COMPUTER Science & Engineering

Fast Secure Processor for Inhibiting Software Piracy and Tampering

Jun Yang, Youtao Zhang*, and Lan Gao
University of California, Riverside
*University of Texas at Dallas

UNIVERSITY OF CALIFORNIA RIVERSIDE

COMPUTER Science & Engineering

Outline

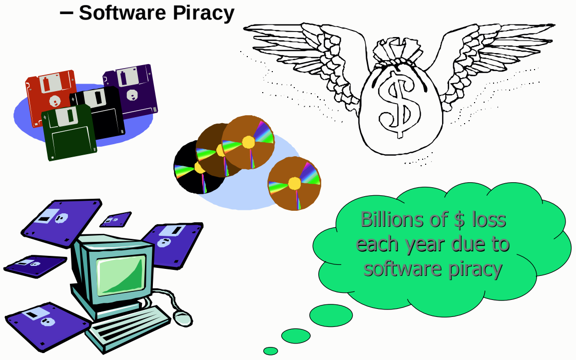
- Motivation
- XOM Architecture Overview
- Offloading Crypto-Computation from Critical Path
- Architecture Design
- Experiment Evaluation
- Summary

UNIVERSITY OF CALIFORNIA RIVERSIDE

COMPUTER Science & Engineering

From Outside Computer

– Software Piracy



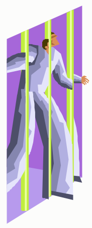
UNIVERSITY OF CALIFORNIA RIVERSIDE

COMPUTER Science & Engineering

From Inside Computer

– Software Tampering

- Debuggers
  - SoftICE, ...
- Disassemblers
  - IDA Pro, ...
- Memory Dump Utilities
  - PEDump, ...



UNIVERSITY OF CALIFORNIA RIVERSIDE

Hardware Support Against Software Piracy

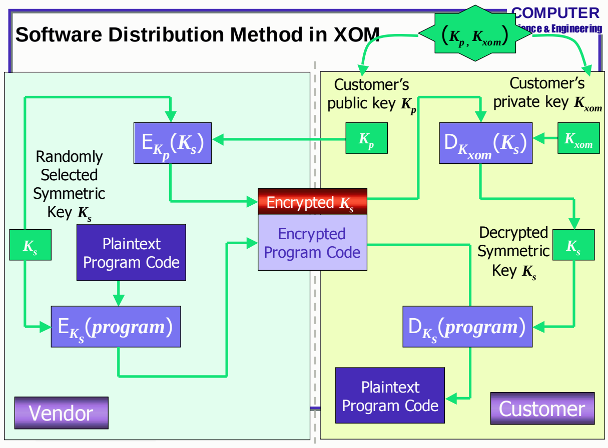
COMPUTER Science & Engineering

- One solution - eXecution Only Memory (XOM)
  [David Lie et al, ASPLOS 2000]
- Who is trustworthy?
  - Only the processor itself is trusted
  - Co-processor, operating system, memory, system bus are NOT tamper resistant
- What needs encryption?
  - Software stored in the system storage
  - Data communicated on the system bus
  - Register values on interrupts

UNIVERSITY OF CALIFORNIA RIVERSIDE 5

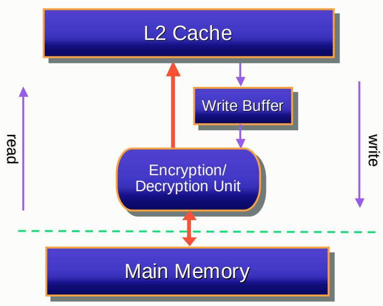
Software Distribution Method in XOM

COMPUTER Science & Engineering



The Lengthened Memory Path in XOM

COMPUTER Science & Engineering



- The crypto-hardware lies on the memory access critical path, thus inserts extra latency.

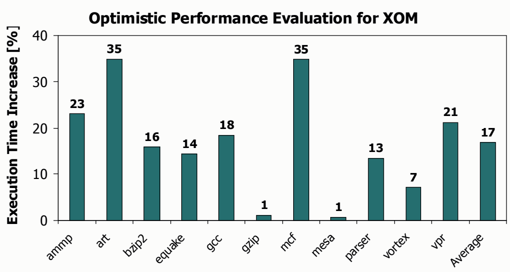
UNIVERSITY OF CALIFORNIA RIVERSIDE 7

Performance Degradation in XOM

COMPUTER Science & Engineering

Assume the following latencies:

- Average Memory Access — 100 cycles
- Encryption/Decryption — 50 cycles (optimistic)



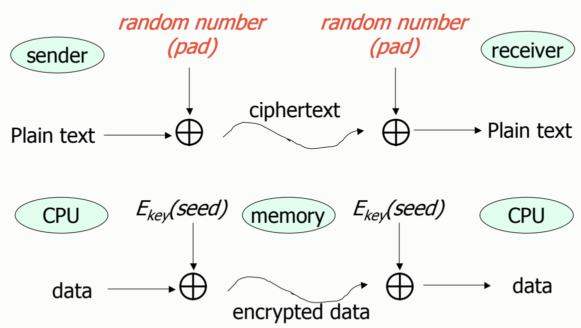
UNIVERSITY OF CALIFORNIA RIVERSIDE 8

Offloading Crypto-Computation from Critical Path

- ❑ The crypto-computation in XOM:
  - ► Data dependent on memory accesses
  - ► Carried in serial with the memory accesses
- ❑ Our One-Time Pad based scheme:
  - ► Decouple en/decryption from memory access
  - ► They can be carried in parallel
- ❑ The memory encryption scheme: G. Edward Suh et al
  - ► Similarity: One-Time-Pad Encryption
  - ► Major Difference:
    - § Timestamp storage: off-chip v.s. on-chip

COMPUTER
Science & Engineering

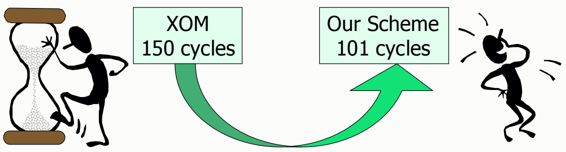
One-Time Pad (OTP) Encryption



COMPUTER
Science & Engineering

Speed Up XOM

- ❑ Let us assume:
  - ► memory access latency = 100 cycles
  - ► encryption/decryption latency = 50 cycles
  - ► XOR needs 1 cycle
- ❑ Memory access latency with crypto operations:



COMPUTER
Science & Engineering

Two Issues

- ❑ OTP based encryption strength
  - ► in authentic OTP:
    - strength(ciphertext) Ξ strength(random number)
  - ► in our scheme:
    - strength(encrypted data) Ξ strength(E_{key}())
- ❑ Seed selection
  - ► independent of data value, known before data is available — address
  - ► multiple accesses of same data use different seeds — one-time seed

Introducing Sequence Numbers

Write V → A

time	t_0	t_1	t_2 …
values at *A*	1	2	3 …
(1) XOM	$E_{key}(1)$	$E_{key}(2)$	$E_{key}(3)$ …
(2) Use A only	$E_{key}(A) \oplus 1$	$E_{key}(A) \oplus 2$	$E_{key}(A) \oplus 3$ …
(3) Use A and t	$E_{key}(A+t_0)\oplus 1$	$E_{key}(A+t_1)\oplus 2$	$E_{key}(A+t_2)\oplus 3$

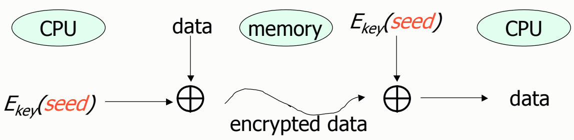
seed = address + timestamp

Comparing XOM and OTP Based XOM

			XOM	XOM w/ OTP
spatially	A_1	100	A_1 $E_{key}(100)$	A_1 $E_{key}(A_1+t_1)\oplus 100$
	A_2	100	A_2 $E_{key}(100)$	A_2 $E_{key}(A_2+t_2)\oplus 100$
temporally A	t_1	100	t_1 $E_{key}(100)$	t_1 $E_{key}(A+t_1)\oplus 100$
	t_2	100	t_2 $E_{key}(100)$	t_2 $E_{key}(A+t_2)\oplus 100$

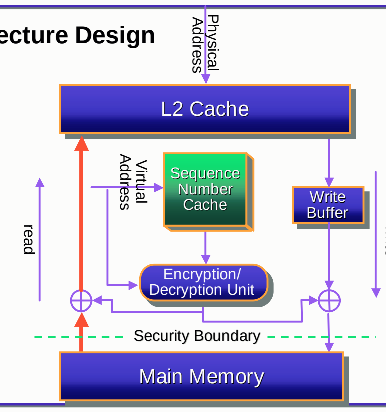
➢ Our scheme better randomizes encrypted data in memory

Seed Storage



- Store seeds in memory
  - not beneficial since mem. accesses are doubled
- Use an on-chip cache to remember seeds
  - need only to store the sequence numbers (t_i) since they can be narrower than the seeds

Architecture Design



SNC Capacity is Limited

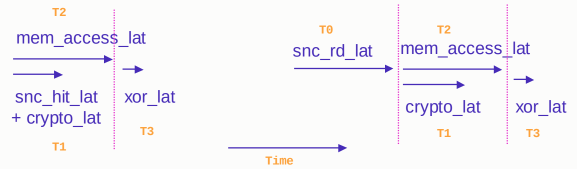
COMPUTER Science & Engineering

- ❑ Stop using OTP once it's full
  - ▶ only partial memory blocks have seeds
  - ▶ simple control, good for programs with modest mem. requirement
- ❑ Use replacement (LRU) to store all the seeds
  - ▶ Three ways to store evicted sequence numbers
    - § Encrypt using one-time pad
      - – They themselves would need the sequence numbers!
    - § Encrypt directly as XOM
      - – Increase memory access latency
    - § Store plaintext
      - – secure since the private key is not revealed

SNC Operation: Best and Worst Cases

COMPUTER Science & Engineering

❑The best case (hit SNC): ❑The worst case (miss SNC):



Lat_{best} = MAX(T1,T2) + T3 = 101 cycles

Lat_{worst} = T0 + MAX(T1,T2) + T3 = 201 cycles

Other Issues

COMPUTER Science & Engineering

- ❑ Context Switching
  - ▶ Flush SNC to the memory
  - ▶ Tag each entry with XOM ID
- ❑ Shared library and program inputs
  - ▶ Both should be provided in plaintexts
  - ▶ Do not need sequence numbers in SNC

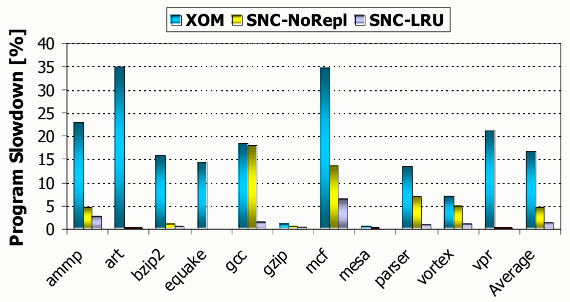
Experiments

COMPUTER Science & Engineering

- ❑ Tools
  - ▶SimpleScalar V3.0
  - ▶11 SPEC2000 benchmarks
- ❑ Baseline
  - ▶4-issue out-of-order processor
  - ▶Caches:
    - § Separate L1 I-cache and D-cache: 32KB, 4-way
    - § Unified L2 cache: 256KB, 4-way, 128B/line
  - ▶Latencies:
    - § Memory access latency: 100 cycles
    - § Encryption latency: 50 cycles
- ❑ Execution
  - ▶Fast forwarded by 10 billion instructions
  - ▶Then execute 10 billion instructions

Performance Comparison

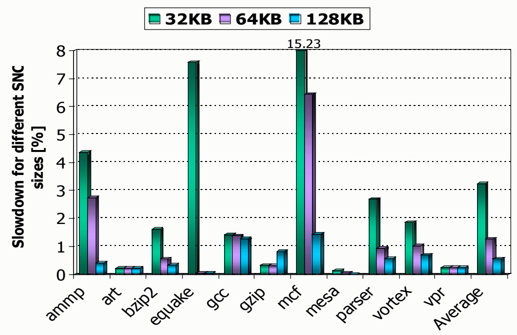
COMPUTER
Science & Engineering



UNIVERSITY OF CALIFORNIA
RIVERSIDE

SNC of Different Size

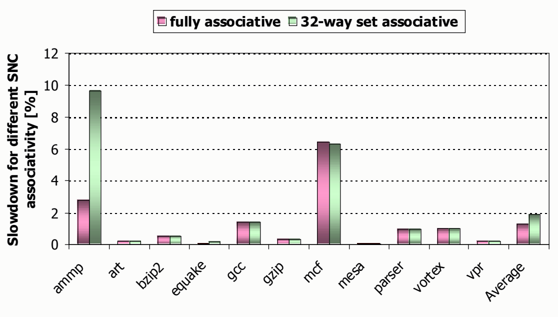
COMPUTER
Science & Engineering



UNIVERSITY OF CALIFORNIA
RIVERSIDE

SNC of Different Associatively

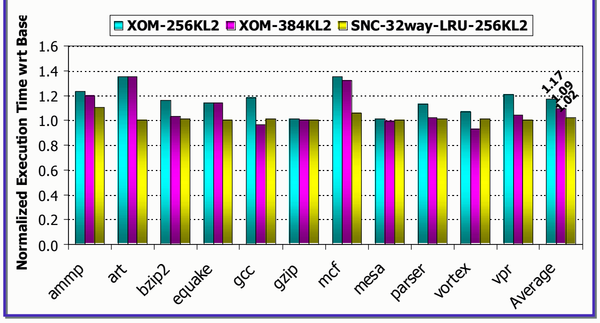
COMPUTER
Science & Engineering



UNIVERSITY OF CALIFORNIA
RIVERSIDE

Equal Area Comparison

COMPUTER
Science & Engineering



UNIVERSITY OF CALIFORNIA
RIVERSIDE

Other Experiments

COMPUTER
Science & Engineering

- SNC Induced Memory Traffic
  - On average, there is only 0.31% of the L2 memory traffic posed by SNC replacements on to the system bus
- Sensitivity to Encryption Latency
  - XOM degrades greatly from 16.7% to 34.2% slowdown
  - The performance of our design with LRU replacements is almost unchanged

UNIVERSITY OF CALIFORNIA RIVERSIDE

25

Conclusion

COMPUTER
Science & Engineering

- Apply one-time pad (OTP) cryptography to speed up the secure processor
- Develop the hardware support
- Reduce the performance overhead from 16.7% for critical path cryptography to 1.28% for OTP cryptography

UNIVERSITY OF CALIFORNIA RIVERSIDE

26

COMPUTER
Science & Engineering

Questions ?

UNIVERSITY OF CALIFORNIA RIVERSIDE

27