Ascend: An Architecture for Performing Secure Computation on Encrypted Data

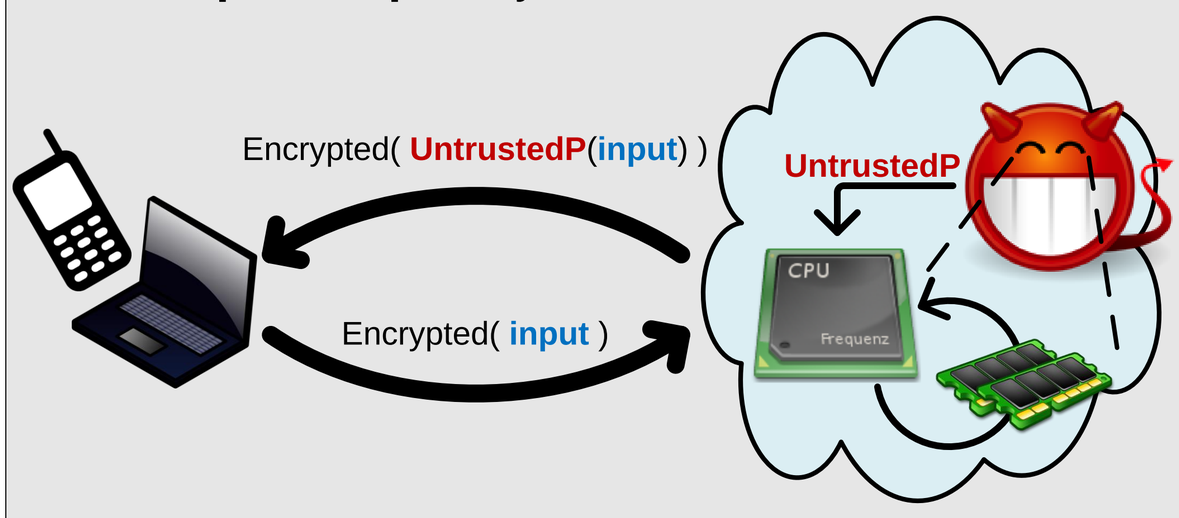
Christopher W. Fletcher[1], Ling Ren[1], Xiangyao Yu[1]
Marten van Dijk[2], Albert Kwon[1], Omer Khan[2], Srinivas Devadas[1]

[1]MIT; Cambridge, MA, USA
[2]University of Connecticut; Storrs, CT, USA

Presented during the ACSC Annual Conference, November 2013

1 Context & Motivation

In cloud computing, users submit **private data** to be **computed upon** by **unvetted cloud software**



Problem: Software causes data-dependent behaviors which are visible outside the chip (e.g., I/O requests, power consumption, execution time)

2 Big Ideas

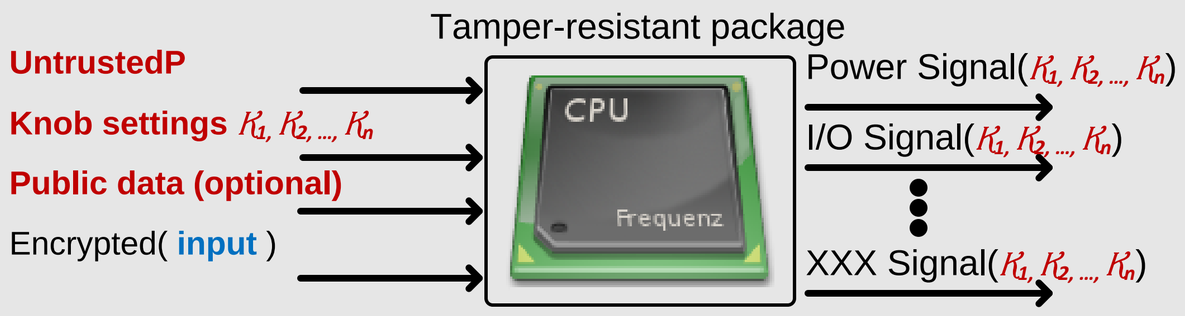
1.) Obfuscated Program Execution

Design processors to generate dummy work, to hide **what** programs are actually doing (i.e, make every program "look" the same)

2.) Public Knob-based Processor Design

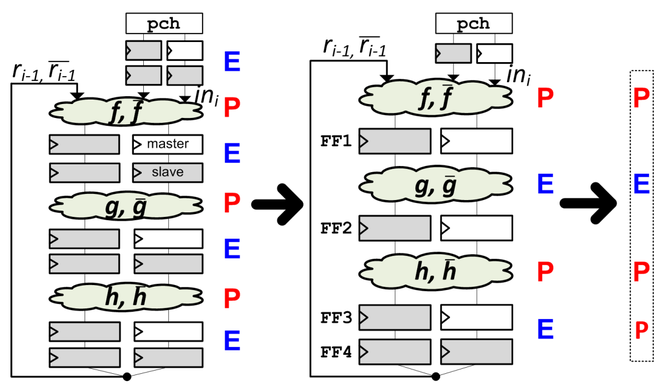
Problem: making all programs "look the same" incurs large overheads

Solution: design processors to emit signals that are purely a function of a set of user/attacker tunable knobs

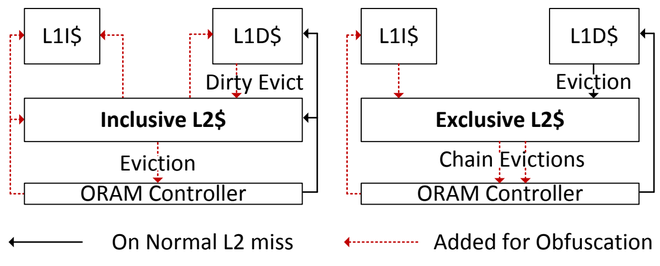


3 Power Channel

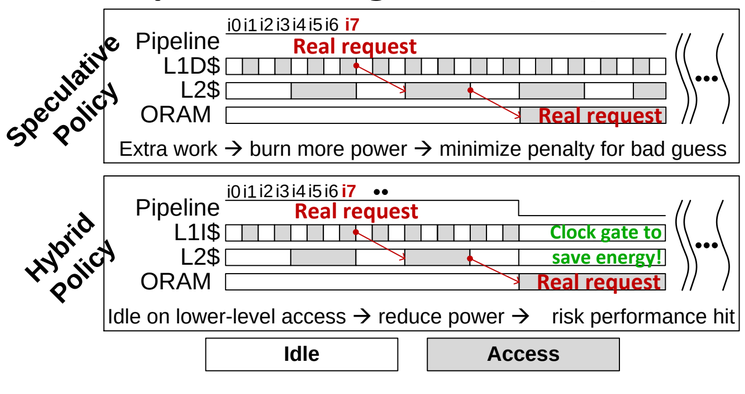
Gate Level Obfuscation



Cache Protocol Level Obfuscation

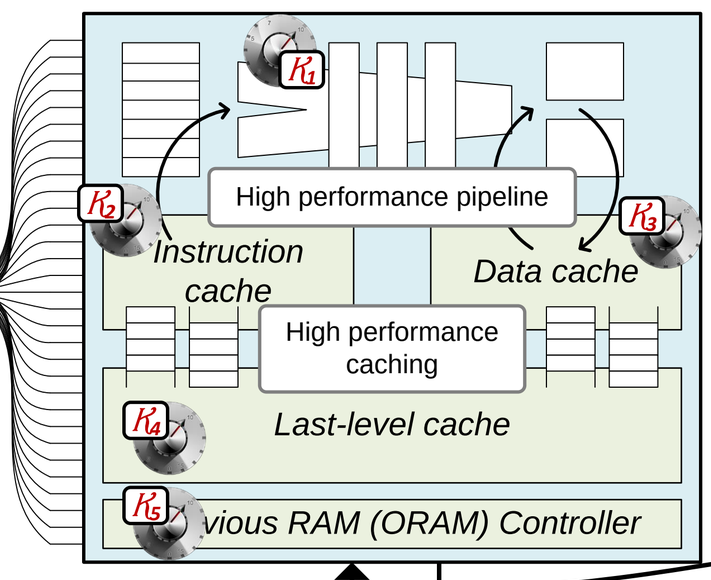


Exploit blocking behavior w/ Knobs



Inputs

Public untrusted program,
Initial Knob settings $K_1, K_2, ...$
Dynamic knob choices $\mathcal{R}$ (if $\mathcal{L} > 0$),
Bit-Leakage Limit ($\mathcal{L}$),
Encrypted Inputs (i.e., @#$KNF@#)

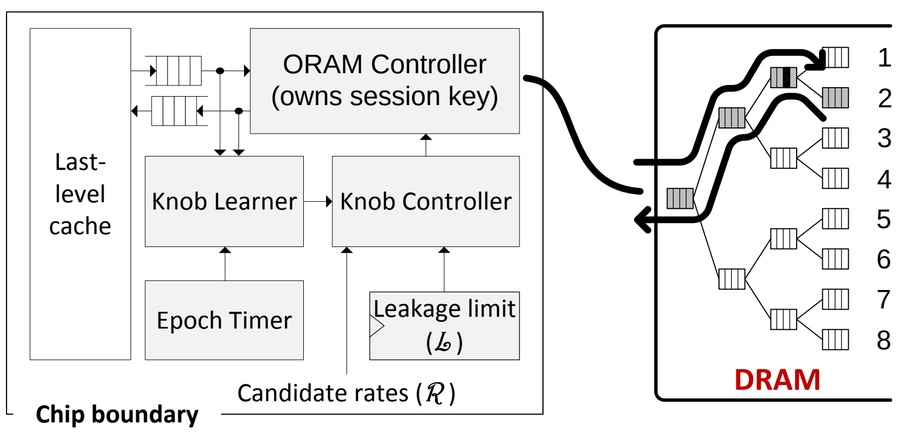


Untrusted external memory (DRAM)

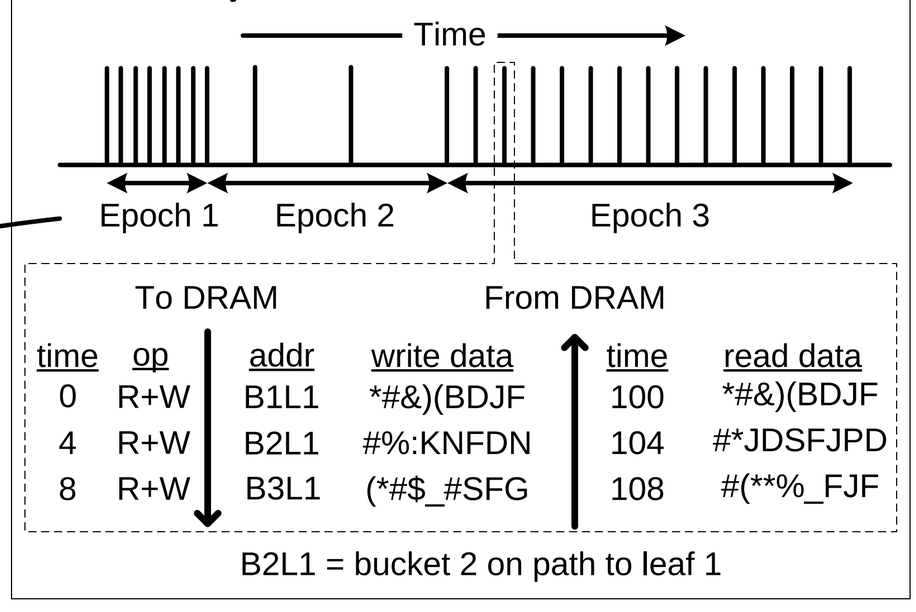
Outputs

Encrypted outputs (i.e., $M#$KLFNC)

Hardware Oblivious RAM (ORAM) with Bounded Timing Channel Leakage:



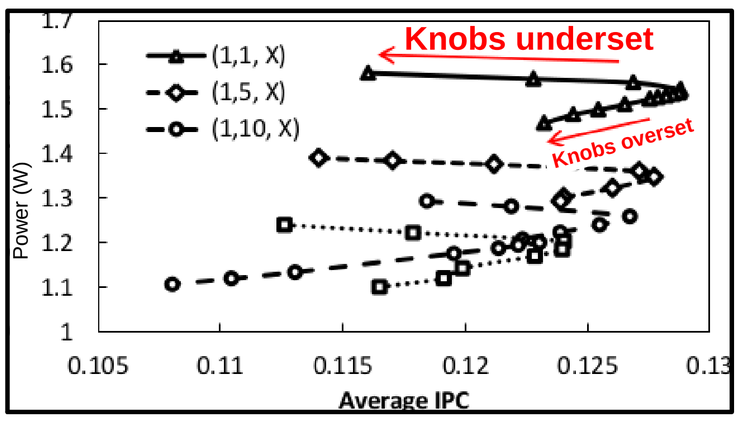
The adverary's view:



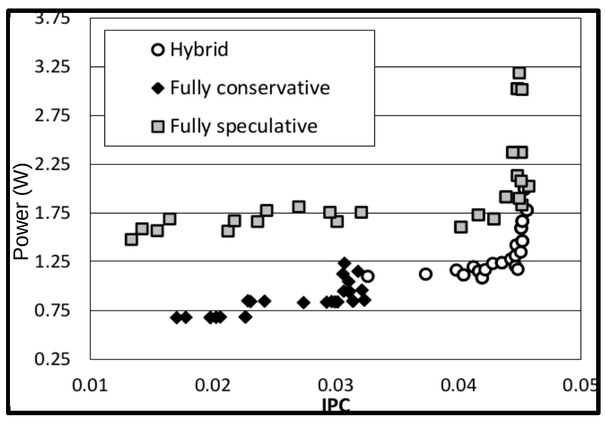
4 I/O Channel

5 Results

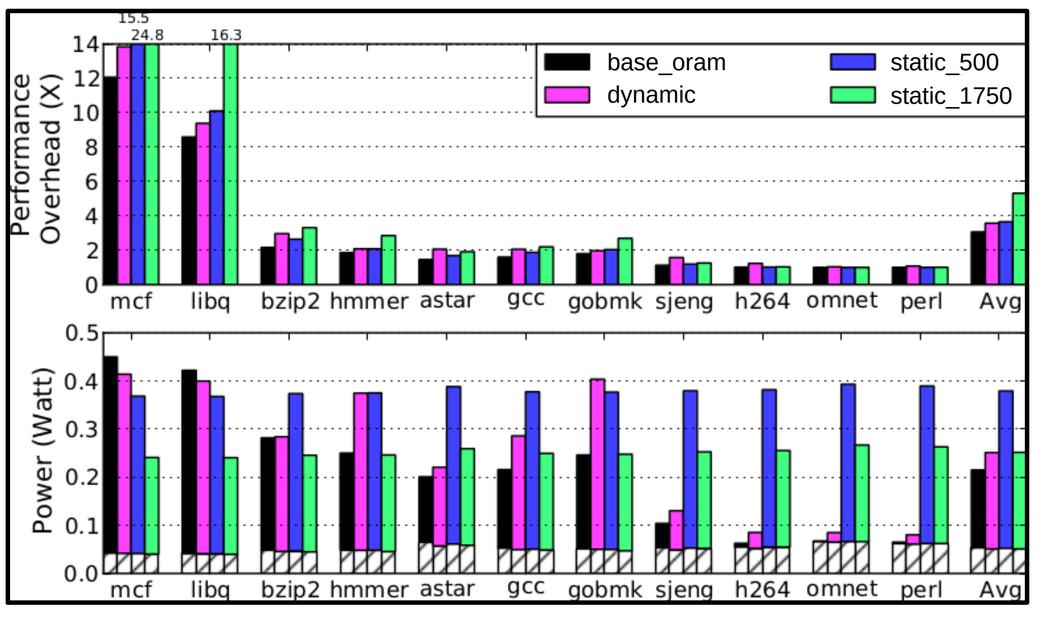
Core model: In order, single issue
On-chip Memory: 1 cycle hit, 32 KB L1 I/D Caches (4 way)
10 cycle hit, 1 MB shared/unified L2 (16 way)
ORAM latency: 1488 cycle hit/64 Byte cache line



[Power+IO obfuscation, static knobs] Pareto curves when changing (L1,L2,ORAM) knobs



[Power+IO obfuscation, static knobs] libq benchmark, varying knob policy



[IO obfuscation only, dynamic knobs] base_oram has no timing protection; static_500 (static knobs) = 500 cycles between each access