Hardware Security Support in General Purpose Processors

Mihai Christodorescu
mihai@cs.wisc.edu

Hao Wang
hbwang@cs.wisc.edu

University of Wisconsin, Madison

Overview

- Concepts
- Architecture
- Performance
- Conclusions
- Future Directions

Secure Processing

Secure Processing

- Goals:
  - Copy and tamper-resistant software
  - Computation privacy

Secure Processing

- Goals:
  - Copy and tamper-resistant software
  - Computation privacy
- How?
  - Software solutions
  - Hardware solutions
    - Adding security features to microprocessors

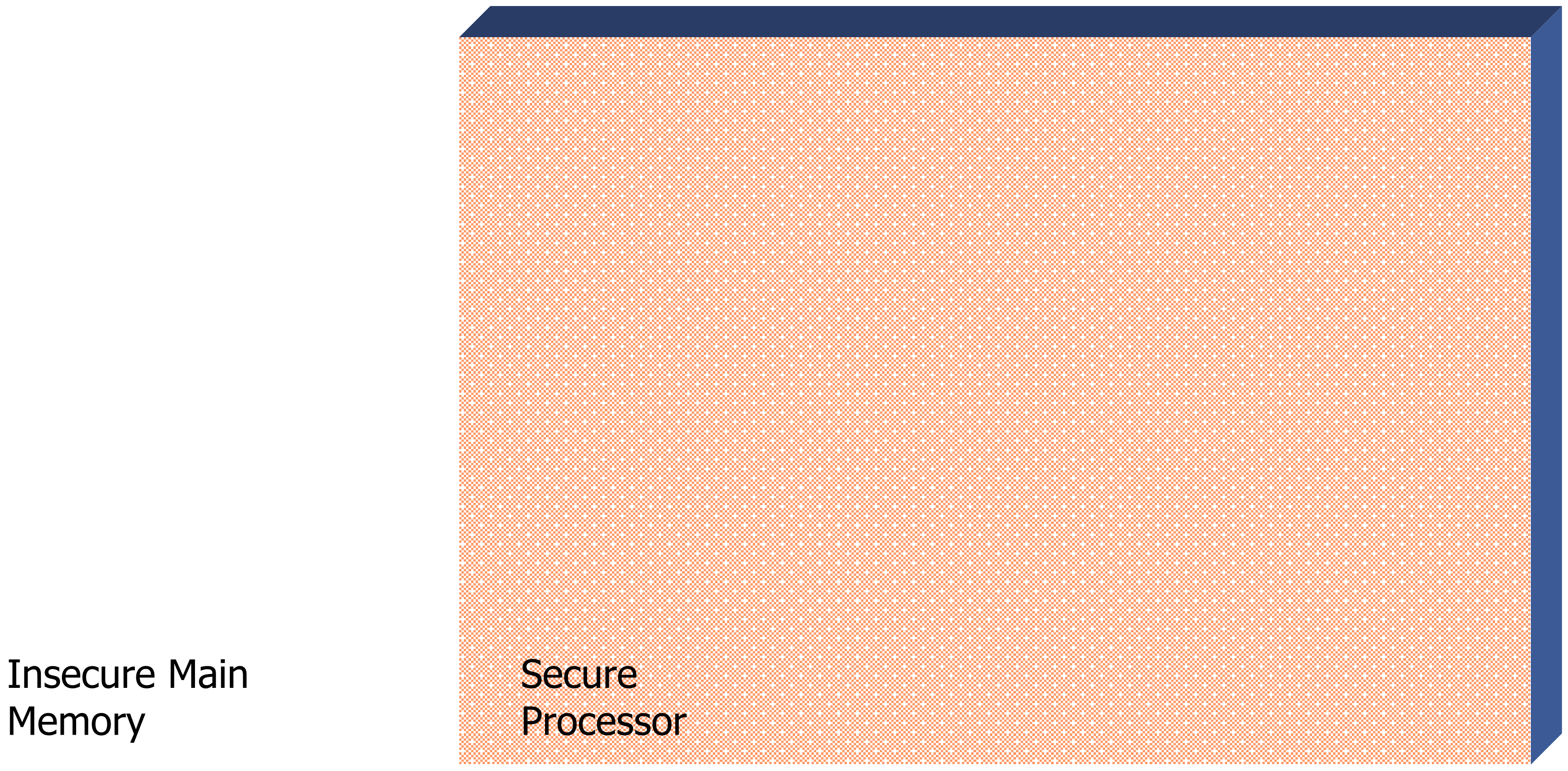
Secure Processing

- Goals:
  - Copy and tamper-resistant software
  - Computation privacy
- How?
  - Software solutions
  - Hardware solutions
    - Adding security features to microprocessors
- Really? – Yes!
  - Intel claims billion-transistor in five years

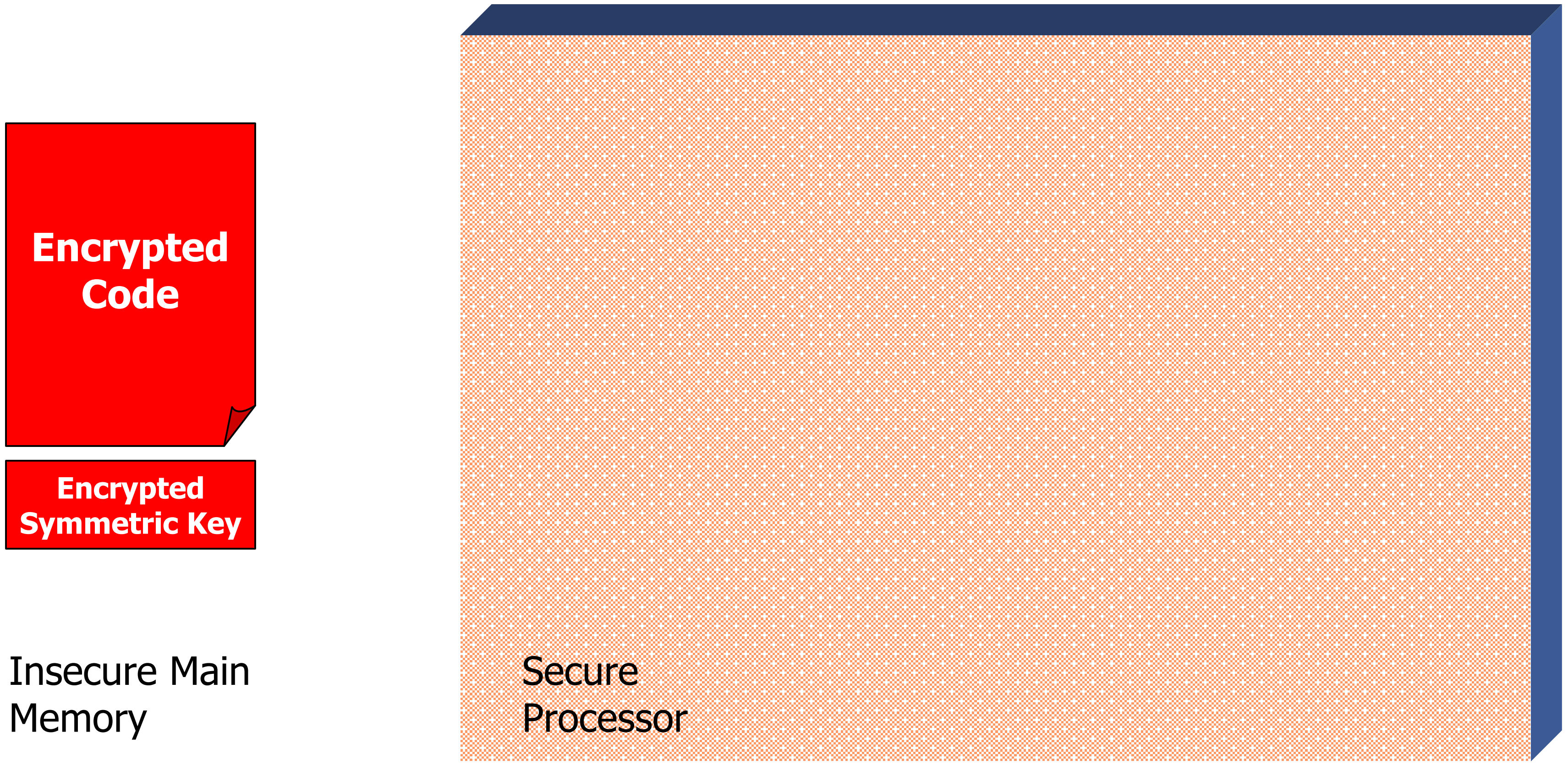
Our Vision

- Secure processor
  - General purpose CPU with Crypto support
- Two approaches
  - Secure Co-Processor (Michigan)
  - Integrated Secure Processor
    - Processor is the only trusted component
    - XOM : e**X**ecute **O**nly **M**emory (Stanford)
      - Encrypt code and data

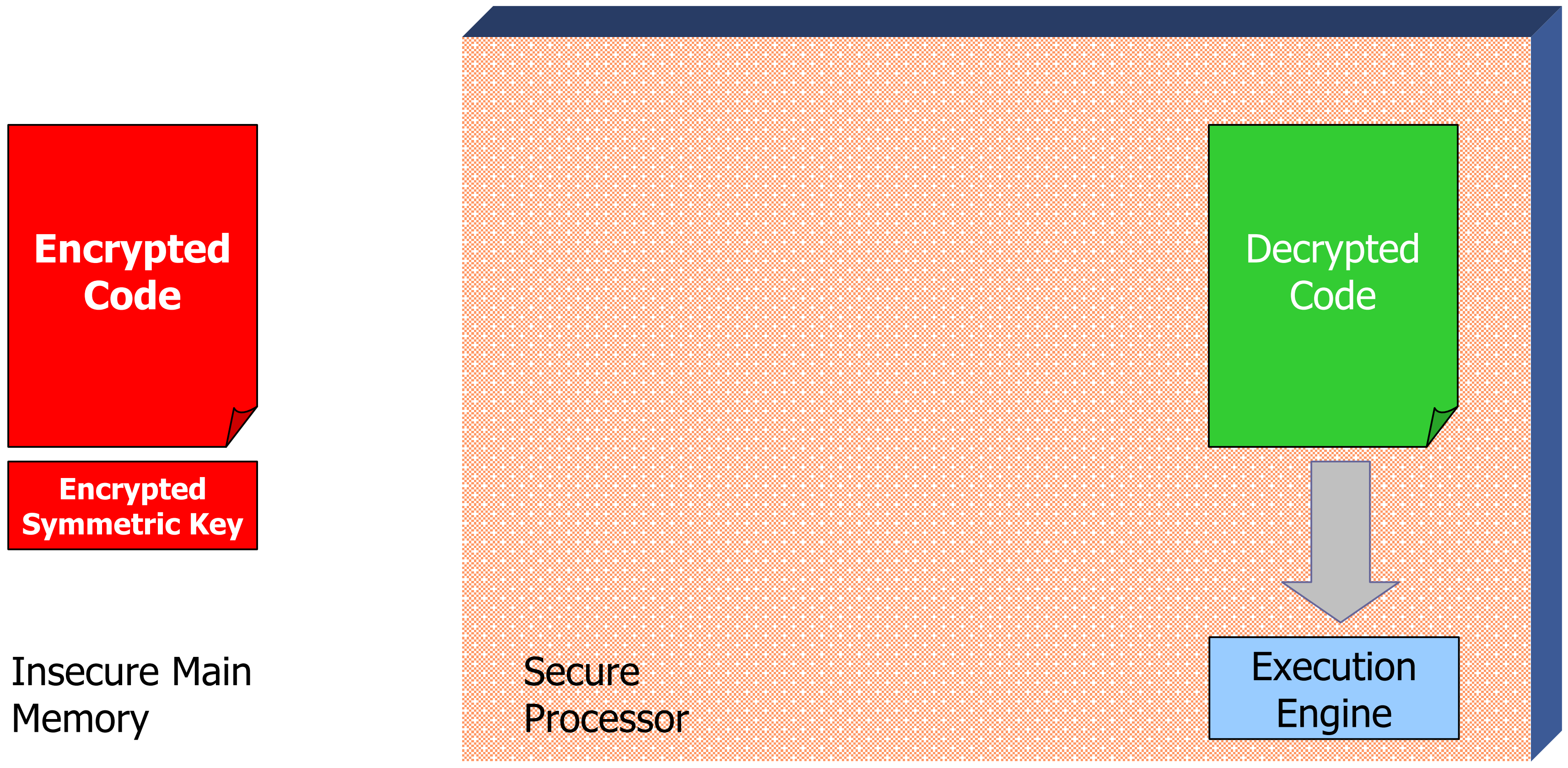
XOM – Initialization



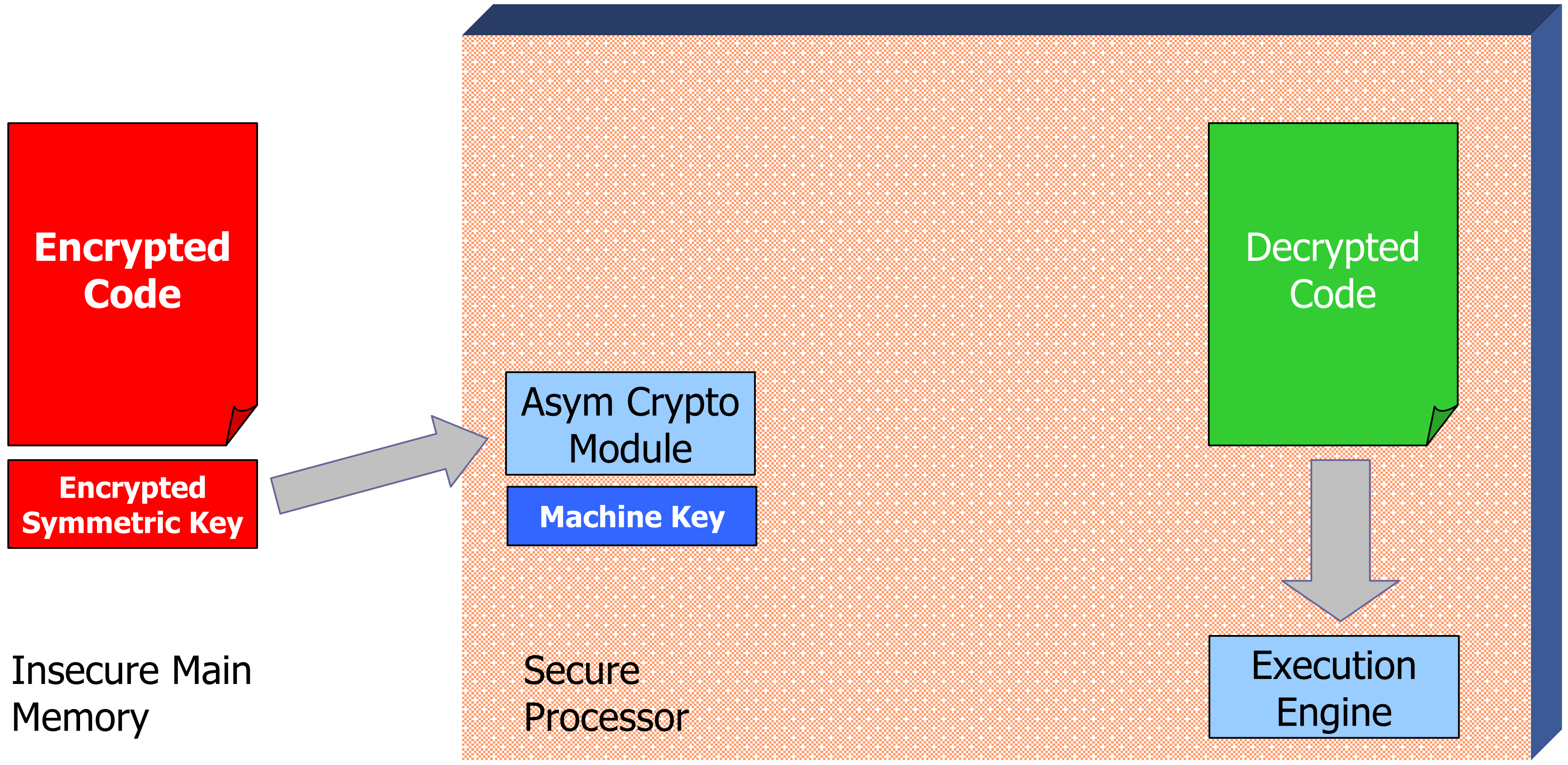
XOM – Initialization



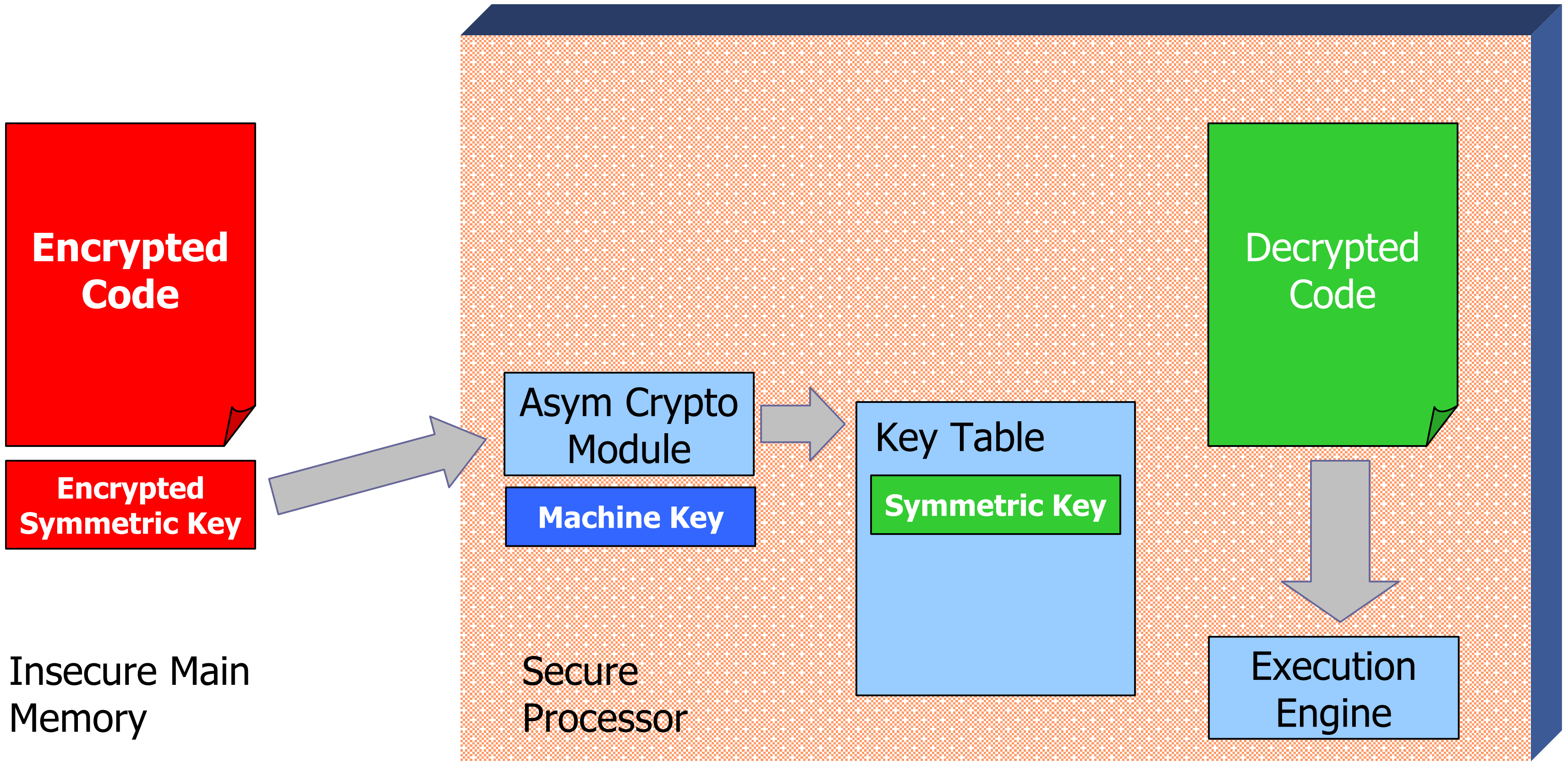
XOM – Initialization



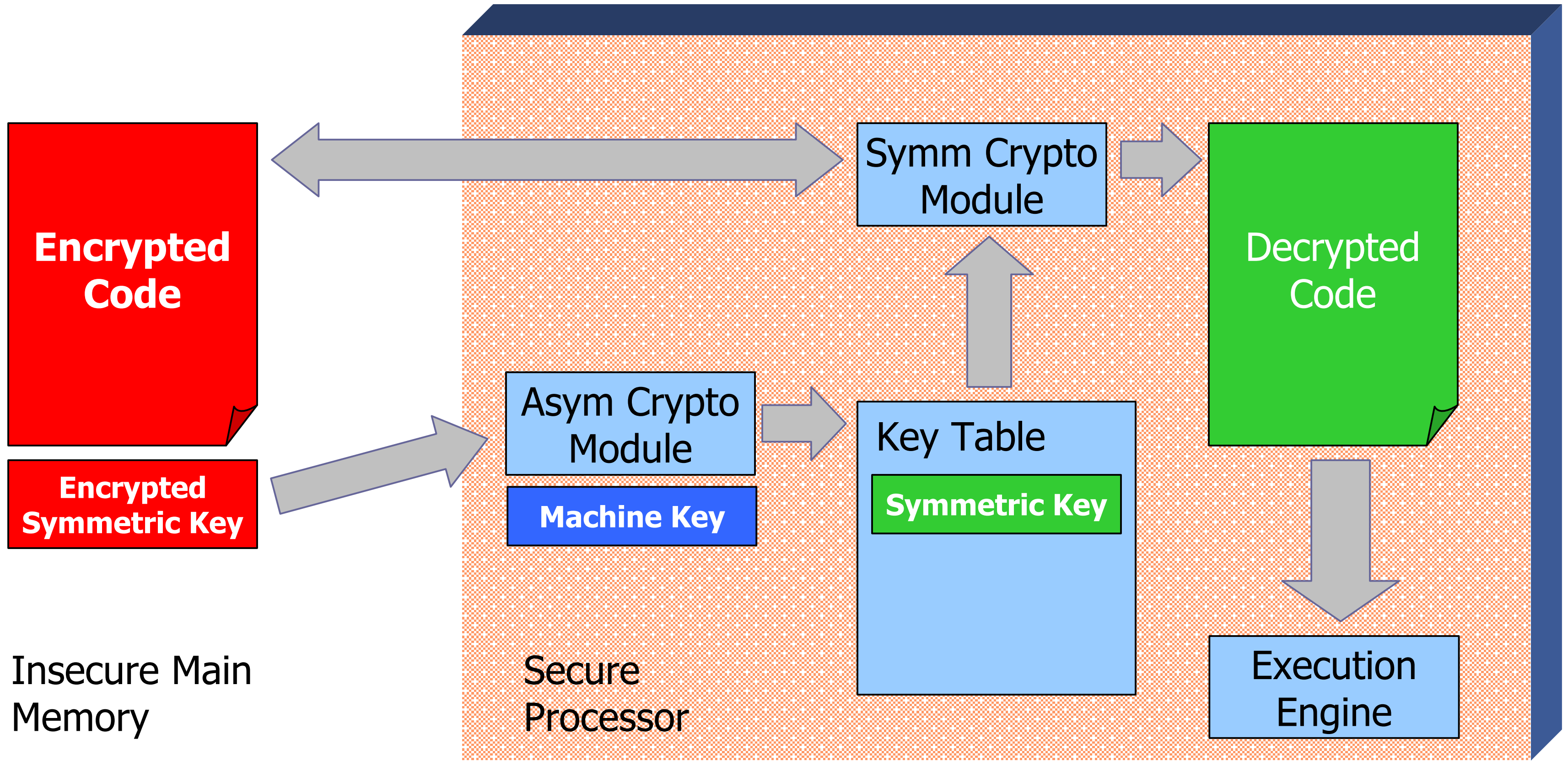
XOM – Initialization



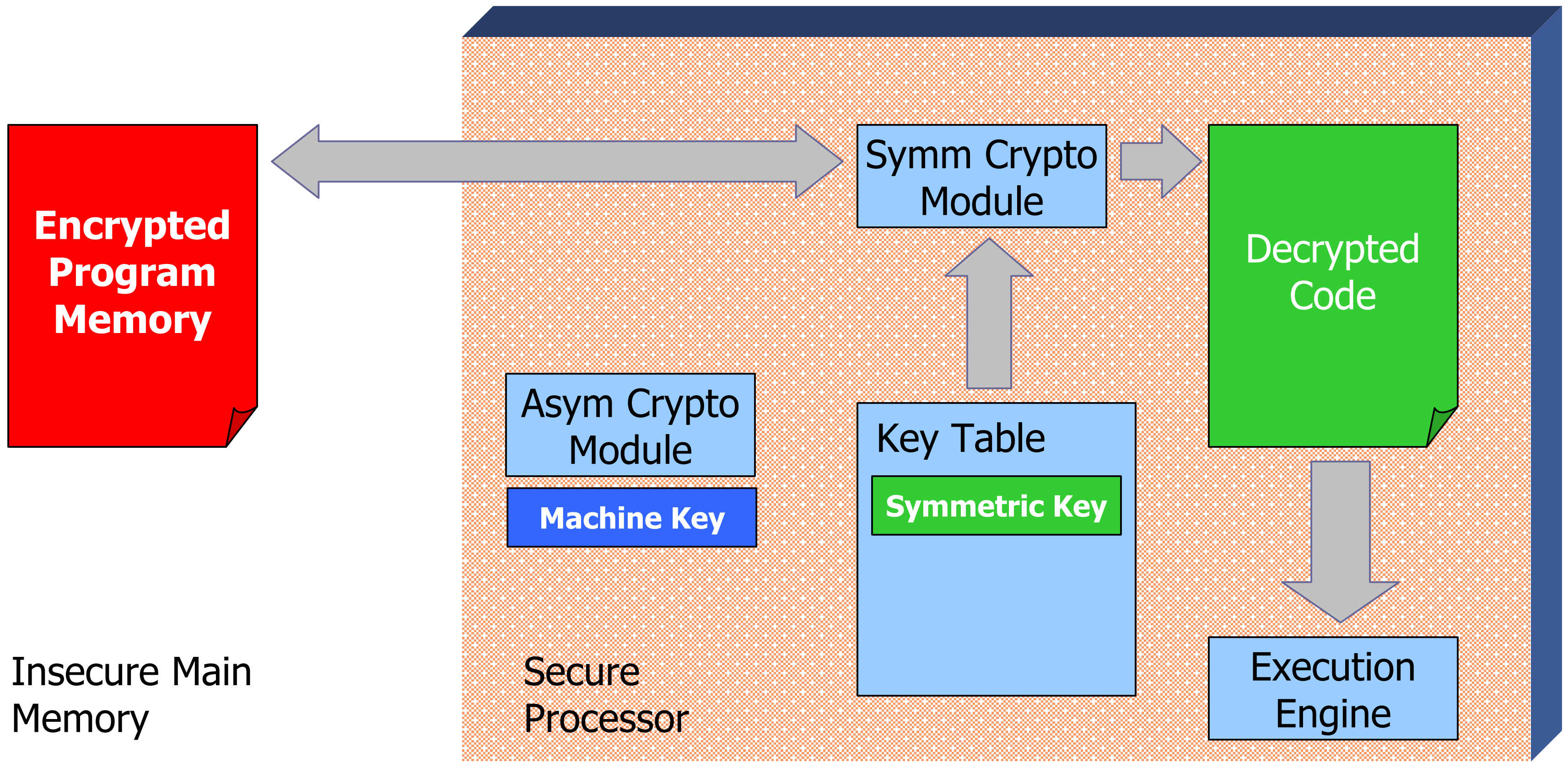
XOM – Initialization



XOM – Initialization



XOM – Execution



XOM Benefits

XOM Benefits

- Encrypted Program Memory
  - No “hijacking”
  - Secure core dumps!

XOM Benefits

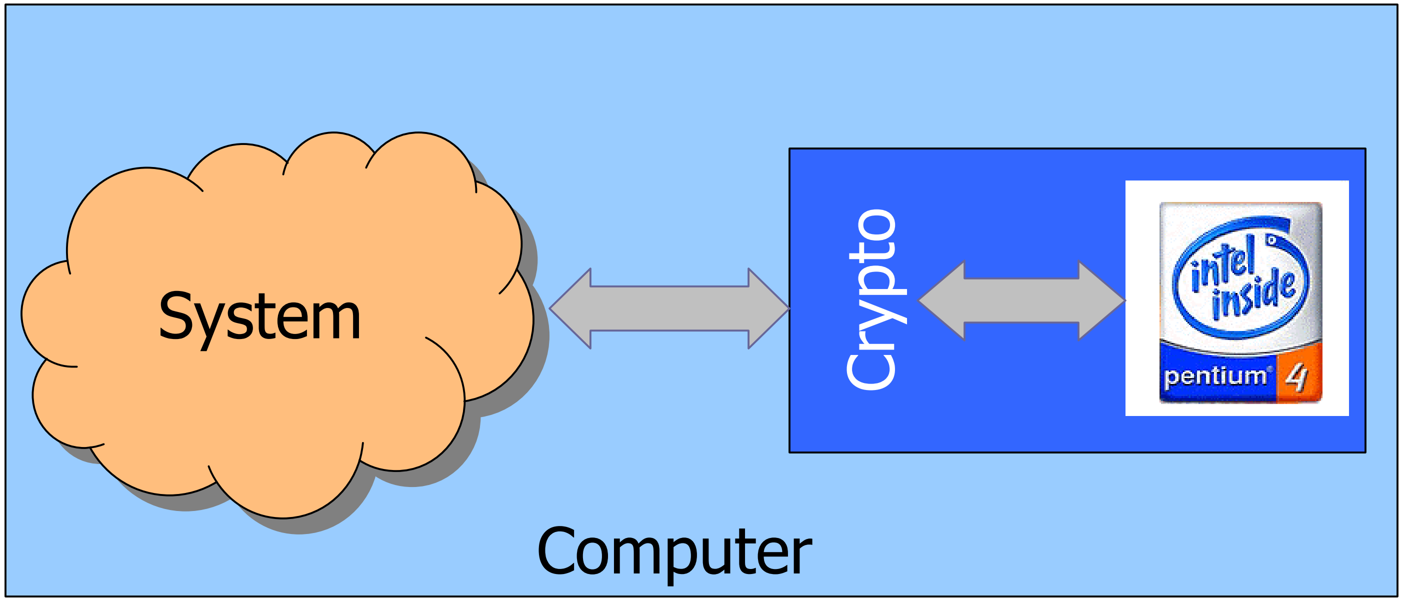
- Encrypted Program Memory
  - No “hijacking”
  - Secure core dumps!
- Encrypted Executable on Disk
  - No viruses
  - No binary modification (e.g. cracks)
  - No evil reverse engineering

Crypto Hardware – AES

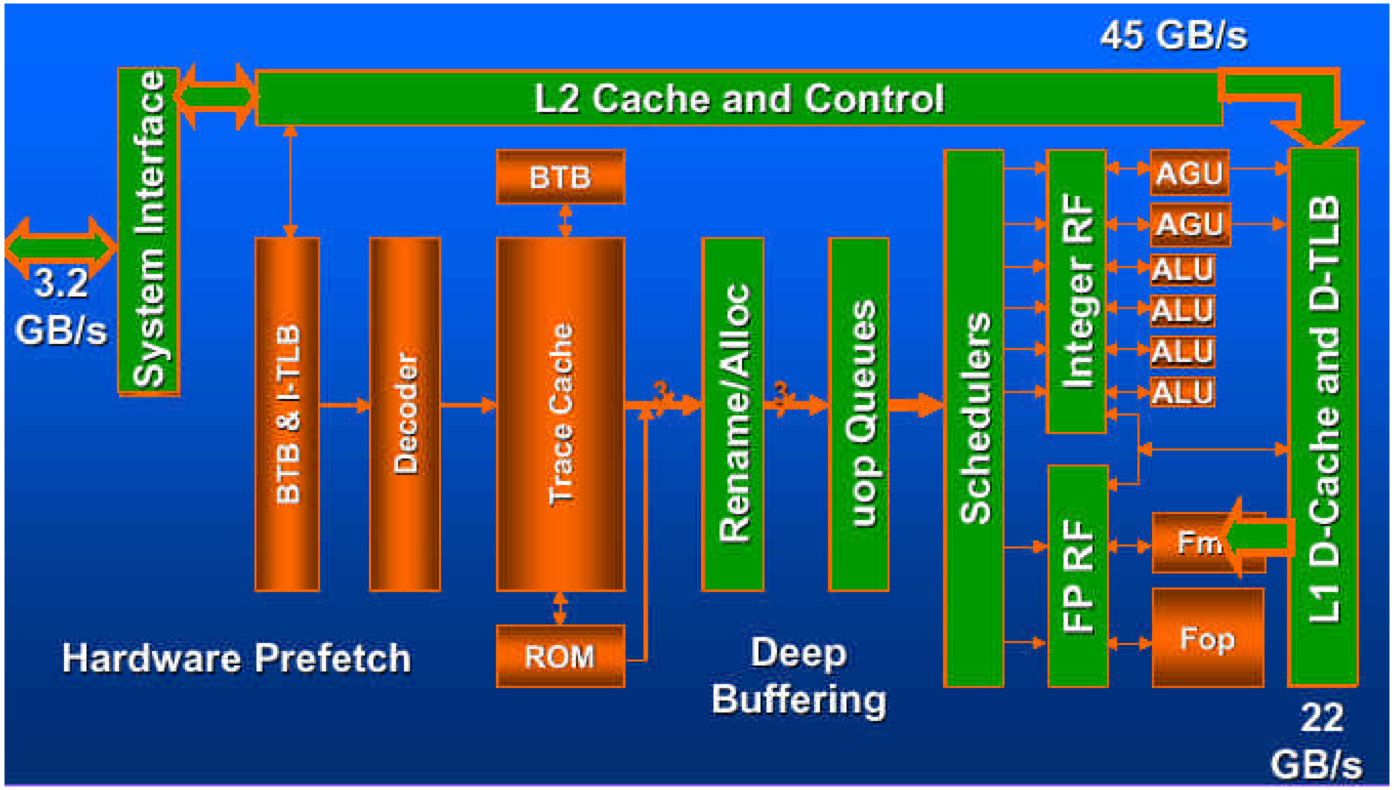
- AES : **A**dvanced **E**ncryption **S**tandard
  - Replaces DES (at last!)
  - Uses Rijndael
- AES specifications
  - Block cipher
  - Fixed block size: 128 bit
  - Variable key size (128, 192, 256-bit)

Simulated Hardware

- Pentium 4 + AES encryption

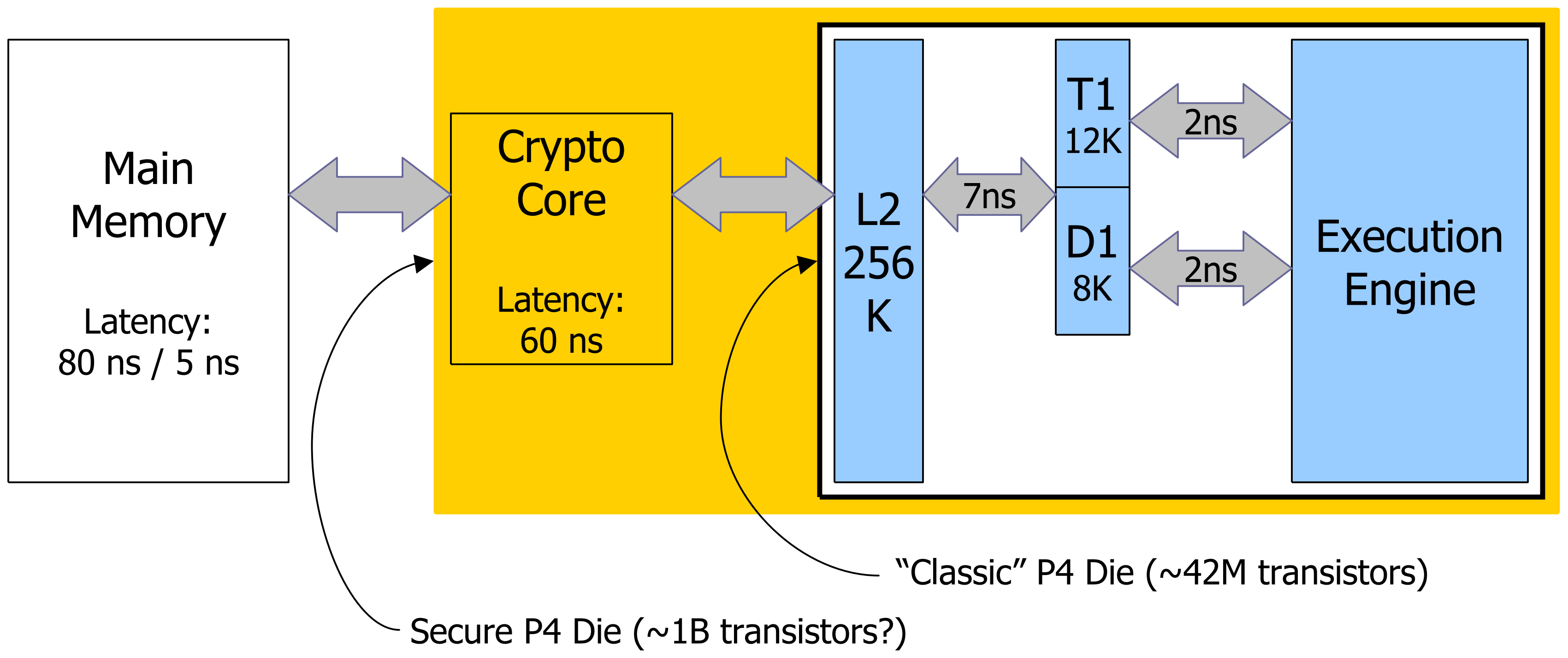


A Better Looking Picture



Copyright 2001 Intel

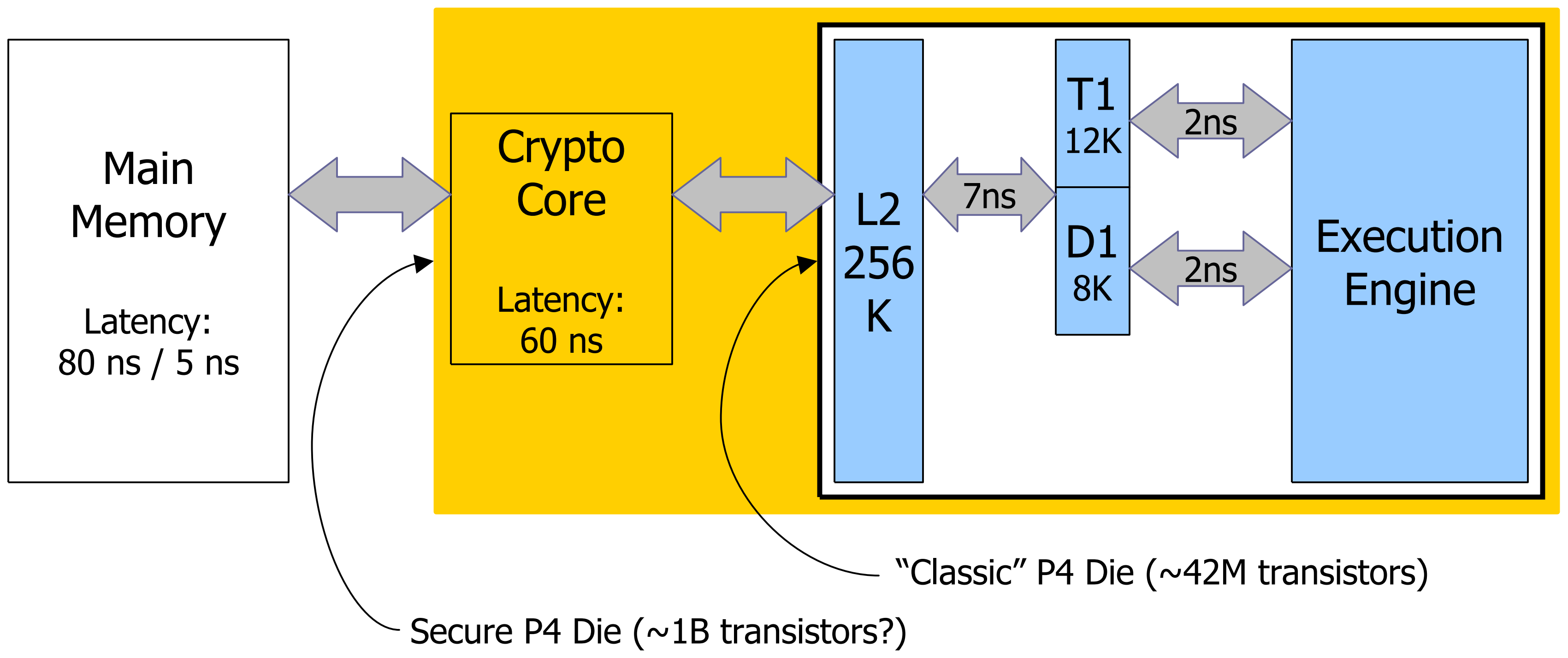
Simulated Hardware



Fixed simulation parameters:

- Cache associativity: 4-way D1, 8-way L2
- Cache block size: 64 B D1, 256 B L2

Simulated Hardware



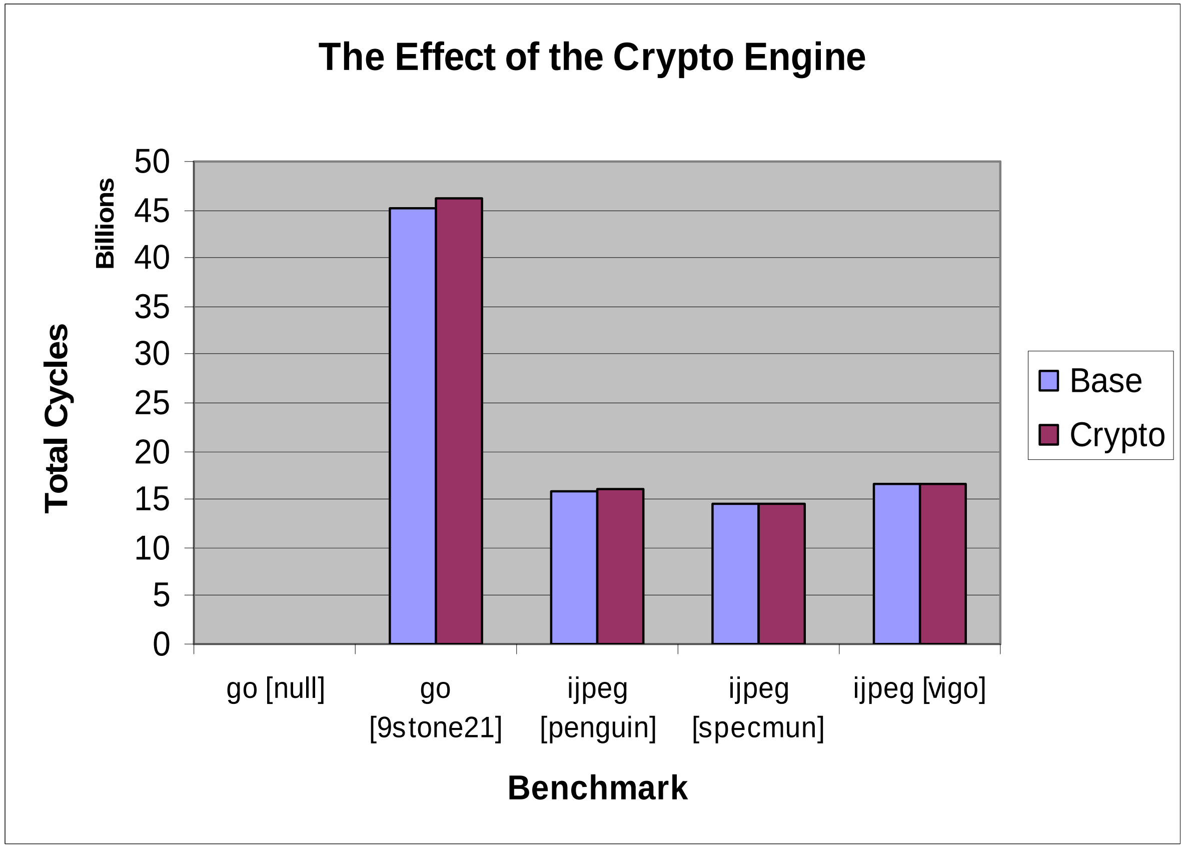
Variable simulation parameters:
- Memory bus width: 8 B, 16 B
- L2 cache size: 256 K, 512K
- Crypto key size: 128, 192, 256 bits

Fixed simulation parameters:
- Cache associativity: 4-way D1, 8-way L2
- Cache block size: 64 B D1, 256 B L2

Simulation Study

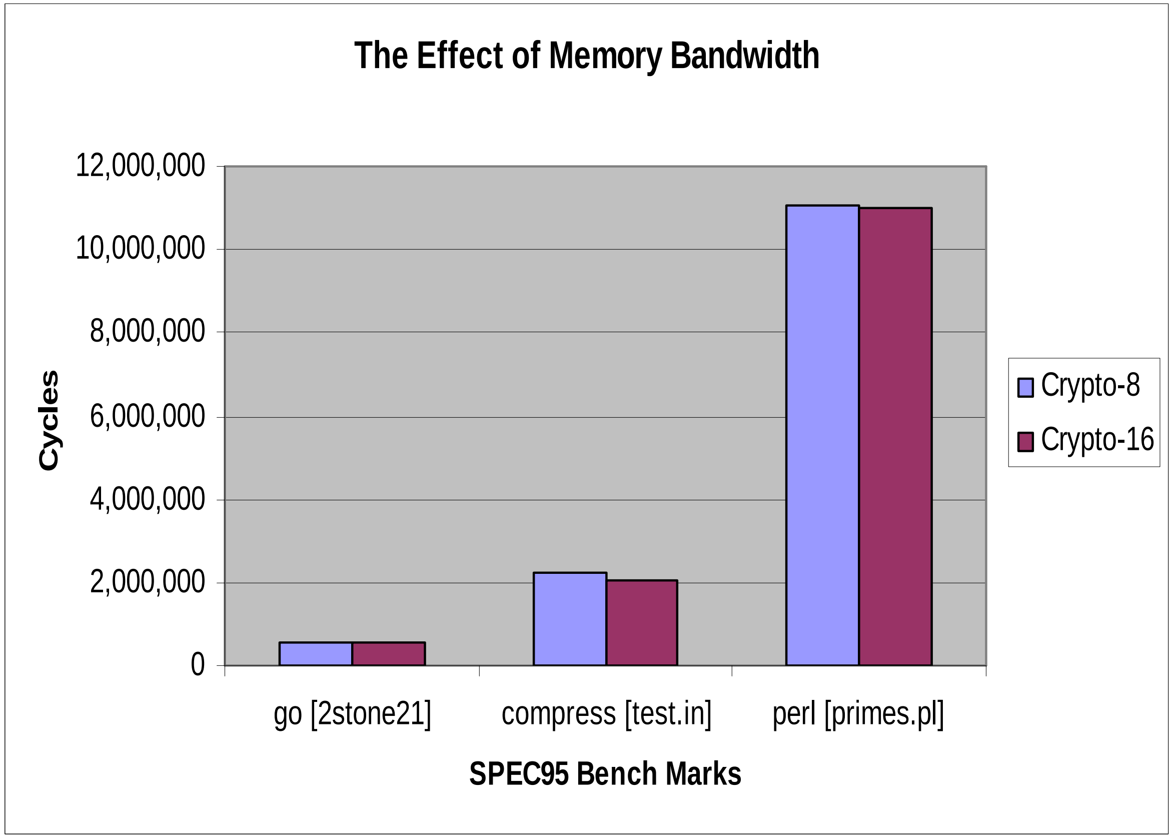
- Goal:
  - As little interference with the execution engine as possible
- SimpleScalar Out-of-Order simulator
  - Better at hiding latency
  - Cache size effect on performance
    - L2 miss now includes crypto latency
    - L1 should not matter that much
  - Memory bandwidth effect

Performance Measurements



Benchmark	Slowdown
go [null]	2.997%
go [9stone21]	2.037%
ijpeg [penguin]	0.796%
ijpeg [specmun]	0.814%
ijpeg [vigo]	0.791%

Performance Measurements

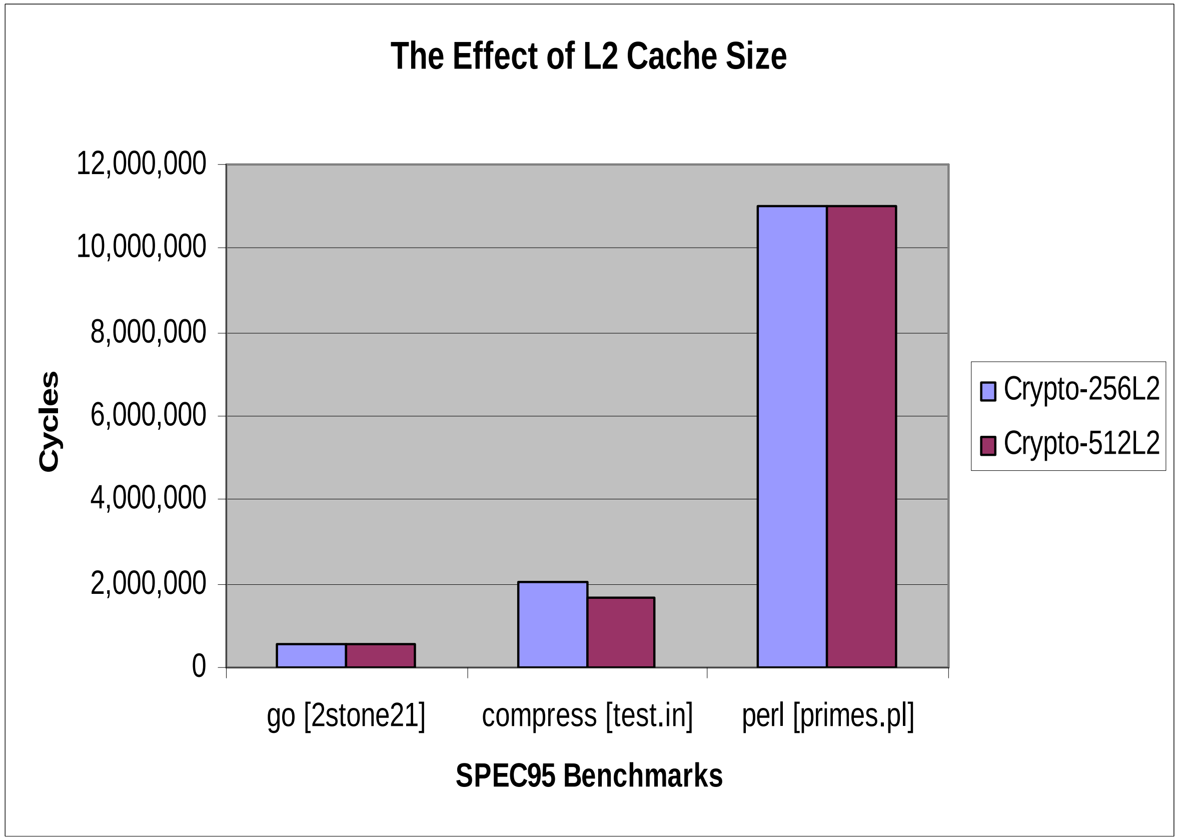


Benchmark	Speedup
go [2stone21]	4.23%
compress [test.in]	8.77%
perl [primes.pl]	0.70%

Crypto-8:
8 bytes memory Bus width

Crypto-16:
16 bytes Memory Bus width

Performance Measurements



Benchmark	Speedup
go [2stone21]	0.51%
compress [test.in]	19.86%
perl [primes.pl]	0.07%

Secure Processor

Benchmark	Speedup
go [2stone21]	0.37%
compress [test.in]	14.31%
perl [primes.pl]	0.05%

Base Processor

Analysis

- Adding Crypto Hardware = Slower Memory
- Memory bus width has an impact on total latency
  - Effective Latency =
    TI + (Size(C)/MemBusBand - 1)* Mem. Int. Lat. + Crypto Latency
    - TI is the base initial memory latency
    - Size(C) is the cipher block size

Conclusions

- Feasible and desirable!
- Some overhead
  - Increases memory latency
    - Solution: hide the latency
      - wider memory bus
      - bigger cache
  - Requires extra hardware
    - For encryption/decryption
    - For storing keys
    - Not a big problem

Future Directions

- Problems
  - I/O
  - Multiprocessors
  - Shared libraries (DLLs, .so)
- Increase efficiency
  - Partial program encryption
- Increase security
  - Per-user-process encryption

References

- David Lie et al. “Architectural Support for Copy and Tamper Resistant Software”, ASPLOS-IX 2000
- XOM Project
  - http://www.stanford.edu/~davidlie/xom.htm
- FIPS-197: AES
  - http://csrc.nist.gov/encryption/aes
- Intel Developer’s Website (P4 data facts)
  - http://developer.intel.com
- Rambus (information on memory latency)
  - http://www.rambus.com
- and much more ….