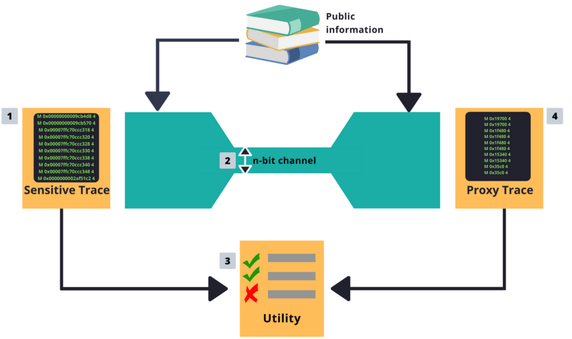
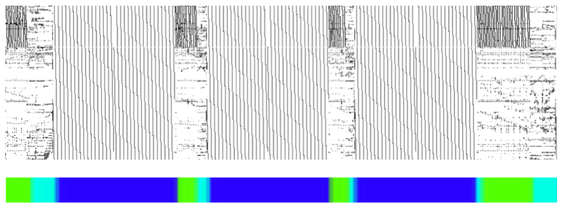
Releasing program traces is effective for application tuning. But, a trace taken from a production server might leak information about the users, the system, or even a computation (e.g. through a side channel). To maintain the privacy of program traces, my research proposes two techniques: **trace wringing**, and **trace scrubbing**. Trace wringing preserves the structure among addresses in the trace without leaking the actual addresses. It uses a simple metric to quantify information leakage—*number of bits*. The pipeline leverages computer vision techniques to **describe repetitive program patterns succinctly into lossily-compressed "packets"**. The size of the packet in bits provides an upper bound on information leakage. **Our argument is: if we only share *n bits* about the trace, then we cannot leak more than *n bits* about that trace.** When sensitive information can be identified at the program level, the impact of that sensitive data can be identified in the resulting address and instruction traces. We present an ensemble of **trace scrubbing** techniques that, at the extreme end simply deletes or **redacts sensitive addresses** from a trace or **replaces sensitive data** with stand-in addresses that are behaviorally similar.

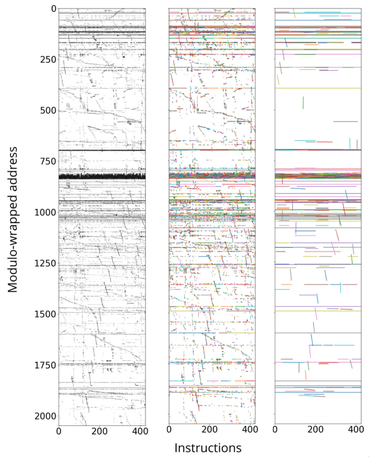
## TRACE WRINGING PRIVACY MODEL




## PHASE DETECTION

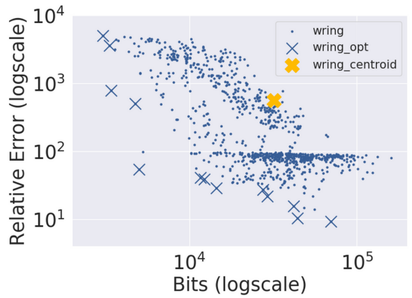


## LINE DETECTION

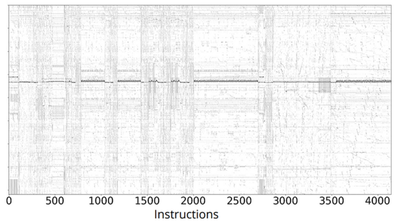



## PACKET GENERATION

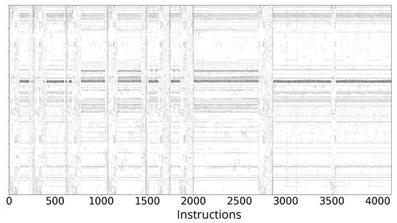
## OPTIMIZING BIT-ERROR POINTS




## PROXY TRACE GENERATION

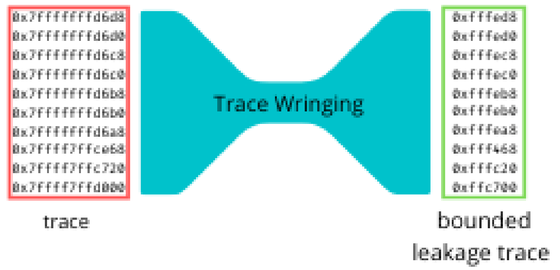



(a) Heatmap of input trace (gcc)

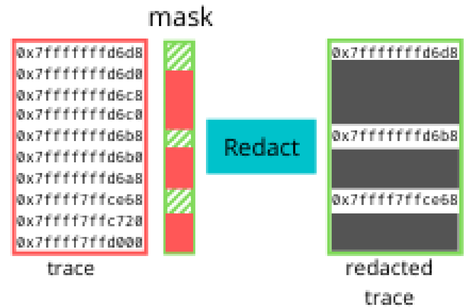


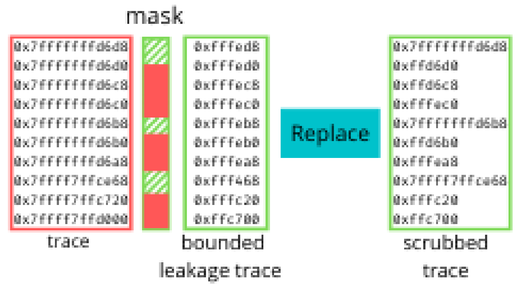

(b) Heatmap of proxy trace (gcc)

## LEVERAGING INFORMATION FLOW TECHNIQUES FOR TRACE SCRUBBING








Both wringing and scrubbing present a *knob* to system designers working with sensitive data and turning this knob can be used to traverse the tradeoff space between privacy and utility. Wringing and scrubbing techniques go beyond traces: AR applications rely on mapping and localization based on user images. To minimize leakage of private user data, we can compress visual information to balance privacy risk and localization accuracy.