PERFORMANCE ANALYSIS OF PARALLEL XOR AND AES ENCRYPTION ON HETEROGENEOUS ARCHITECTURES: APPLE SILICON M4 PRO VS NVIDIA RTX 3070

Maciej Biegan, Mateusz Nyko, Filip Kruzel

Department of Computer Science, Cracow University of Technology

KEYWORDS

Parallel Computing, Heterogeneous Architectures, Unified Memory, AES-256-CTR, Apple Silicon, CUDA, OpenMP.

ABSTRACT

This paper presents a comparative analysis of encryption performance between two fundamentally different computing architectures: Apple M4 Pro with unified memory and Intel i5-8600K paired with NVIDIA RTX 3070 discrete GPU. We evaluate both memory-bound (XOR) and compute-bound (AES-256-CTR) algorithms across sequential, OpenMP, Metal, and CUDA implementations. Our results reveal that the M4 Pro achieves 8.7 GB/s sequential AES throughput compared to 3.5 GB/s on the Intel platform, primarily due to dedicated ARMv8 cryptographic instructions. For parallel XOR operations, the M4 Pro reaches 7.8 GB/s with OpenMP, while the RTX 3070 achieves 3.8 GB/s via CUDA. The unified memory architecture eliminates PCIe transfer bottlenecks that limit discrete GPU performance in data-intensive encryption tasks. Energy measurements show the M4 Pro consumes 15-30 watts during peak operation compared to 220 watts for the RTX 3070 system, resulting in superior energy efficiency for the Apple platform across all tested workloads.

INTRODUCTION

Modern computing systems increasingly rely on encryption for data protection. The growing volume of encrypted data demands high-performance cryptographic implementations that can process gigabytes per second while maintaining energy efficiency. Traditional approaches leverage discrete GPUs for parallel processing, but these solutions face inherent limitations from PCIe bus bandwidth constraints during host-device data transfers.

The emergence of unified memory architectures, exemplified by Apple Silicon processors, presents an alternative approach. These systems eliminate the distinction between CPU and GPU memory spaces, potentially removing data transfer bottlenecks that plague discrete GPU implementations. However, the performance implications of this architectural choice for encryption workloads remain understudied.

This paper addresses two research questions. First, how does unified memory architecture affect encryption throughput compared to discrete GPU systems? Second, what are the energy efficiency implications of each approach? We answer these questions through systematic benchmarking of XOR and AES-256-CTR algorithms across multiple implementation strategies.

The XOR algorithm serves as a memory-bound baseline, where performance depends primarily on memory bandwidth rather than computational capacity. AES-256-CTR represents compute-bound encryption, requiring significant arithmetic operations per byte processed. Together, these algorithms characterise the performance envelope for symmetric encryption workloads.

HARDWARE PLATFORMS

Our experimental setup comprises two distinct computing platforms representing different architectural philosophies. The first platform uses an Apple MacBook Pro with the M4 Pro processor running macOS 15.1. The second platform combines an Intel i5-8600K processor with an NVIDIA GeForce RTX 3070 graphics card running Windows 11 through WSL2.

Apple M4 Pro Architecture

The M4 Pro processor integrates 14 CPU cores in a heterogeneous configuration: 10 high-performance cores operating at 4.51 GHz and 4 efficiency cores at 2.6 GHz. The chip is fabricated using TSMC's 3nm process technology, enabling high transistor density within a 26-watt thermal envelope.

The unified memory architecture provides 48 GB of LPDDR5X memory shared between CPU and GPU without explicit data transfers. The 192-bit memory bus delivers 273 GB/s theoretical bandwidth to all processing units simultaneously. This design eliminates the PCIe bottleneck that affects discrete GPU systems during encryption operations requiring frequent data movement.

The M4 Pro includes dedicated cryptographic acceleration through ARMv8 Crypto Extensions. These hardware instructions accelerate AES operations directly within the CPU pipeline, achieving throughput levels that exceed general-purpose GPU implementations for single-stream encryption.

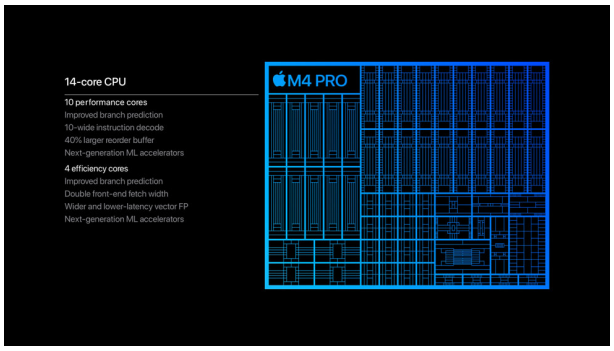

Figure 1: Apple M4 Pro Architecture Layout

Figure 2: NVIDIA RTX 3070 Ampere Architecture

Intel and NVIDIA RTX 3070 Platform

The comparison platform uses an Intel Core i5-8600K processor overclocked to 4.4 GHz across all 6 cores. This 14nm Coffee Lake processor provides 32 GB of DDR4 system memory with approximately 40 GB/s bandwidth to the CPU.

The NVIDIA GeForce RTX 3070 graphics card contains 5888 CUDA cores based on the Ampere architecture. Fabricated on Samsung's 8nm process, the GPU provides 8 GB of GDDR6 memory with 448 GB/s bandwidth through a 256-bit interface. The card operates within a 220-watt power envelope.

Communication between CPU and GPU occurs through a PCIe 4.0 x16 interface providing 32 GB/s bidirectional bandwidth. This creates a fundamental bottleneck for encryption workloads: data must traverse the PCIe bus twice (to GPU for encryption, back to CPU for storage), limiting effective throughput regardless of GPU computational capacity.

Table 1: Test Platform Specifications

Component	**Intel/RTX**	**M4 Pro**
CPU Cores	6	14 (10P+4E)
CPU Frequency	4.4 GHz	4.51 GHz
RAM	32 GB DDR4	48 GB Unified
GPU	RTX 3070	M4 Pro GPU
GPU Memory	8 GB GDDR6	48 GB Shared
Memory Bus	256-bit	192-bit
TDP/TGP	220 W	26 W
Process Node	8nm/14nm	3nm
OS	WSL2	macOS 15.1

METHODOLOGY

Algorithms

We evaluate two symmetric encryption algorithms with distinct computational characteristics. The XOR cipher performs byte-wise exclusive-or operations between plaintext and a repeating key. This algorithm requires minimal computation per byte, making performance entirely dependent on memory bandwidth. XOR serves as an upper bound for achievable encryption throughput on each platform.

AES-256-CTR implements the Advanced Encryption Standard with 256-bit keys in Counter mode. Each 16-byte block requires 14 rounds of SubBytes, ShiftRows, MixColumns, and AddRoundKey transformations. The algorithm demands significant arithmetic operations per byte, classifying it as compute-bound. CTR mode enables parallel processing of independent blocks without inter-block dependencies.

Implementation Strategies

Four implementation strategies span the available parallelism on each platform. Sequential implementations establish baseline performance using single-threaded CPU execution. The macOS implementation uses OpenSSL with hardware-accelerated AES through ARMv8 Crypto Extensions. The Linux implementation uses OpenSSL compiled for x86-64 with AES-NI instructions.

OpenMP implementations distribute encryption across available CPU cores using parallel for-loops with static scheduling. Thread counts scale from 1 to the maximum available (14 for M4 Pro, 6 for i5-8600K) to measure parallel efficiency.

Metal implementations target the Apple GPU using compute shaders written in Metal Shading Language. The unified memory architecture allows zero-copy buffer sharing between CPU and GPU.

CUDA implementations target the RTX 3070 using T-table based AES for coalesced memory access. Data transfers between host and device memory are included in timing measurements to reflect realistic application performance.

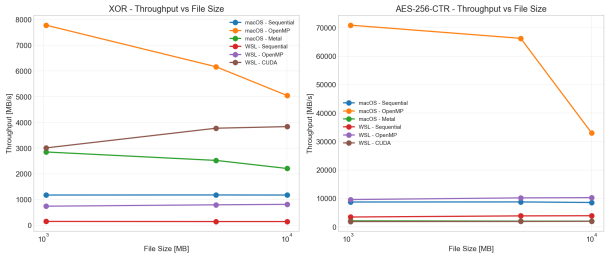

Figure 3: Throughput Comparison Across File Sizes

Measurement Protocol

Each configuration executes three iterations with averaged results. File sizes of 1 GB, 5 GB, and 10 GB stress different aspects of system performance. Verification uses CRC32 checksums comparing decrypted output against original plaintext.

Energy measurements on the Intel/NVIDIA platform use nvidia-smi for GPU power and RAPL counters for CPU power. The M4 Pro estimates power consumption from CPU utilisation and typical Apple Silicon power characteristics. All measurements include complete encryption cycles from plaintext input to ciphertext output.

RESULTS

Sequential Performance

Sequential AES throughput reveals substantial architectural differences between platforms. The M4 Pro achieves 8756 MB/s compared to 3490 MB/s on the Intel i5-8600K, a 2.5x performance advantage. This gap stems from the M4 Pro's dedicated ARMv8 Crypto Extensions, which implement AES transformations in hardware rather than through general-purpose instructions.

Sequential XOR performance shows the opposite pattern. The M4 Pro reaches 1178 MB/s while the Intel platform achieves only 153 MB/s. This 7.7x difference reflects memory subsystem efficiency rather than computational capability. The unified memory architecture of the M4 Pro provides consistently low latency for sequential memory access patterns.

OpenMP Scaling

Thread scaling measurements reveal the parallel efficiency of each platform. The M4 Pro scales XOR throughput from 958 MB/s (1 thread) to 7782 MB/s (14 threads), achieving 6.6x speedup with 47% parallel efficiency at maximum thread count. Efficiency degradation beyond 8 threads indicates memory bandwidth saturation rather than computational limits.

AES scaling on the M4 Pro demonstrates near-linear behaviour through 8 threads, reaching 66 GB/s throughput with 94% efficiency. At 14 threads, throughput peaks at 70.9 GB/s, approaching the theoretical memory bandwidth limit. This indicates that even compute-bound AES

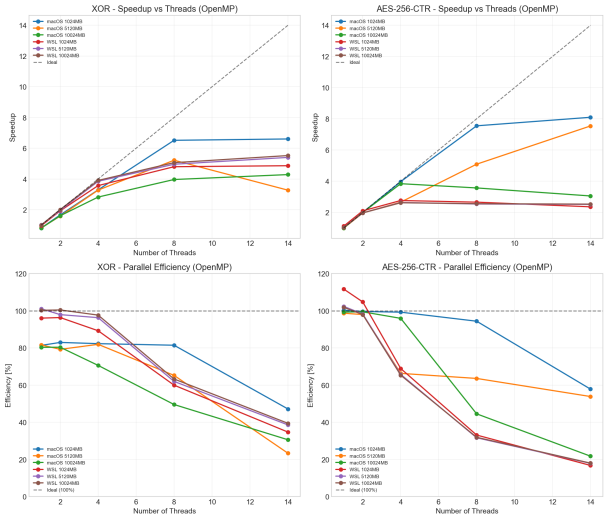

Figure 4: Speedup and Efficiency Relative to Sequential

becomes memory-limited at sufficient parallelism on the unified memory architecture.

The Intel platform shows different scaling characteristics. XOR reaches 815 MB/s at 14 threads (5.5x speedup, 39% efficiency), limited by DDR4 memory bandwidth. AES peaks at 10.2 GB/s with 4 threads before declining, suggesting cache pressure effects at higher thread counts.

GPU Performance

GPU implementations expose the data transfer bottleneck affecting discrete GPU systems. The RTX 3070 achieves 3837 MB/s for XOR on 10 GB files, while Metal on the M4 Pro reaches 2210 MB/s. Despite higher raw GPU throughput, the CUDA implementation must transfer data twice across PCIe, limiting effective performance.

AES results present an unexpected finding. Metal achieves only 2037 MB/s for AES, slower than the 8756 MB/s sequential CPU implementation on the same chip. CUDA reaches 1979 MB/s, similarly underperforming compared to CPU implementations. This counterintuitive result arises from the overhead of GPU kernel launches and memory transfers for workloads where dedicated CPU instructions outperform general-purpose GPU shaders.

The T-table AES implementation on GPU requires multiple memory lookups per round, introducing latency that dedicated CPU crypto instructions avoid. For single-stream encryption, CPU implementations with hardware acceleration consistently outperform GPU approaches.

Execution Time Analysis

Processing 10 GB files demonstrates practical performance differences. The M4 Pro completes XOR encryption in 1.3 seconds using Metal and 8.5 seconds sequentially. OpenMP at 14 threads requires 2.0 seconds, balancing throughput against power consumption.

The Intel/NVIDIA platform requires 67.9 seconds for sequential XOR, 12.3 seconds with OpenMP at 14 threads,

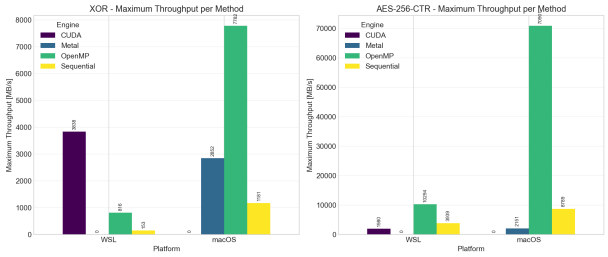

Figure 5: Peak Throughput Per Implementation Method

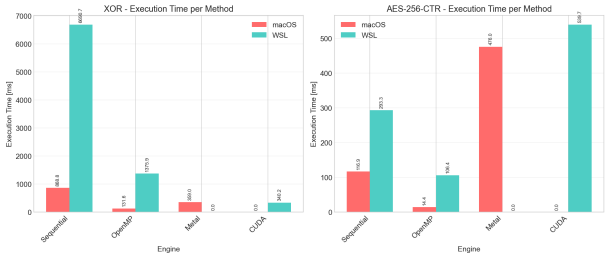

Figure 6: Execution Time for 10 GB File Processing

and 2.6 seconds with CUDA. The discrete GPU provides substantial acceleration over CPU implementations but cannot match the M4 Pro's integrated solution.

AES processing shows smaller absolute time differences due to lower data volumes at equivalent computational effort. The M4 Pro processes 10 GB in 1.17 seconds sequentially, while the Intel platform requires 2.54 seconds. GPU implementations increase processing time on both platforms due to transfer overhead.

Algorithm Comparison

Comparing XOR and AES performance illuminates the distinction between memory-bound and compute-bound workloads. On the M4 Pro, sequential XOR achieves 1178 MB/s while AES reaches 8756 MB/s, indicating that the memory subsystem limits XOR while dedicated crypto hardware accelerates AES.

The Intel platform shows XOR at 153 MB/s and AES at 3490 MB/s sequentially. The 23x performance difference between algorithms (compared to 7.4x on M4 Pro) reflects both memory subsystem limitations and the effectiveness of AES-NI instructions on x86-64.

Energy Efficiency

Power measurements reveal dramatic efficiency differences between platforms. The M4 Pro consumes 13-30 watts during encryption operations, while the RTX 3070 system draws 55-72 watts for GPU operations alone, with total system power exceeding 220 watts under load.

Processing 10 GB with XOR on the M4 Pro requires 85 joules using Metal and 111 joules sequentially. The Intel/NVIDIA platform consumes 1379 joules sequentially and 148 joules with CUDA. Despite lower absolute throughput, the M4 Pro achieves 16x better energy efficiency for XOR encryption.

AES energy consumption follows similar patterns. The M4 Pro uses 31 joules for 10 GB sequential AES, while the Intel platform requires 54 joules. CUDA processing increases Intel platform consumption to 361 joules due to GPU power draw, while Metal on M4 Pro uses 85 joules.

DISCUSSION

The unified memory architecture provides measurable advantages for encryption workloads. Eliminating PCIe transfers removes the primary bottleneck limiting discrete GPU performance for data-intensive operations. The M4 Pro's GPU can access the same physical memory as the CPU without explicit copies, enabling efficient heterogeneous processing.

Dedicated cryptographic hardware in the M4 Pro outperforms GPU-based AES implementations. This result challenges the assumption that GPU parallelism always benefits encryption performance. For single-stream encryption with dedicated CPU instructions, the overhead of GPU kernel management exceeds any parallel processing benefit.

Energy efficiency emerges as a decisive differentiator. The M4 Pro delivers competitive or superior throughput while consuming an order of magnitude less power than the discrete GPU system. For battery-powered devices or data centre deployments with power constraints, this efficiency advantage translates directly to operational benefits.

The 3nm fabrication process enables the M4 Pro's performance-per-watt advantage. Smaller transistors permit higher clock speeds and more processing units within a fixed thermal envelope. The 8nm RTX 3070, while capable, cannot achieve equivalent efficiency with current process technology.

CONCLUSIONS

This study demonstrates that unified memory architectures offer substantial advantages for encryption workloads. The Apple M4 Pro achieves 2.5x higher sequential

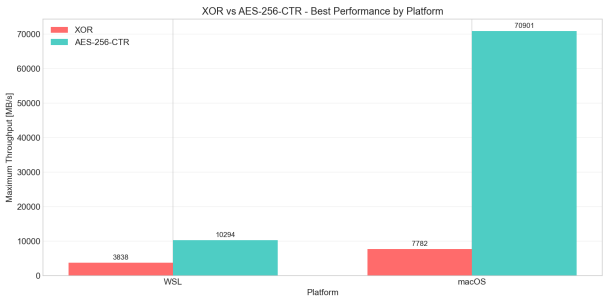

Figure 7: XOR vs AES Performance Comparison

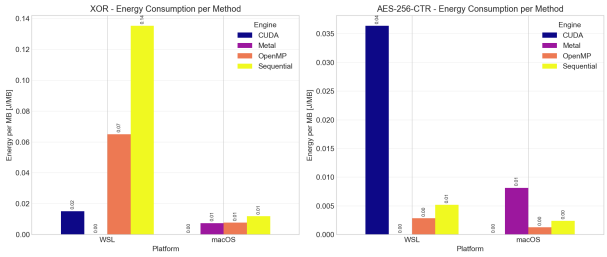

Figure 8: Energy Consumption in Joules

AES throughput than the Intel i5-8600K, 2x higher parallel XOR throughput than CUDA on RTX 3070, and 16x better energy efficiency for XOR operations.

Discrete GPU acceleration provides diminishing returns when PCIe transfer overhead dominates execution time. For encryption tasks requiring frequent data movement between host and device memory, integrated solutions with unified memory prove more effective than raw computational throughput.

Future work should examine encryption of streaming data, where persistent GPU residence eliminates transfer overhead. Additionally, newer discrete GPUs with CXL or similar coherent interconnects may narrow the efficiency gap with unified memory systems.

ACKNOWLEDGEMENTS

This research was conducted at the Cracow University of Technology as part of the High-Performance Computing curriculum. The author thanks the faculty for providing access to computing resources.

References

AMD. AMD Infinity Cache Technology. Technical White Paper, Advanced Micro Devices, 2020. Apple Inc. Apple M4 Pro Chip Architecture. Developer Documentation, Apple Inc., 2024. D.J. Bernstein and P. Schwabe. New AES software speed records. In Progress in Cryptology INDOCRYPT 2008, pages 322-336. Springer, 2008. J.W. Bos, D.A. Osvik, and D. Stefan. Fast implementations of AES on various platforms. IACR Cryptology ePrint Archive, 2009. L. Dagum and R. Menon. OpenMP: an industry standard API for shared-memory programming. IEEE Computational Science and Engineering, 5(1):46-55, 1998. Intel Corporation. Intel Advanced Encryption Standard Instructions (AES-NI). Technical Reference, Intel Corporation, 2010. D.B. Kirk and W.W. Hwu. Programming Massively Parallel Processors. Morgan Kaufmann, 2016. NVIDIA Corporation. CUDA C++ Programming Guide. Developer Documentation, NVIDIA Corporation, 2024. NVIDIA Corporation. NVIDIA Ampere GA102 GPU Architecture. Technical White Paper, NVIDIA Corporation, 2020. D.A. Patterson and J.L. Hennessy. Computer Organization and Design: The Hardware/Software Interface. Morgan Kaufmann, 2017.

AUTHOR BIOGRAPHIES

MACIEJ BIEGAN is a graduate student at the Cracow University of Technology, Department of Computer Science. His research focuses on high-performance computing, parallel algorithms, and heterogeneous system architectures. He is pursuing studies in Data Science with emphasis on computational optimisation and energy-efficient computing.

MATEUSZ NYKO is a graduate student at the Cracow University of Technology, Department of Computer Science. His research interests include parallel computing, GPU programming, and performance optimisation of computational algorithms.

FILIP KRUZEL is a graduate student at the Cracow University of Technology, Department of Computer Science. His work focuses on system architecture analysis, benchmarking methodologies, and energy-efficient computing solutions.