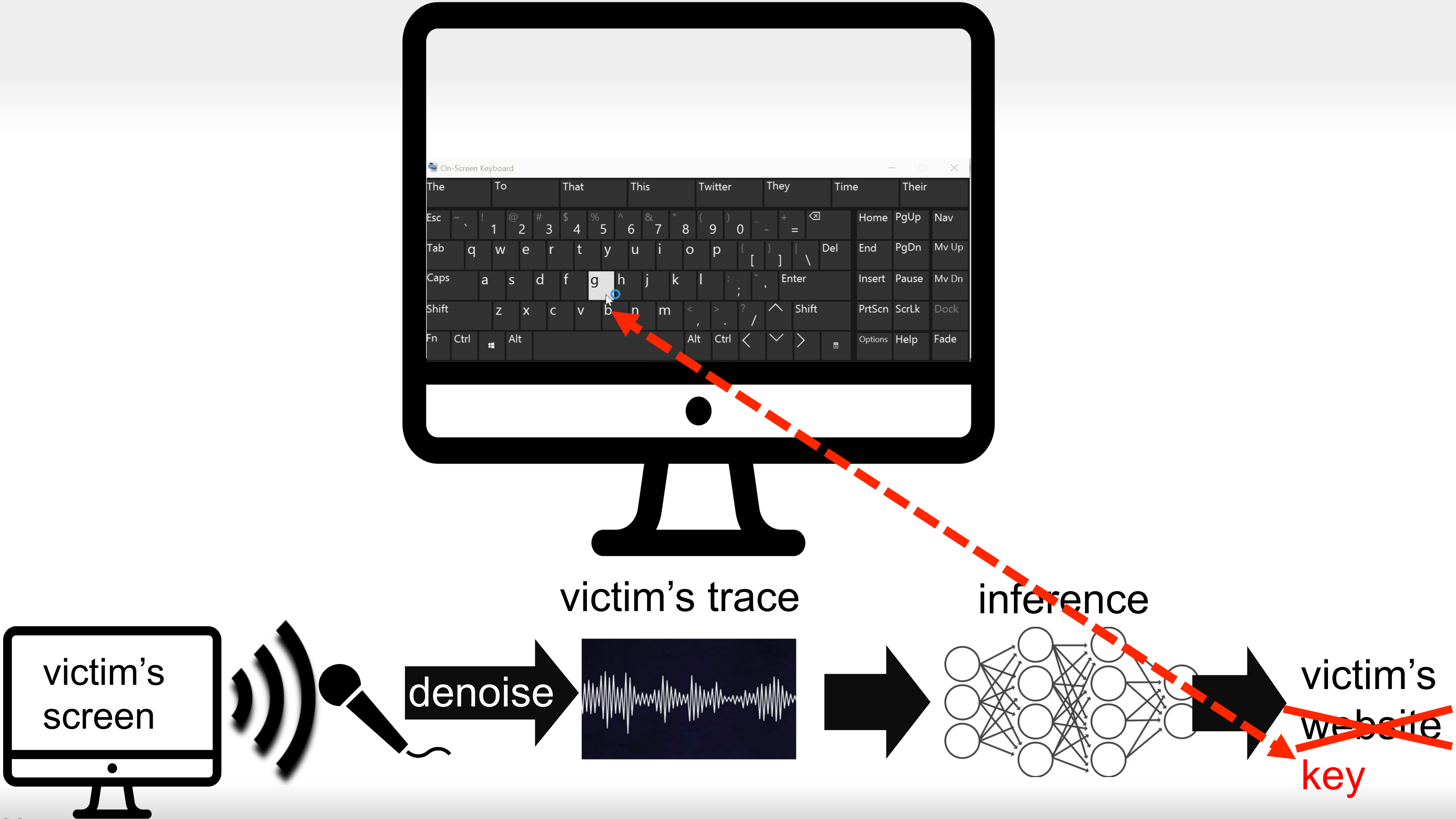
- Website distinguishing
- On-screen keyboard snoop
- Text extraction

On-screen keyboards

Considered “safe” against audio-recording attacks on physical keyboards

[AA’04, BWY’06, VP’09, HS’12, BCV’08, HS’15, ZZT09, CCLT’17]

Sometimes required for security, e.g., by online banking websites

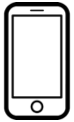

On-Screen Keyboard
The
To
That
This
Twitter
They
Time
Their
victim's trace
inference
victim's screen
denoise
victim's
~~website~~
key

Results: keyboard snooping 1

attacker	screen layout	key accuracy	key top-3 accuracy
			71.9%
		96.4%	99.6%

Extract whole words with high accuracy?

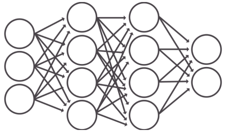
Results: keyboard snooping 2 (grouping horizontally-aligned keys)

attacker	screen layout	word contained in small “prediction set”
		94%
		98%

How?

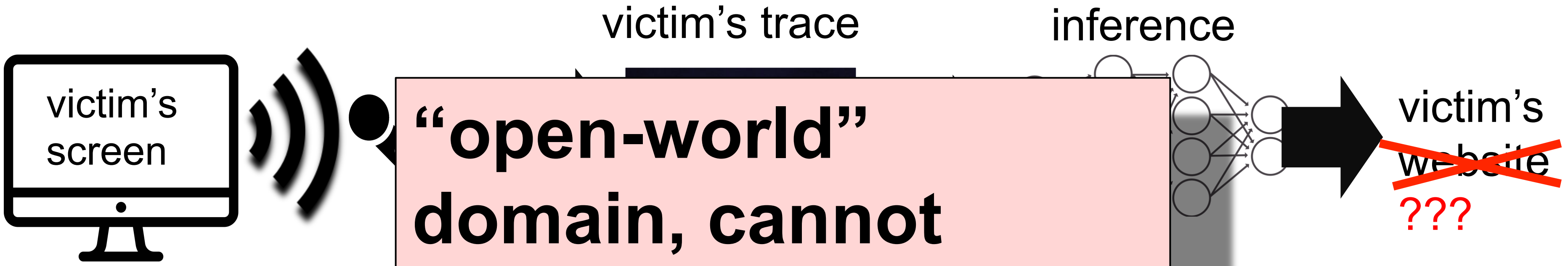
1. denoising

2. ML-based attacks



- Website distinguishing
- On-screen keyboard snoop
- Text extraction

ML-based attacker: text extraction



"open-world" domain, cannot directly apply classifier

Extracting on-screen text

- Idea:
    1. Train separate classifier for each character location

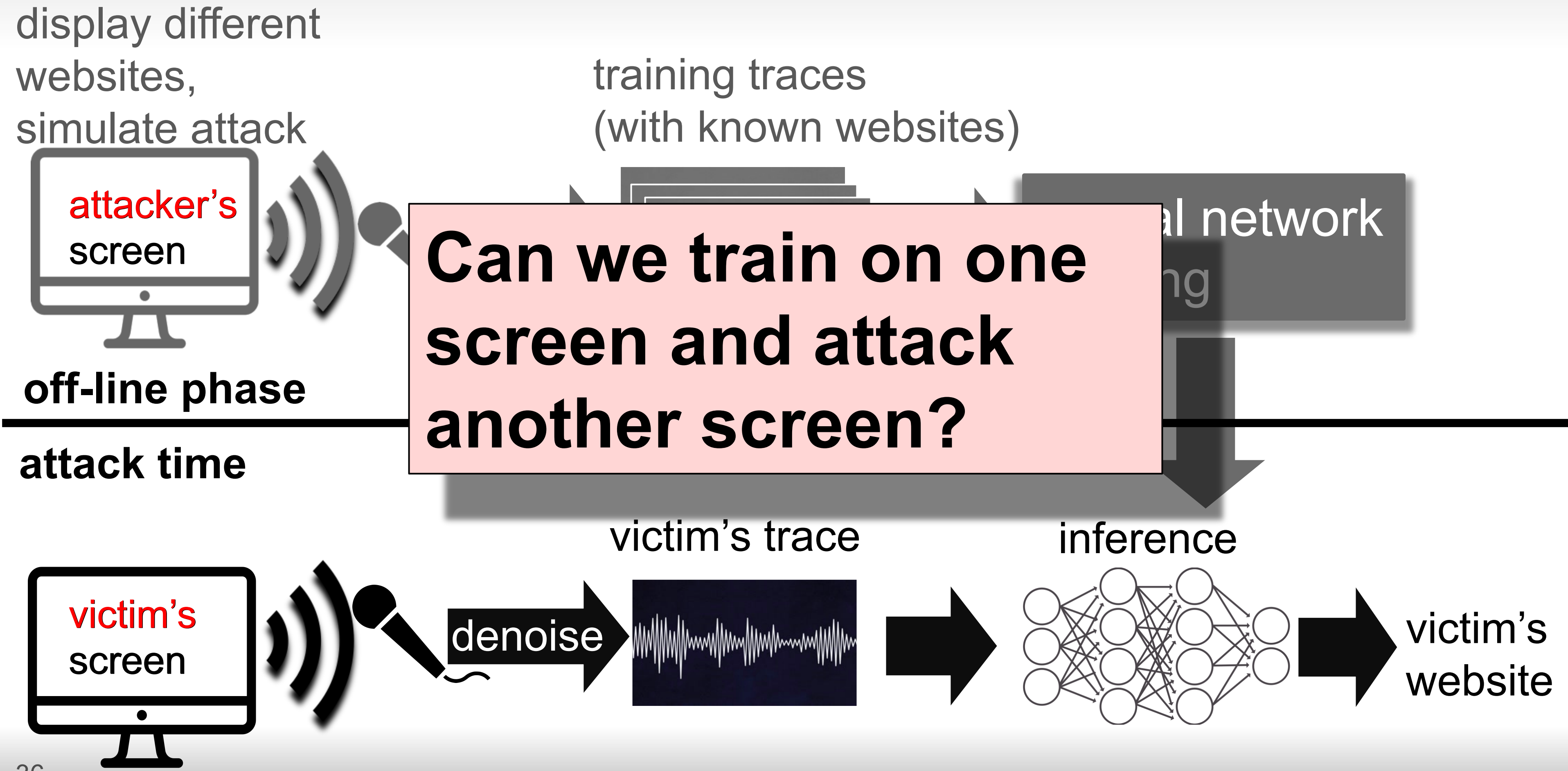
       ➔ Up to 98% per-character accuracy

    2. Error-correction exploiting natural language redundancy

       ➔Exact word extracted with probability >1/2

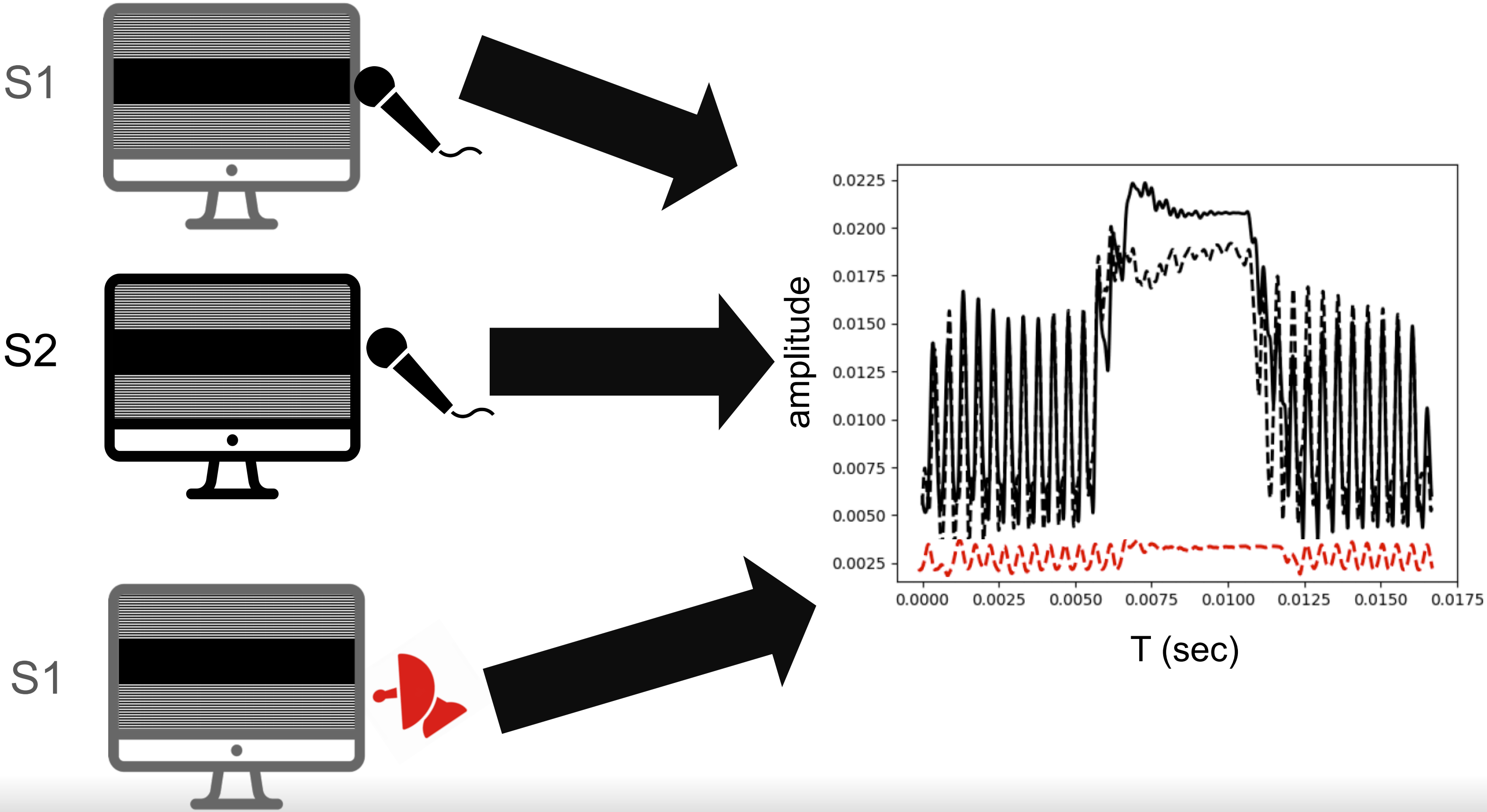
   Some limitations: large monospace font, known layout…

Cross-screen train-test



Can we train on one screen and attack another screen?

Are traces from different screens similar?



Learning from multiple screens

- Challenge: overfitting to training screen
- Idea: learn from **multiple** screens

Trend: more training screens → higher accuracy

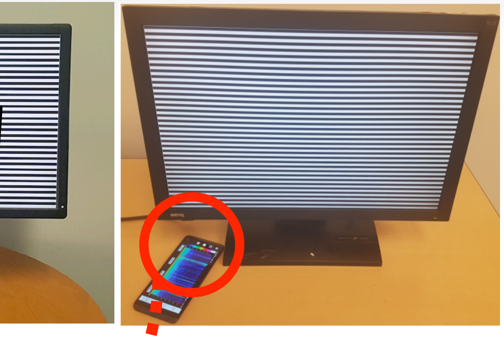
Up to 94% accuracy

Distinguishing between 25 websites, training on up to 10 screens

Victim screen

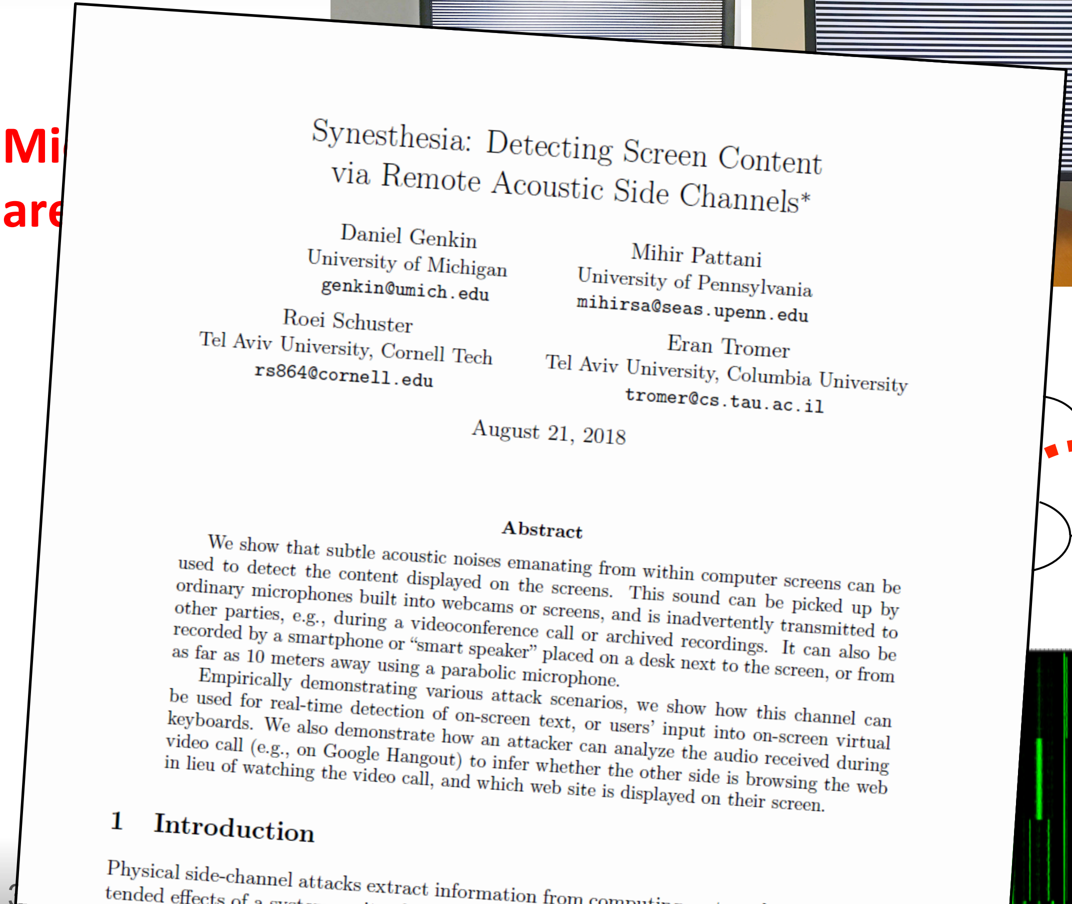
Train set	Dell4#0	Dell4#1	Dell4#2	Dell4#3	Dell4#4	Dell5#0	Dell5#1	DellB#0	DellB#1	Soyo#0	mean
Dell4#0	0.99	0.19	0.67	0.5	0.092	0.14	0.18	0.13	0.24	0.064	0.32
Dell4#1	0.47	1	0.54	0.48	0.06	0.12	0.41	0.7	0.12	0.048	0.4
Dell4#2	0.47	0.11	0.97	0.74	0.013	0.05	0.49	0.33	0.076	0.053	0.33
Dell4#3	0.45	0.19	0.77	1	0.096	0.048	0.61	0.33	0.035	0.033	0.36
Dell4#4	0.18	0.15	0.021	0.0093	1	0.8	0.01	0.11	0.052	0.097	0.24
Dell5#0	0.15	0.03	0.054	0.03	0.57	0.98	0.00093	0.082	0.034	0.092	0.2
Dell5#1	0.21	0.46	0.72	0.6	0.071	0.065	0.98	0.46	0.055	0.027	0.36
DellB#0	0.2	0.48	0.28	0.19	0.086	0.11	0.38	0.99	0.11	0.045	0.29
DellB#1	0.41	0.15	0.15	0.036	0.084	0.097	0.082	0.24	0.99	0.05	0.23
Soyo#0	0.096	0.071	0.013	0.08	0.16	0.14	0.021	0.038	0.019	1	0.16
Dell4	0.71	0.35	0.91	0.78	0.09	0.75	0.53	0.74	0.22	0.088	0.52
Dell5	0.41	0.35	0.68	0.53	0.55	0.0077	0.0019	0.56	0.11	0.087	0.33
DellB	0.38	0.4	0.48	0.31	0.077	0.24	0.33	0.23	0.033	0.037	0.25
all	0.71	0.72	0.9	0.8	0.48	0.73	0.62	0.8	0.27	0.098	0.61
mixed	0.44	0.43	0.83	0.77	0.52	0.24	0.45	0.62	0.17	0.078	0.46
nosoyo	0.84	0.68	0.94	0.81	0.52	0.7	0.64	0.81	0.22	0.12	0.63

cs.tau.ac.il/~tromer/synesthesia



Mi

are



Synesthesia: Detecting Screen Content via Remote Acoustic Side Channels*

Daniel Genkin
University of Michigan
genkin@umich.edu

Mihir Pattani
University of Pennsylvania
mihirsa@seas.upenn.edu

Roei Schuster
Tel Aviv University, Cornell Tech
rs864@cornell.edu

Eran Tromer
Tel Aviv University, Columbia University
tromer@cs.tau.ac.il

August 21, 2018

Abstract

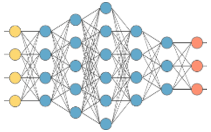
We show that subtle acoustic noises emanating from within computer screens can be used to detect the content displayed on the screens. This sound can be picked up by ordinary microphones built into webcams or screens, and is inadvertently transmitted to other parties, e.g., during a videoconference call or archived recordings. It can also be recorded by a smartphone or "smart speaker" placed on a desk next to the screen, or from as far as 10 meters away using a parabolic microphone.

Empirically demonstrating various attack scenarios, we show how this channel can be used for real-time detection of on-screen text, or users' input into on-screen virtual keyboards. We also demonstrate how an attacker can analyze the audio received during video call (e.g., on Google Hangout) to infer whether the other side is browsing the web in lieu of watching the video call, and which web site is displayed on their screen.

1 Introduction

Physical side-channel attacks extract information from

tended effects of a



A thousand words are worth a picture