SAFER PROGRAM BEHAVIOR SHARING THROUGH TRACE WRINGING

Deeksha Dangwal, Weilong Cui, Joseph McMahan, Timothy Sherwood

UC SANTA BARBARA

PERFORMANCE

APPLICATION TUNED SYSTEMS CAN IMPROVE PERFORMANCE

APPLICATION TUNED SYSTEMS CAN IMPROVE PERFORMANCE

GOOD UNDERSTANDING OF THE APPLICATION

APPLICATION TUNED SYSTEMS CAN IMPROVE PERFORMANCE

GOOD UNDERSTANDING OF THE APPLICATION

(IN THE WILD)

APPLICATION TUNED SYSTEMS CAN IMPROVE PERFORMANCE

GOOD UNDERSTANDING OF THE APPLICATION

REAL PROGRAM TRACES

PRODUCTION CODE

PRODUCTION ENVIRONMENTS

APPLICATION TUNED SYSTEMS CAN IMPROVE PERFORMANCE

GOOD UNDERSTANDING OF THE APPLICATION

WHY ARE SO FEW PROGRAM TRACES AVAILABLE FOR ANALYSIS?

REAL PROGRAM TRACES

PRODUCTION CODE

PRODUCTION ENVIRONMENTS

PROGRAM TRACES LEAK

ARBITRARY INFORMATION *ABOUT SYSTEMS*

PROGRAM TRACES LEAK
ARBITRARY INFORMATION *ABOUT SYSTEMS*

PROGRAM TRACES LEAK
ARBITRARY INFORMATION *ABOUT SYSTEMS*

BEHAVIOR OF A
CRITICAL **CRYPTO FUNCTION**

LOOKUPS CORRESPONDING
TO **SOCIAL SECURITY NUMBERS**

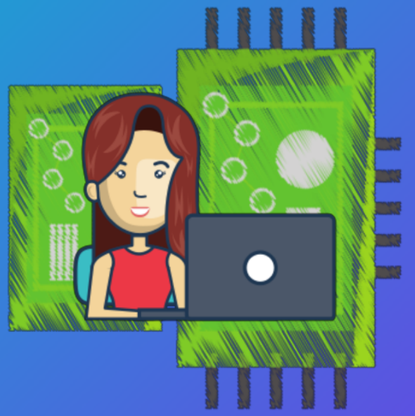
PROGRAM TRACES LEAK
ARBITRARY INFORMATION *ABOUT SYSTEMS*

BEHAVIOR OF A CRITICAL **CRYPTO FUNCTION**

LOOKUPS CORRESPONDING TO **SOCIAL SECURITY NUMBERS**

SYSTEM CONFIGURATION, *A TRADE SECRET*

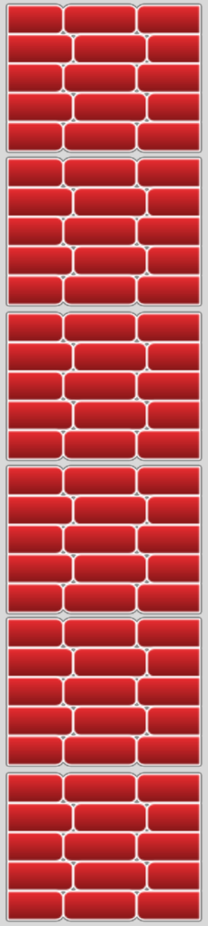
SOFTWARE
DEVELOPER

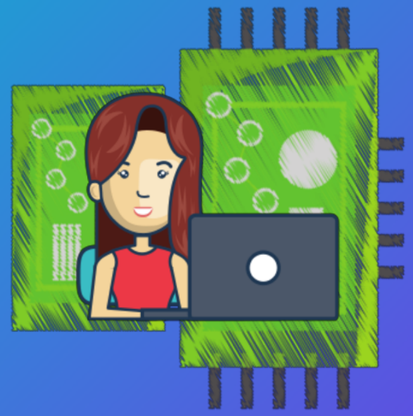


HARDWARE
DESIGNER

MUTUAL DISTRUST BETWEEN HARDWARE AND SOFTWARE COLLABORATORS

SOFTWARE DEVELOPER



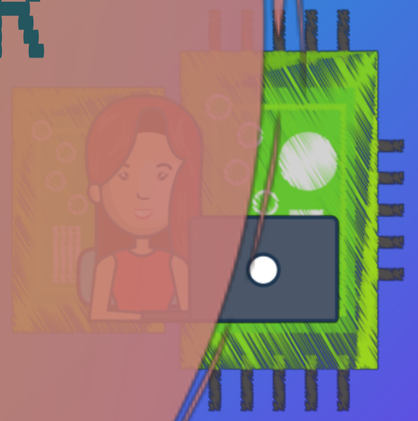


HARDWARE DESIGNER

MUTUAL DISTRUST BETWEEN HARDWARE AND SOFTWARE COLLABORATORS

HOW DO WE SHARE PROGRAM BEHAVIOR TODAY?

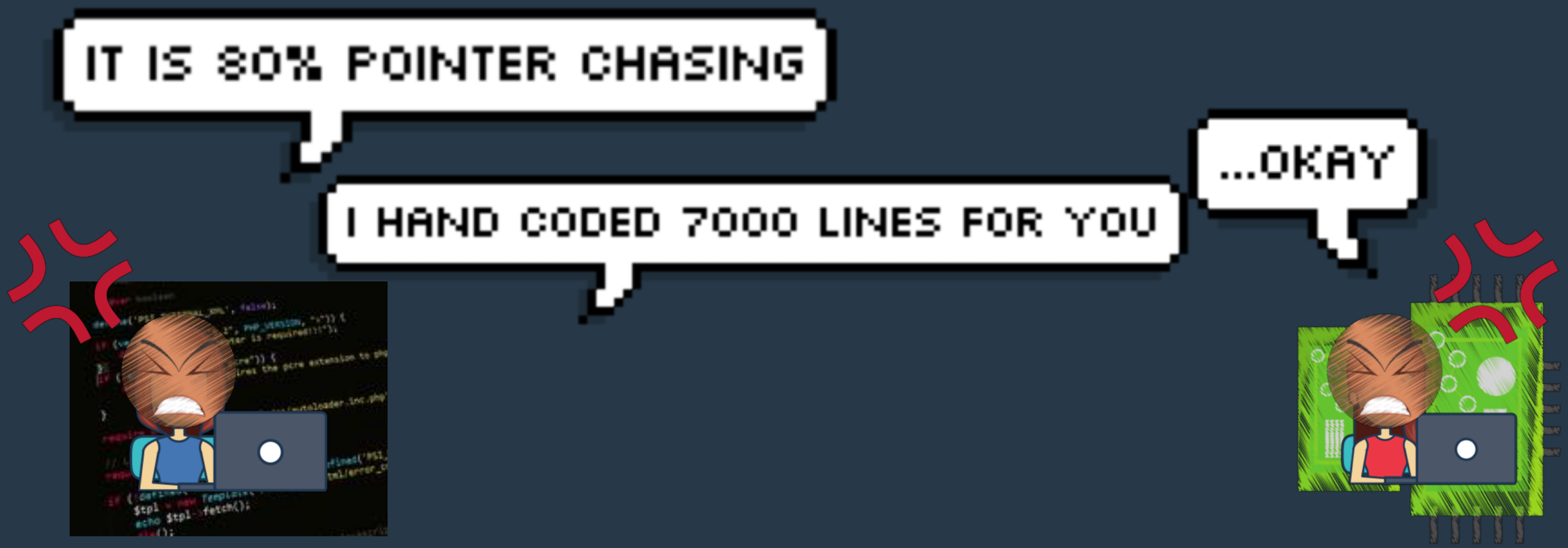
SOFTWAR
DEVELOPER



HARDWARE
DESIGNER

IT IS 80% POINTER CHASING

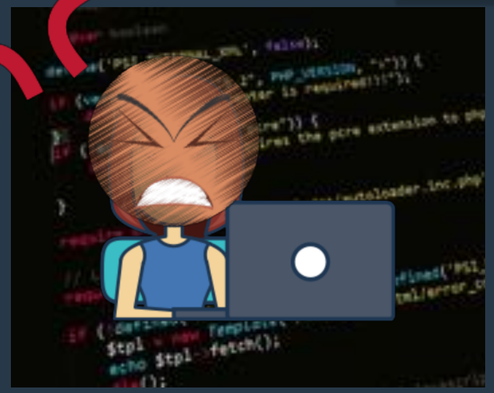
...OKAY


IT IS 80% POINTER CHASING
I HAND CODED 7000 LINES FOR YOU
...OKAY

IT IS 80% POINTER CHASING

I HAND CODED 7000 LINES FOR YOU

...OKAY

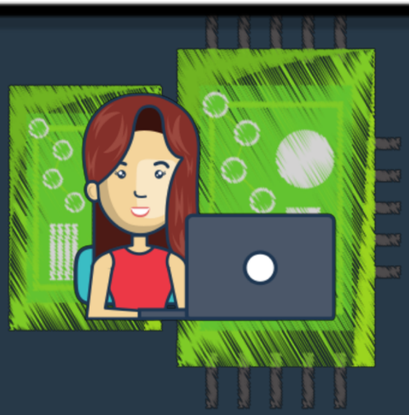


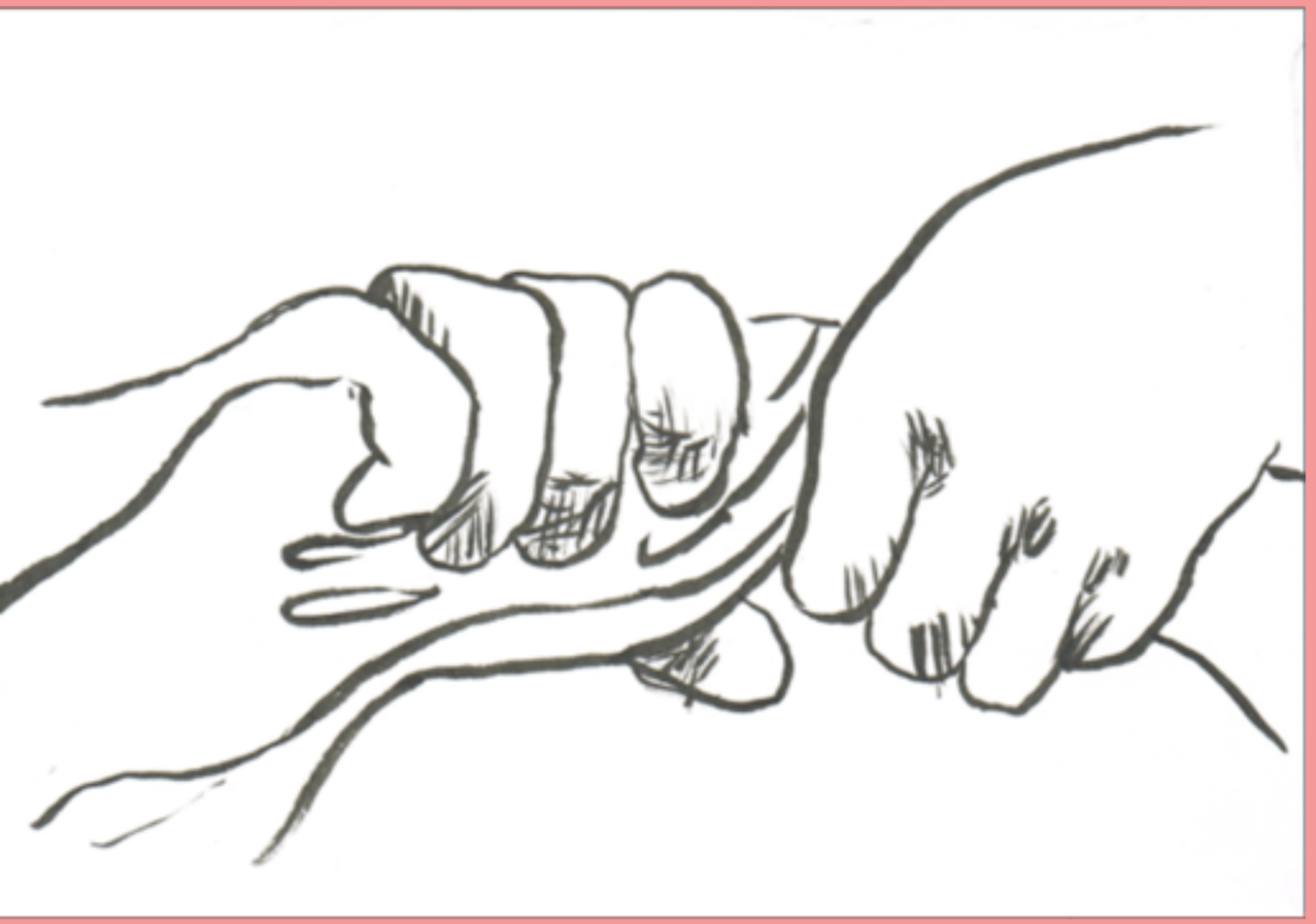
PROGRAMMERS ARE ASKED TO OBFUSCATE KEY BEHAVIORS


IT IS 80% POINTER CHASING
I HAND CODED 7000 LINES FOR YOU
...OKAY
TECHNOLOGY PARTNERS NEED BETTER WAYS TO COMMUNICATE ABOUT PROGRAM BEHAVIOR!
PROGRAMMERS ARE ASKED TO OBFUSCATE KEY BEHAVIORS

WISH WE HAD TOOLS TO ELIMINATE SENSITIVE INFORMATION FROM PROGRAM TRACES

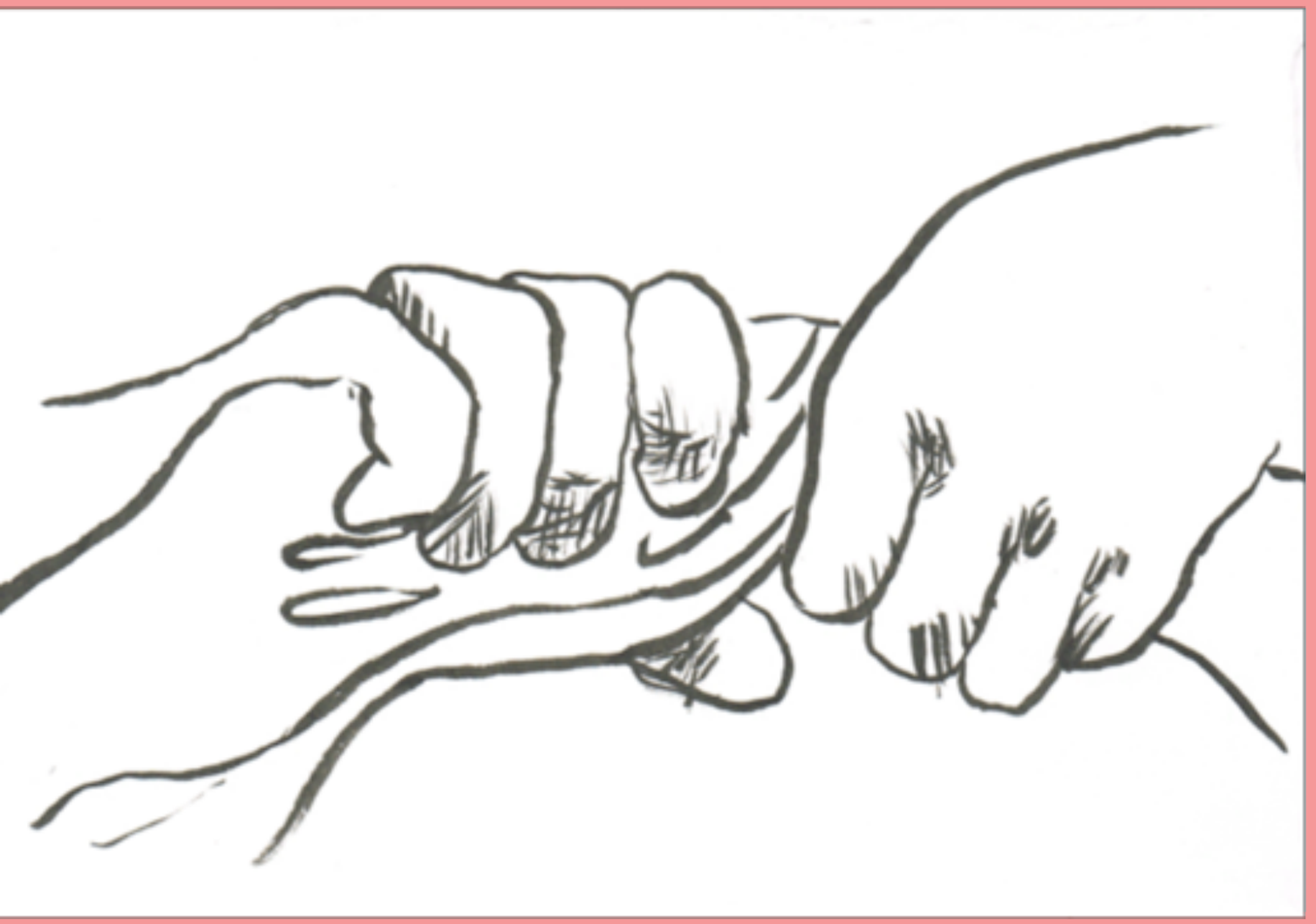
WHILE STILL CAPTURING THE PROGRAM'S BEHAVIOR!





INTRODUCING

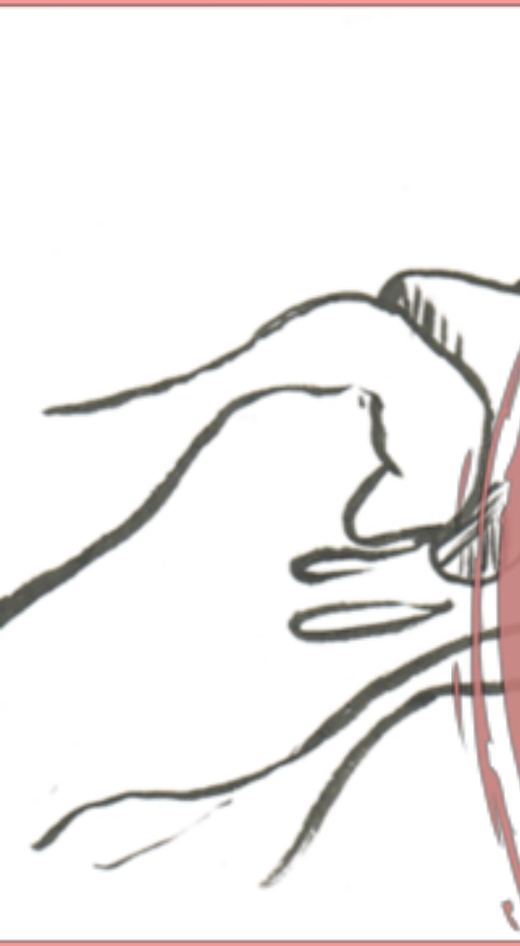
TRACE WRINGING



INTRODUCING

TRACE WRINGING

bounds how much information a trace is leaking

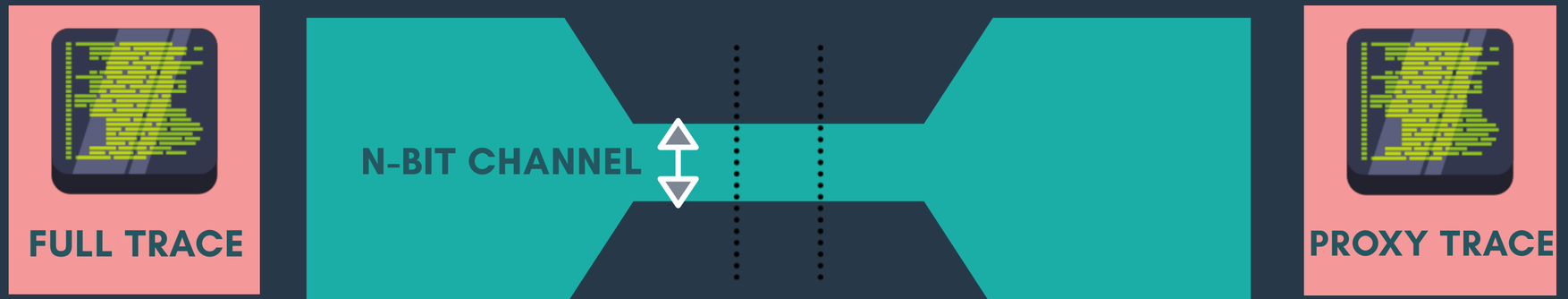


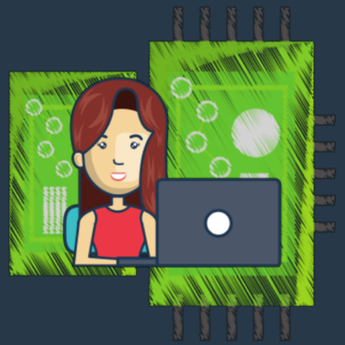
HOW CAN WE USE TRACE WRINGING TO REDUCE THE INFORMATION IN A TRACE DOWN TO N BITS?

INTRODUCING TRACE WRINGING

bounds how much information a trace is leaking

CANNOT LEAK MORE THAN N BITS



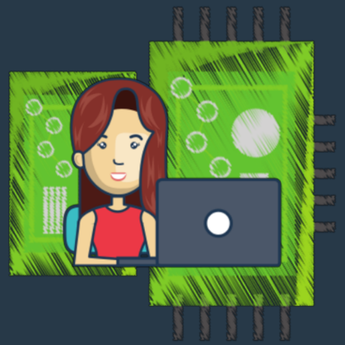


PUBLIC INFORMATION

FULL TRACE

N-BIT CHANNEL

PROXY TRACE



PUBLIC INFORMATION

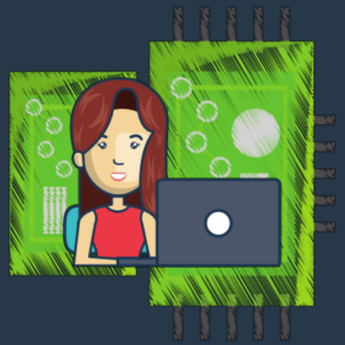
FULL TRACE

N-BIT CHANNEL

PROXY TRACE

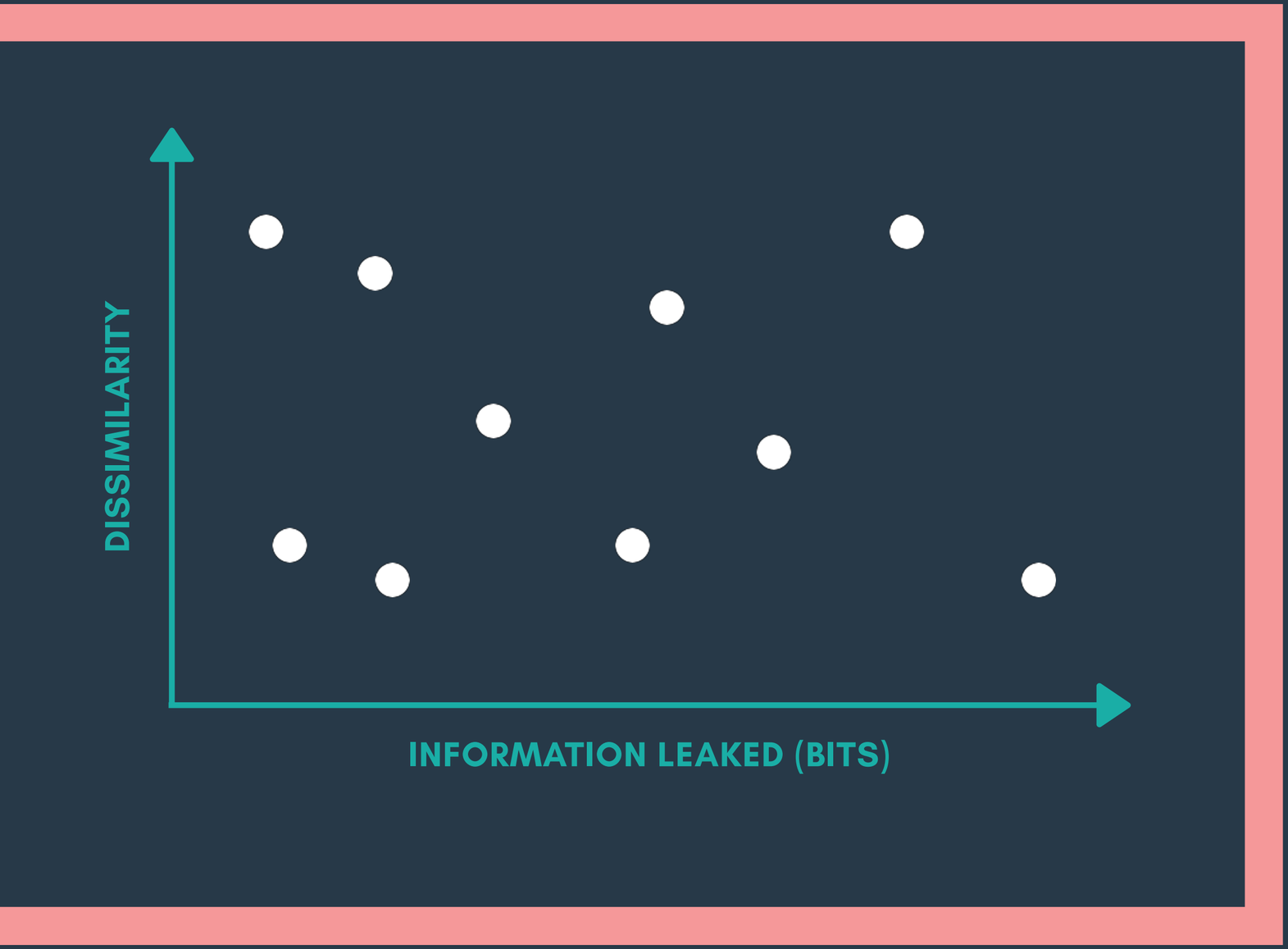
UTILITY

(CACHE MISS RATES)



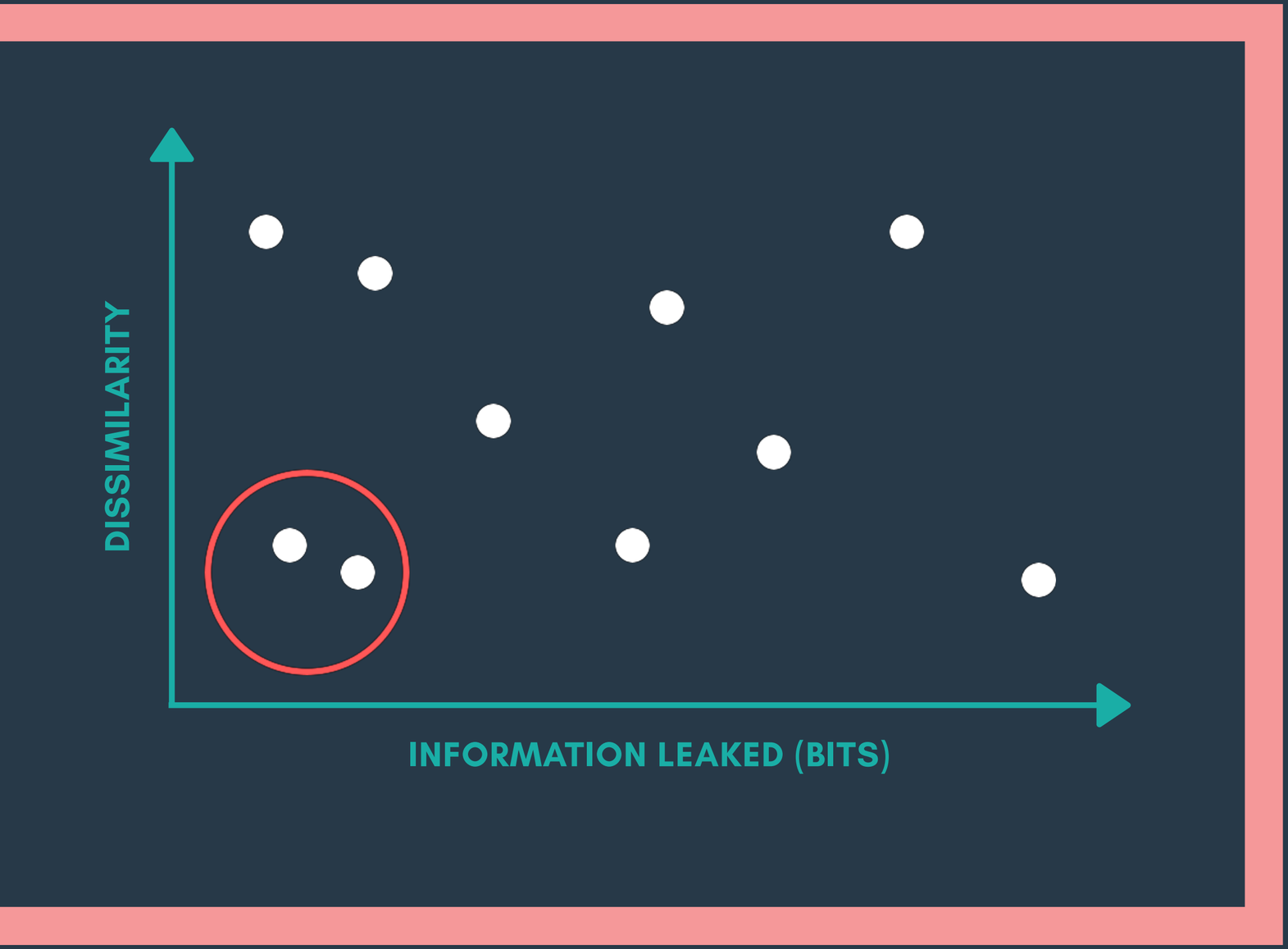
TRACE WRINGING PROVIDES A USEFUL UPPER BOUND

IF WE ONLY SHARE **N BITS** *ABOUT A SPECIFIC TRACE THEN WE* **CANNOT LEAK MORE THAN N BITS** *ABOUT THAT TRACE*



TRACE WRINGING PROVIDES A USEFUL UPPER BOUND

IF WE ONLY SHARE **N BITS** *ABOUT A SPECIFIC TRACE THEN WE* **CANNOT LEAK MORE THAN N BITS** *ABOUT THAT TRACE*



TRACE WRINGING PROVIDES A USEFUL UPPER BOUND

IF WE ONLY SHARE **N BITS** *ABOUT A SPECIFIC TRACE THEN WE* **CANNOT LEAK MORE THAN N BITS** *ABOUT THAT TRACE*

MEMORY ACCESS TRACES

WELL STUDIED, UNDERSTOOD

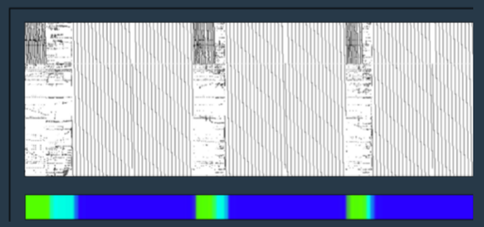
RELATED WORK

- TRACE COMPRESSION AND APPROXIMATION
- PROGRAM BEHAVIOR CHARACTERIZATION
- SYNTHETIC TRACE AND BENCHMARK GENERATION
- PRIVACY PRESERVATION

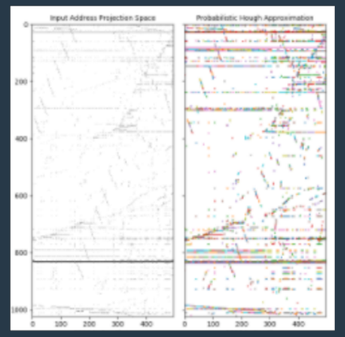
A signal processing pipeline for trace wringing

GENERATE HEATMAPS

PHASE DETECTION



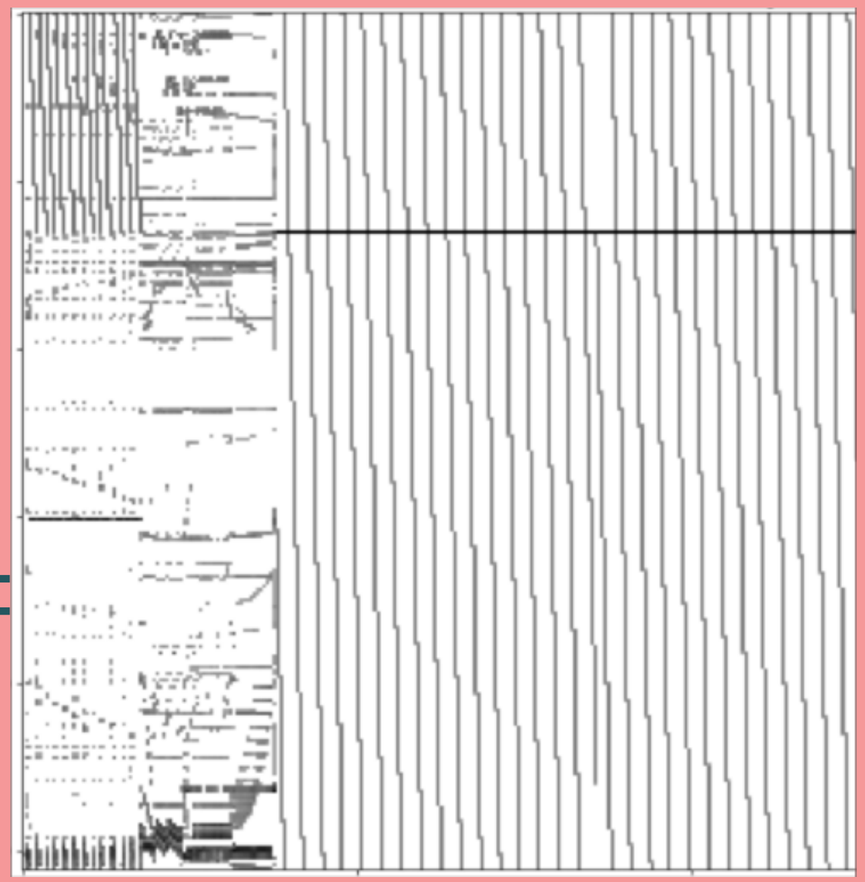
LINE DETECTION



PACKET GENERATION

RLE +
FIXED POINT +
GZIP/BZIP2

GENERATING HEATMAPS



X AXIS

Instruction windows of ten thousand instructions

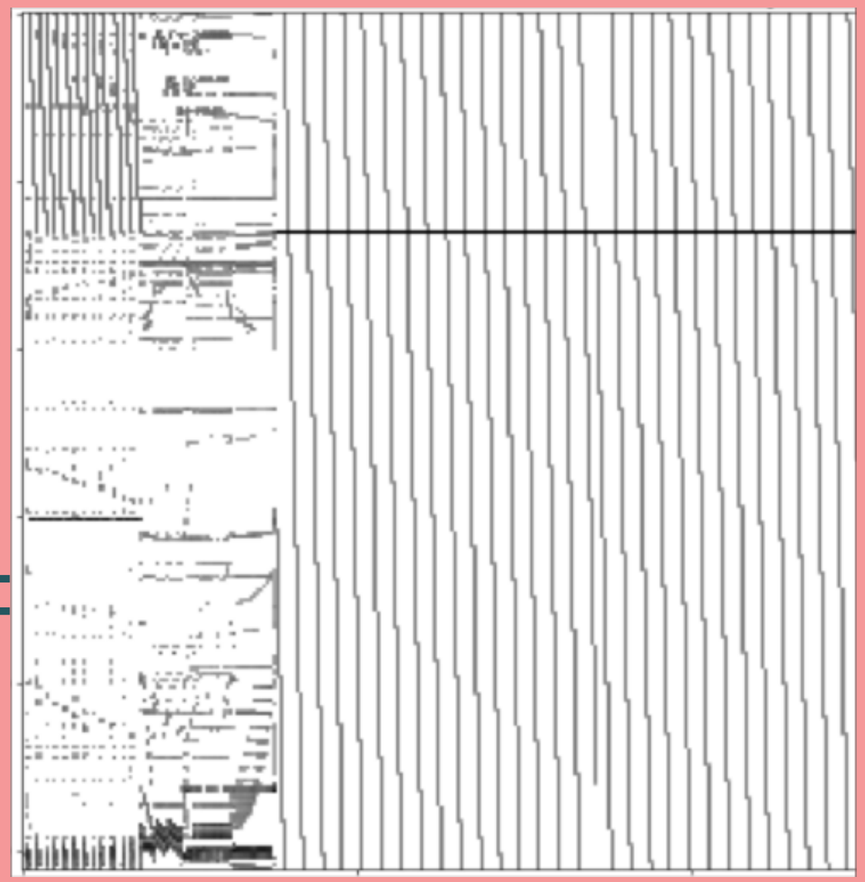
Y AXIS

Modulo wrapped address

DARKNESS OF PIXELS

Number of accesses to the modulo address in the given window

GENERATING HEATMAPS



VISUALIZING MEMORY ACCESS PATTERNS CAN HELP US CONCISELY DESCRIBE THEM

X AXIS

Instruction windows of ten thousand instructions

Y AXIS

Modulo wrapped address

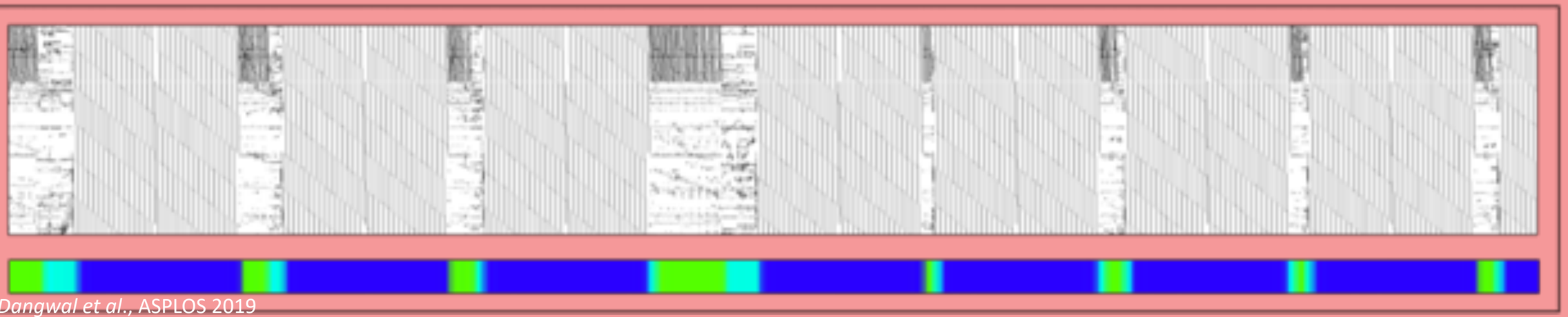
DARKNESS OF PIXELS

Number of accesses to the modulo address in the given window

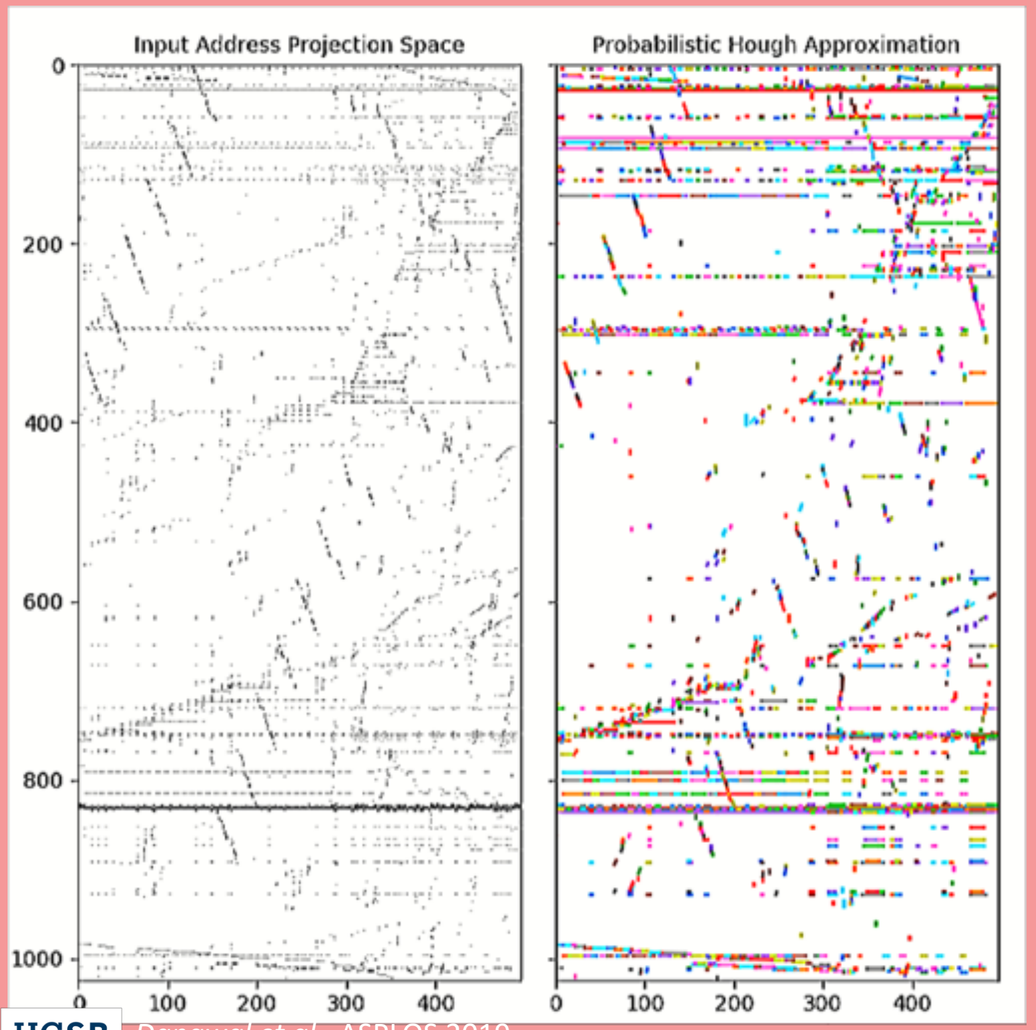
PHASE DETECTION HELPS US CAPTURE RECURRING PATTERNS IN TRACES

K-MEANS CLUSTERING

ENCODED SEQUENTIAL DATA USING RUN LENGTH ENCODING



LINE DETECTION



MEMORY BEHAVIORS MANIFEST AS STRIDES IN THE HEATMAP

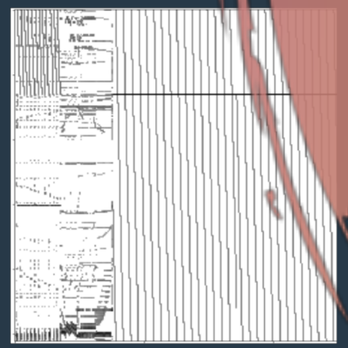
HOUGH LINE TRANSFORM

Computer vision algorithm which finds lines (even in noisy images) in a polar search space

A signal processing pipeline for trace wringing

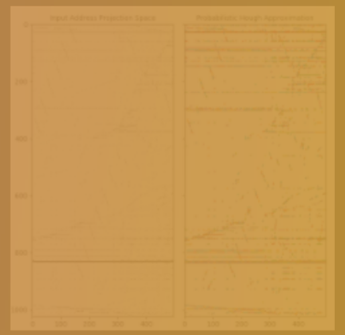
DOES THIS PIPELINE WORK?

GENERATE HEATMAPS



PHASE DETECTION

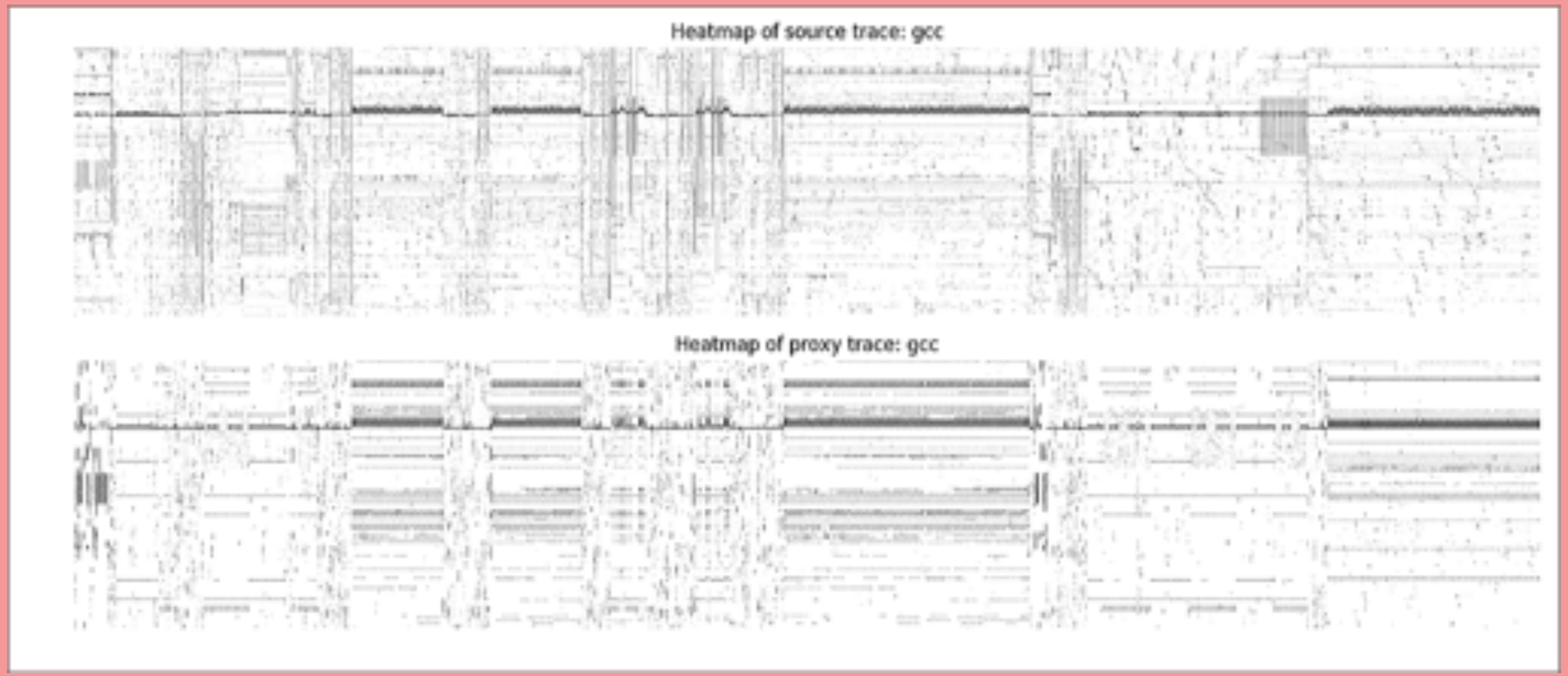
LINE DETECTION

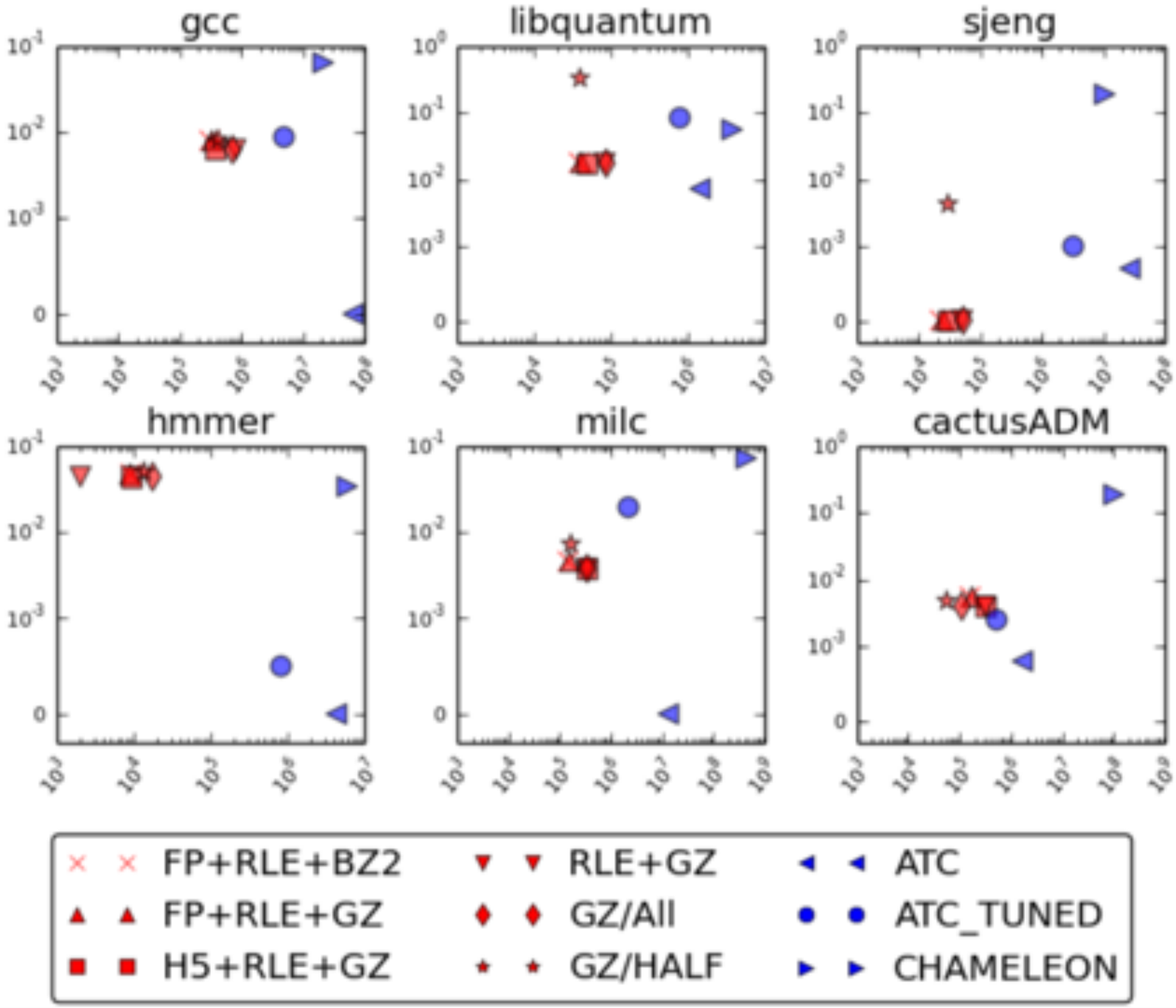


PACKET GENERATION

RLE +
FIXED POINT +
GZIP/BZIP2

VISUAL COMPARISON OF TRACES





BIT-ERROR POINTS

X AXIS

BITS LEAKED

Y AXIS

GEOMEAN OF ERROR

Knobs in the signal processing pipeline

GENERATE HEATMAPS

- **window size**

PHASE DETECTION

- **number of phases**

LINE DETECTION

- **line length**
- **line gap**
- **threshold**
- **theta**

PACKET GENERATION

- **encoding scheme**

PUBLIC INFORMATION

FULL TRACE

+

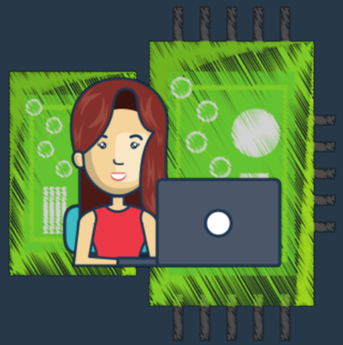
HYPERPARAMETERS

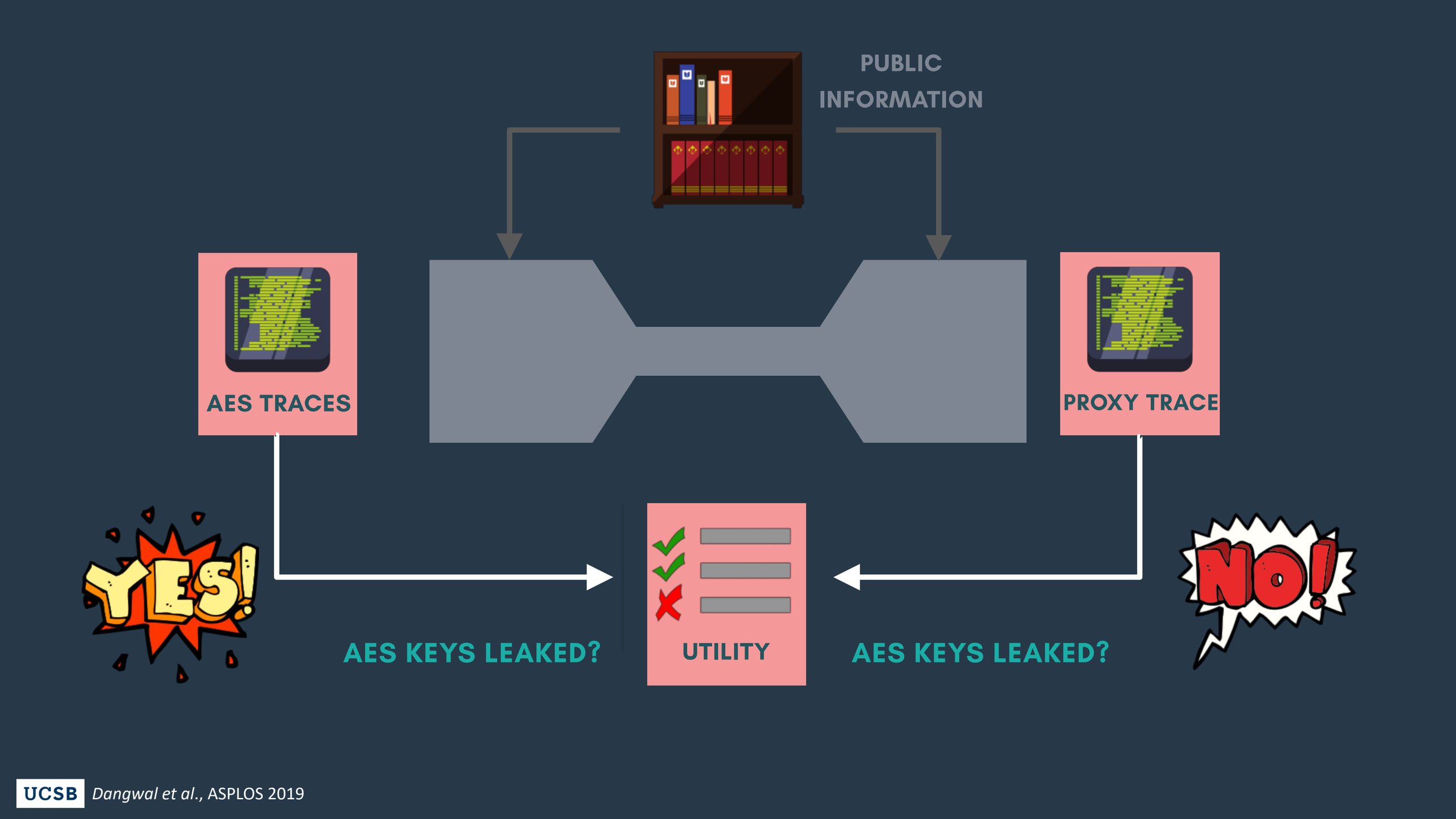
N-BIT CHANNEL

PROXY TRACE

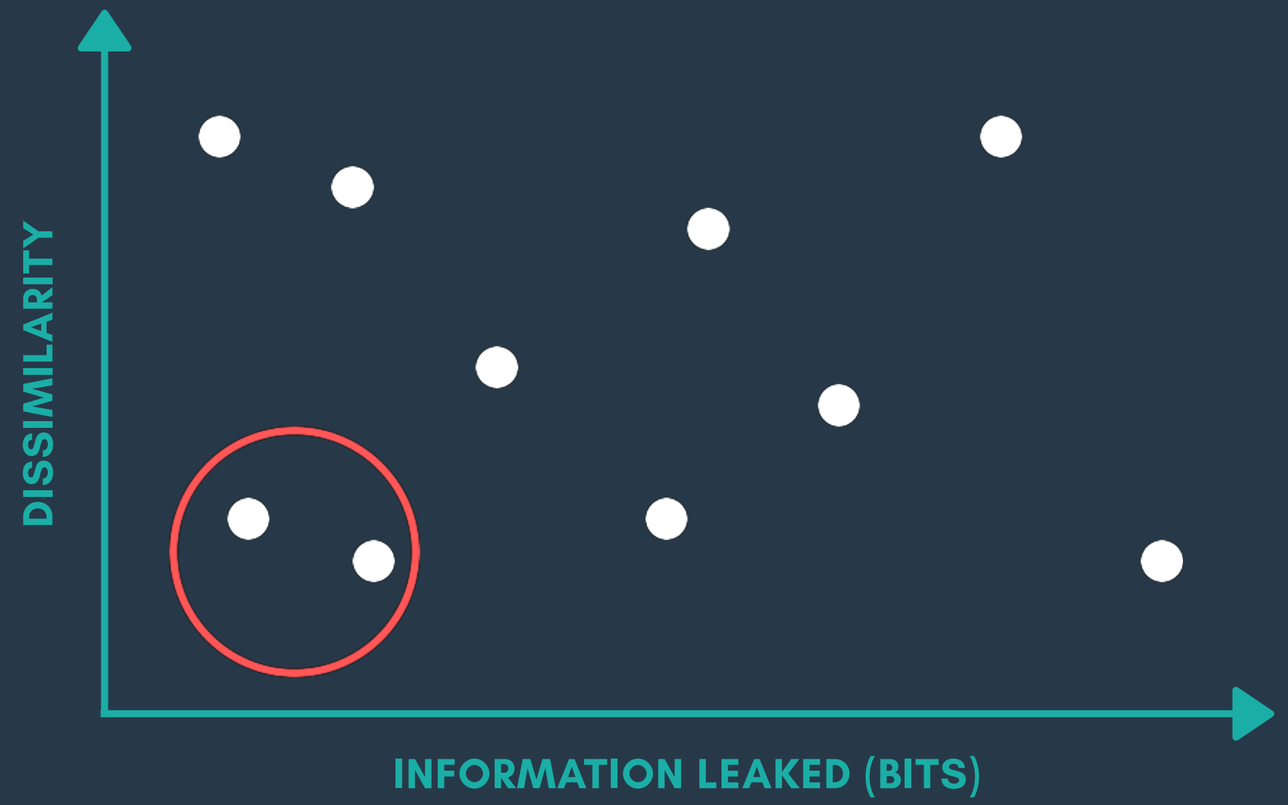
UTILITY

(CACHE MISS RATES)

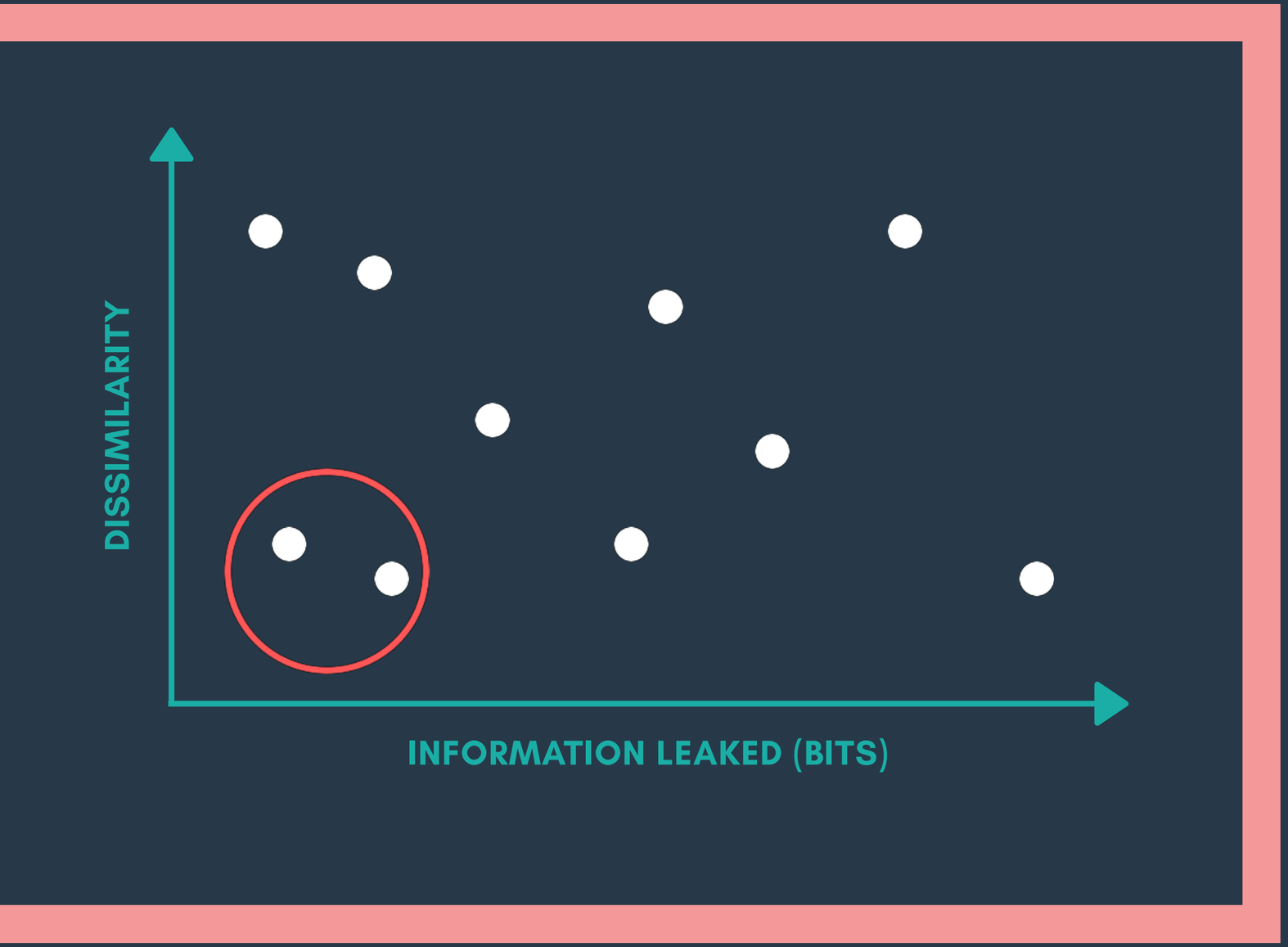



PUBLIC
INFORMATION
AES TRACES
PROXY TRACE
YES!
AES KEYS LEAKED?
UTILITY
AES KEYS LEAKED?
NO!

- SAFE SHARING TO A MEASURABLE SYSTEMS PROBLEM
- STRONG AND CLEAR BOUND ON INFORMATION LEAKED



- SAFE SHARING TO A MEASURABLE SYSTEMS PROBLEM
- STRONG AND CLEAR BOUND ON INFORMATION LEAKED



- SAFE SHARING TO A MEASURABLE SYSTEMS PROBLEM
- STRONG AND CLEAR BOUND ON INFORMATION LEAKED

THANK YOU!