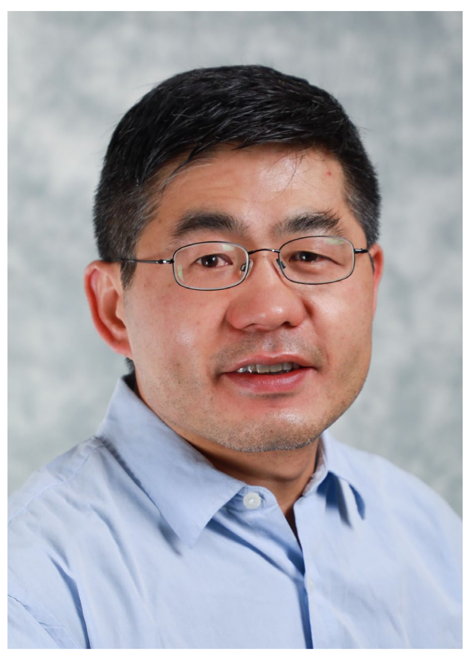
Dynamic and Static Program Analysis for Side Channel Vulnerability Detection and Mitigation



Dinghao Wu, Pennsylvania State University

https://faculty.ist.psu.edu/wu/projects/toa/toa.html

Side Channel Attacks

An attacker does not have access to the normal computation data or process, but has access to some "side" information.

- Power Dissipation
- Timing Information
- Electromagnetic Fields (EM)
- Cache Status
- Speculative execution (Spectre)
- Race conditions (Meltdown)
- etc.

Cache-based Side Channel Attacks

x = A[k] (k is secret dependent)

Whether A[k] is cached or not reveal some information about the secret key.

An attacker has a way to figure out the cache status and thus infer the key.

- Evict + Time
- Prime + Probe
- Flush + Reload

Challenges

- How to model cache behaviors?
- Modern computer memory systems are too complicated to model in a precise way.
- LRU, for example, is too complicated for program analysis.
- How to make the analysis scalable?
- How to reduce false positives and false negatives?
- How to quantify or rank the severity of the discovered vulnerabilities?

Solution

$$F(p,k) >> L \neq F(p,k') >> L$$

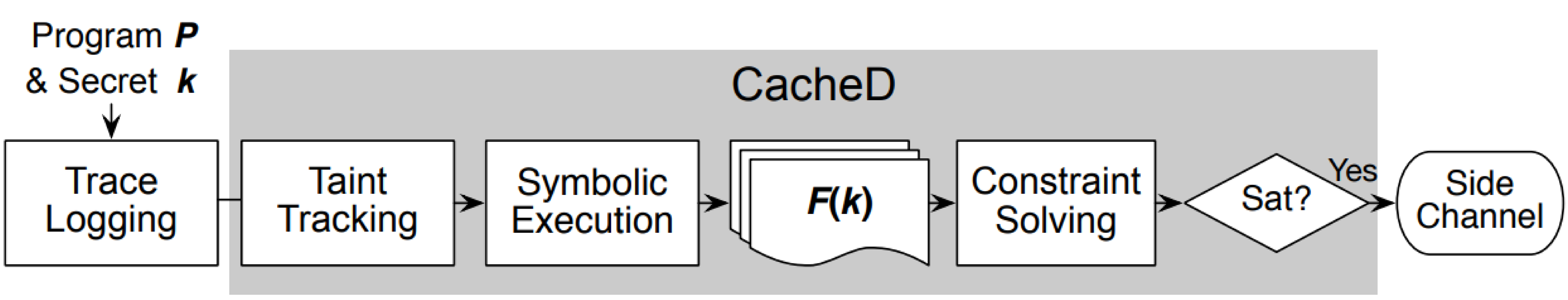
p – public information

k – secret information

F(p,k) a symbolic memory address accessed

F(p,k') replace the secret symbol k with a fresh variable k'

L cache line width



Broader Impact (impact on society – who will care)

- Found many new vulnerabilities in the production crypto systems

Broader Impact (education and outreach)

- Built course modules from the binary code analysis research prototypes

Broader Impact (quantify potential impact)

- Some of the new vulnerabilities discovered have been fixed

The 4th NSF Secure and Trustworthy Cyberspace Principal Investigator Meeting (2019 SaTC PI Meeting)
October 28-29, 2019 | Alexandria, Virginia

Project: CAREER: Advanced Trace-oriented Binary Code Analysis
Award ID#: NSF CNS-1652790