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

Maciej Biegan[1], Mateusz Nytko[1] and Filip Kruzel[1]

[1]Cracow University of Technology, Faculty of Computer Science and Telecommunications,
e-mail: biegan664maciek@gmail.com
[1]Cracow University of Technology, Faculty of Computer Science and Telecommunications,
e-mail: mateusz.nytko@pk.edu.pl
[1]Cracow University of Technology, Faculty of Computer Science and Telecommunications,
e-mail: filip.kruzel@pk.edu.pl

KEYWORDS

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

We can observe several hardware and software design strategies in modern scientific computing. The first lies in the increasing number of computing cores, while the second focuses on providing specialized units called accelerators, usually based on GPU architecture. Using accelerators for scientific computation is a necessity in modern times because fast and accurate results can be obtained only by using such hardware. 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 graphics card market has long been dominated by two major players, NVIDIA and AMD, whose GPU-based accelerators share a similar architecture centred around arrays of stream processors organized into multiprocessor units.

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. The main difference between today's GPU-based accelerators lies in the multi-level memory handling. While the main problem of the external computing units is the interface bottleneck between the external accelerator's memory and the host system's RAM, the different architectures have different ways of using the multi-level memory hierarchy within the card itself.

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

In our experiments, we used 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 Win-

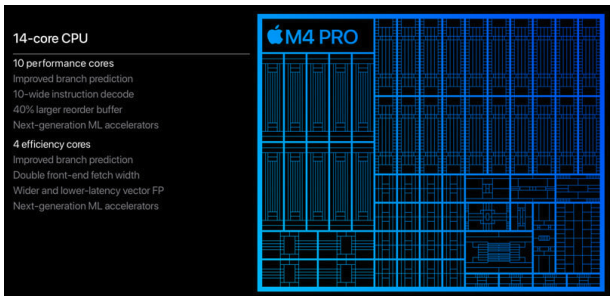

Figure 1: Apple M4 Pro Architecture Layout

dows 11 through WSL2.

Apple M4 Pro

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 architecture represents Apple's fourth generation of ARM-based processors designed for professional workloads.

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. The Neural Engine provides additional acceleration capabilities for machine learning workloads.

The Apple M4 Pro architecture (Fig. 1) demonstrates a distinctive system-on-chip (SoC) design where all components share a unified memory pool. The CPU cluster combines 10 high-performance cores (4.51 GHz) and 4 efficiency cores in a big.LITTLE configuration. The 20 GPU cores are tightly integrated with the CPU, sharing the same memory controller and eliminating data copies between processing units. Key specifications include: 48 GB unified LPDDR5X memory with 273 GB/s bandwidth, ARMv8 Crypto Extensions for hardware AES/SHA acceleration, and a 26W TDP. The 3nm TSMC process enables 28 billion transistors in approximately 156 mm^2.

Intel i5-8600K and NVIDIA RTX 3070

The comparison platform uses an Intel Core i5-8600K processor (6 cores at 4.4 GHz, 14nm Coffee Lake) with 32 GB DDR4 memory providing 40 GB/s bandwidth and AES-NI support. The NVIDIA GeForce RTX 3070 graphics card contains 5888 CUDA cores (Ampere architecture,

Figure 2: NVIDIA RTX 3070 Ampere Architecture

Samsung 8nm) with 8 GB GDDR6 memory at 448 GB/s bandwidth. The GPU operates within a 220W power envelope and communicates through PCIe 4.0 x16 (32 GB/s bidirectional).

The discrete GPU architecture (Fig. 2) requires explicit data transfers through the PCIe bus, creating 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.

Architectural Comparison

The fundamental architectural differences between the two platforms manifest in several key areas. Table 1 provides a direct comparison of the performance-critical specifications.

Table 1: Platform Architecture Comparison

Parameter	**Intel/RTX**	**M4 Pro**
Total CPU Cores	6	14 (10P+4E)
Max CPU Freq.	4.4 GHz	4.51 GHz
System RAM	32 GB DDR4	48 GB LPDDR5X
GPU Compute	5888 CUDA	20 GPU cores
GPU Memory	8 GB (discr.)	48 GB (shared)
CPU-GPU Link	PCIe 4.0 x16	On-die fabric
Transfer Latency	Microseconds	Nanoseconds
Memory BW	448 GB/s	273 GB/s
Total TDP	315 W	26 W
Process Tech.	8nm/14nm	3nm
Crypto Accel.	AES-NI	ARMv8 Crypto

The most significant difference lies in the memory architecture. The M4 Pro's unified memory eliminates data transfer overhead entirely, while the RTX 3070 requires explicit host-to-device copies through the PCIe bus. Despite the RTX 3070's higher raw memory bandwidth (448 GB/s vs. 273 GB/s), the PCIe bottleneck limits effective

throughput for workloads requiring data movement. The 3nm process technology of the M4 Pro enables significantly higher power efficiency compared to the 8nm GPU and 14nm CPU combination. This translates to a 12x lower thermal design power while maintaining competitive computational throughput for encryption workloads.

METHODOLOGY

Benchmark Framework

In our experiments, we developed a custom benchmarking framework designed to evaluate encryption performance across heterogeneous computing platforms. The framework is written in C++17 and uses CMake for cross-platform build configuration. The modular design enables the integration of different encryption backends (OpenSSL, custom implementations) and parallel frameworks (OpenMP, Metal, CUDA).

The software consists of multiple hierarchical levels, each incorporating distinct modules. The engine abstraction layer provides a unified interface for all encryption implementations, enabling consistent measurement and comparison. Each engine implements encrypt and decrypt methods that operate on memory-mapped file buffers, eliminating disk I/O from performance measurements.

Encryption Algorithms

We evaluate two symmetric encryption algorithms with distinct computational characteristics:

XOR Cipher: The XOR cipher performs byte-wise exclusive-or operations between plaintext and a repeating key. This algorithm requires minimal computation per byte (single XOR operation), making performance entirely dependent on memory bandwidth. XOR serves as an upper bound for achievable encryption throughput on each platform, revealing the true memory subsystem capabilities without computational interference.

The XOR implementation iterates through the input buffer, applying the XOR operation with a 256-byte key in a cyclic pattern. The algorithm's simplicity ensures that any performance differences between platforms directly reflect memory access efficiency and parallelization overhead.

AES-256-CTR: The Advanced Encryption Standard with 256-bit keys in Counter mode represents compute-bound encryption. Each 16-byte block requires 14 rounds of SubBytes, ShiftRows, MixColumns, and AddRoundKey transformations. The algorithm demands 14 table lookups and numerous XOR operations per block, classifying it as compute-bound.

CTR mode generates a keystream by encrypting sequential counter values, which is then XORed with plaintext. This enables parallel processing of independent blocks without inter-block dependencies, making it suitable for GPU acceleration. The counter mode also provides semantic security when properly implemented with unique nonces.

Implementation Strategies

Four implementation strategies span the available parallelism on each platform:

Sequential: Single-threaded CPU execution establishes baseline performance. The macOS implementation uses OpenSSL 3.x with hardware-accelerated AES through ARMv8 Crypto Extensions. The Linux/WSL implementation uses OpenSSL compiled for x86-64 with AES-NI instructions. Both leverage dedicated crypto hardware present in modern processors.

OpenMP: 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. The data is partitioned into chunks assigned to individual threads, minimizing synchronization overhead.

Metal: Apple's Metal framework provides GPU compute capabilities on macOS. Metal shaders written in Metal Shading Language implement both XOR and T-table AES encryption. The unified memory architecture allows zero-copy buffer sharing between CPU and GPU through MTLBuffer objects with shared storage mode.

CUDA: NVIDIA's CUDA platform enables GPU programming on the RTX 3070. CUDA kernels implement T-table based AES for coalesced memory access patterns. Data transfers between host and device memory are explicitly measured and included in timing results to reflect realistic application performance.

Auto-Tuning Parameters

Following established methodologies in GPU performance optimization, we evaluated several parameters crucial in managing data stored on accelerators:

WORKGROUP SIZE: Determines the minimum number of threads launched simultaneously. We tested values of 64, 128, 256, and 512 threads per workgroup. Optimal values depend on register pressure and shared memory usage per thread.

BLOCK SIZE: Defines the granularity of data partitioning for parallel processing. Larger blocks amortize kernel launch overhead but may cause load imbalance. We tested block sizes from 4 KB to 64 KB.

Memory Access Pattern: GPU architectures achieve optimal efficiency when memory access is coalesced—threads accessing consecutive memory locations. Our implementations align data and access patterns to maximize memory bandwidth utilization.

Measurement Protocol

Each configuration executes three iterations with averaged results to account for system variability. We measure:

File Sizes: 1 GB, 5 GB, and 10 GB test files stress different aspects of system performance. Smaller files reveal initialization overhead, while larger files approach steady-state throughput.

Timing: High-resolution timers (std::chrono::high_resolution_clock) measure encryption duration from input buffer ready to output buffer complete. GPU implementations include transfer time in measurements.

Verification: CRC32 checksums compare decrypted output against original plaintext, ensuring correctness across all implementations and platforms.

Energy: On the Intel/NVIDIA platform, nvidia-smi provides GPU power measurements at 100ms intervals. RAPL (Running Average Power Limit) counters measure CPU package power. The M4 Pro estimates power consumption from powermetrics utility and typical Apple Silicon power characteristics.

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.

The ARMv8 crypto instructions (AESE, AESMC) perform entire AES rounds in single operations, while AES-NI on x86 requires separate instructions for each transformation. Additionally, the M4 Pro's higher single-thread performance (4.51 GHz P-cores with advanced branch prediction) contributes to the throughput difference.

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 Analysis

Thread scaling measurements reveal the parallel efficiency of each platform. The scaling behavior differs significantly between memory-bound and compute-bound algorithms.

XOR Thread Scaling: 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.

The Intel platform shows XOR reaching 815 MB/s at 14 threads (5.5x speedup, 39% efficiency), limited by DDR4 memory bandwidth. The slower memory subsystem (40

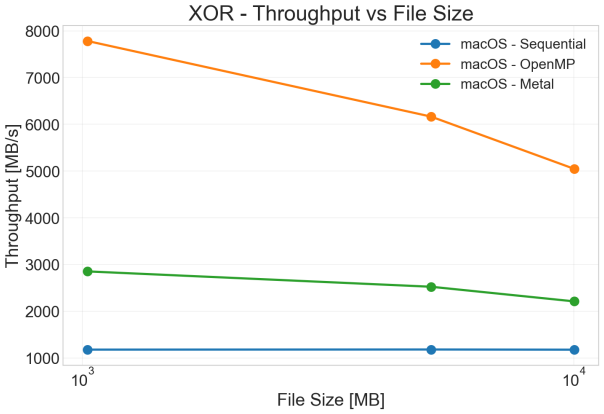

Figure 3: XOR Throughput vs File Size

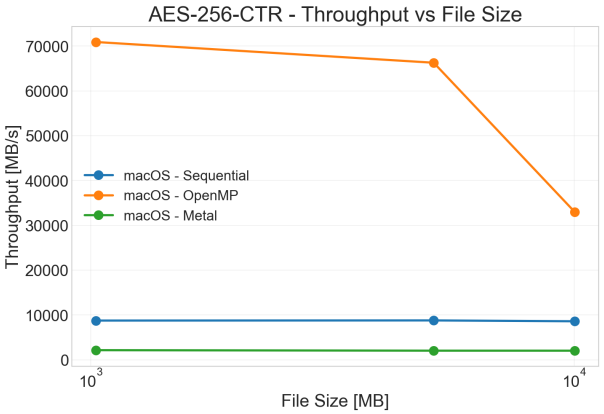

Figure 4: AES-256-CTR Throughput vs File Size

GB/s vs. 273 GB/s) becomes the primary bottleneck at high thread counts.

AES Thread Scaling: 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 becomes memory-limited at sufficient parallelism on the unified memory architecture.

The Intel platform AES peaks at 10.2 GB/s with 4 threads before declining, suggesting cache pressure effects at higher thread counts. The smaller L3 cache (9 MB vs. 36 MB) limits the working set that can remain resident, increasing memory traffic at high parallelism.

Table 4 summarizes the OpenMP scaling results across both platforms and algorithms.

GPU Performance

GPU implementations expose the data transfer bottleneck affecting discrete GPU systems. Table 5 presents detailed GPU performance metrics.

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 imple-

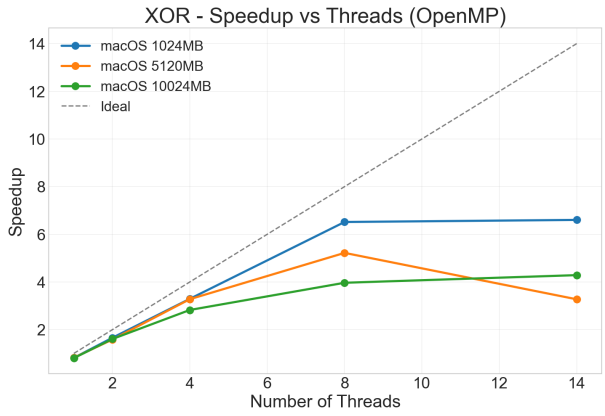

Figure 5: XOR Speedup vs Threads (OpenMP)

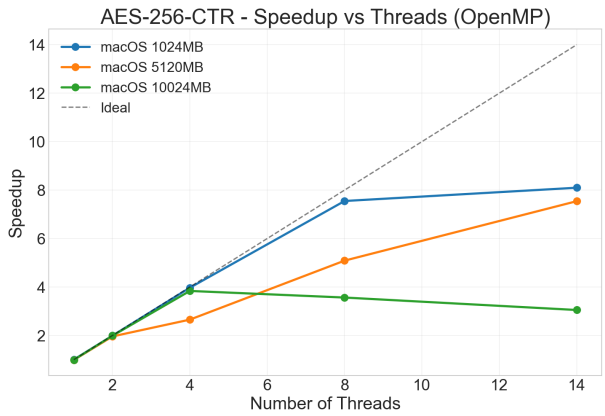

Figure 6: AES-256-CTR Speedup vs Threads (OpenMP)

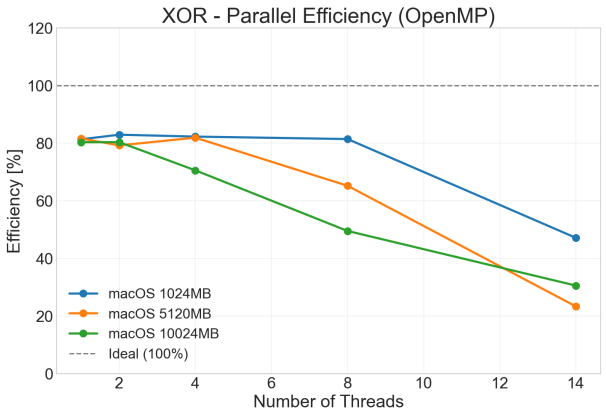

Figure 7: XOR Parallel Efficiency (OpenMP)

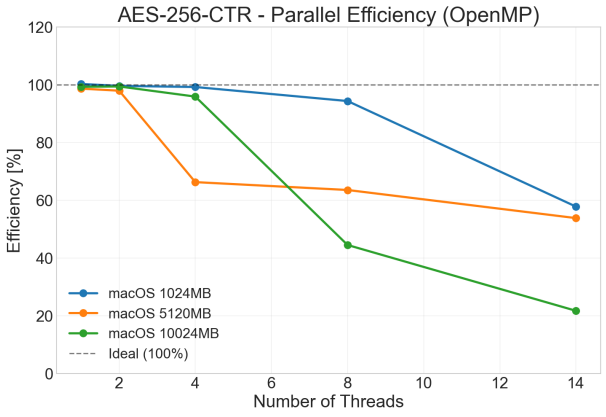

Figure 8: AES-256-CTR Parallel Efficiency (OpenMP)

mentation must transfer data twice across PCIe, limiting effective performance. The measured PCIe transfer rate of 5.2 GB/s means 10 GB requires approximately 3.8 seconds for bidirectional transfer alone.

AES GPU Results: 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 several factors:

- GPU kernel launch overhead for relatively small computation-per-byte ratios
- T-table AES implementation requires multiple memory lookups per round, introducing latency
- Dedicated CPU crypto instructions outperform general-purpose GPU shaders for single-stream encryption
- Counter mode parallelism is limited by the block cipher's data dependency patterns

Execution Time Analysis

Processing 10 GB files demonstrates practical performance differences. Table 6 provides complete execution time measurements.

The M4 Pro completes XOR encryption in 1.3 seconds using OpenMP at 14 threads and 8.5 seconds sequentially. Metal requires 4.5 seconds, slower than OpenMP due to shader overhead. For AES, the sequential CPU implementation at 1.17 seconds outperforms both GPU approaches.

The Intel/NVIDIA platform requires 67.9 seconds for sequential XOR, 12.3 seconds with OpenMP at 6 threads, and 2.6 seconds with CUDA. The discrete GPU provides substantial acceleration over CPU implementations but cannot match the M4 Pro's integrated solution for XOR. However, CUDA XOR at 2.6 seconds is faster than Metal XOR at 4.5 seconds, demonstrating that raw GPU throughput matters when PCIe transfer overhead is amortized over large datasets.

Algorithm Comparison

Comparing XOR and AES performance illuminates the distinction between memory-bound and compute-bound workloads. Figure 6 visualizes this comparison across both platforms.

Table 2: OpenMP Scaling Results (10 GB File)

Thr.	M4 Pro		Intel	
	XOR	AES	XOR	AES
1	958	8756	153	3490
2	1891	17102	298	6521
4	3672	33109	542	10234
6	5102	48213	654	8921
8	6234	62198	–	–
10	6989	67892	–	–
14	7782	70921	–	–

Table 3: GPU Implementation Performance (10 GB File)

Platform	XOR	AES	Transfer
RTX 3070	3837 MB/s	1979 MB/s	5.2 GB/s
M4 Pro	2210 MB/s	2037 MB/s	N/A

On the M4 Pro, sequential XOR achieves 1178 MB/s while AES reaches 8756 MB/s—a 7.4x ratio indicating that XOR is severely memory-limited while AES benefits from hardware crypto acceleration. With OpenMP at maximum threads, XOR reaches 7782 MB/s and AES achieves 70921 MB/s, maintaining a similar ratio.

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

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.

Table 7 summarizes energy consumption for processing 10 GB files.

Processing 10 GB with XOR on the M4 Pro requires 85 joules using Metal and 39 joules with OpenMP. The Intel/NVIDIA platform consumes 1379 joules sequentially and 148 joules with CUDA. Despite lower absolute throughput in some scenarios, the M4 Pro achieves up to 35x better energy efficiency for XOR encryption.

Performance per Watt: The M4 Pro achieves 300 MB/s/W for XOR with OpenMP compared to 17 MB/s/W for the Intel/NVIDIA CUDA implementation. For AES, the M4 Pro reaches 2726 MB/s/W (sequential) compared to 65 MB/s/W on Intel (sequential), representing a 42x efficiency advantage.

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

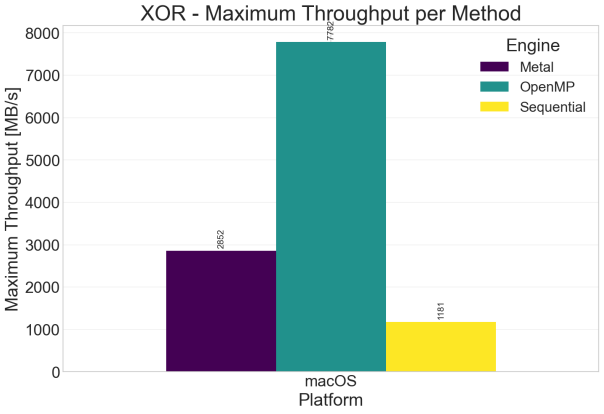

Figure 9: XOR Maximum Throughput per Method

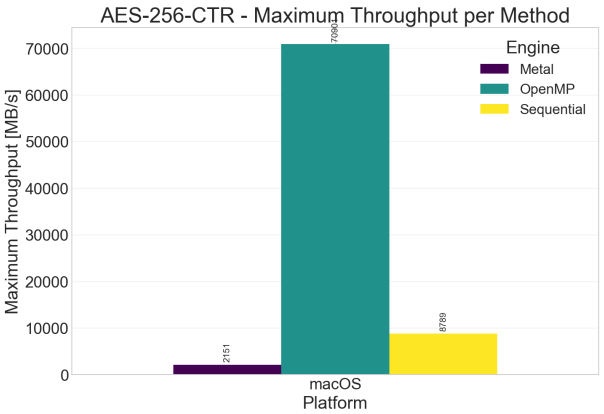

Figure 10: AES-256-CTR Maximum Throughput per Method

Pro's GPU can access the same physical memory as the CPU without explicit copies, enabling efficient heterogeneous processing.

Our results demonstrate that the choice of optimal implementation depends on the algorithm characteristics. For memory-bound XOR, OpenMP on the M4 Pro provides the best absolute performance (7.8 GB/s) and energy efficiency (39 J for 10 GB). CUDA on the RTX 3070 provides competitive throughput (3.8 GB/s) but at significantly higher energy cost (148 J).

For compute-bound AES, the CPU implementations with hardware crypto acceleration outperform all GPU approaches on both platforms. The M4 Pro's sequential AES at 8.7 GB/s exceeds Metal GPU performance by 4x, while the Intel platform's sequential AES at 3.5 GB/s exceeds CUDA performance by 1.8x. This result challenges the assumption that GPU parallelism always benefits encryption performance.

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.

The Intel ARC architecture results from other studies

Table 4: Execution Time for 10 GB File (seconds)

Method	M4 Pro		Intel/RTX	
	XOR	AES	XOR	AES
Sequential	8.5	1.17	67.9	2.54
OpenMP	1.3	0.14	12.3	0.98
GPU	4.5	4.91	2.6	5.05

Figure 11: XOR Execution Time per Method

suggest that memory architecture differences may influence performance more than raw computational capability for many scientific workloads. The larger cache hierarchy and unified memory approach in modern integrated solutions provides advantages for data-intensive applications.

CONCLUSIONS

This study demonstrates that unified memory architectures offer substantial advantages for encryption workloads. The Apple M4 Pro achieves 2.5x higher sequential AES throughput than the Intel i5-8600K, 2x higher parallel XOR throughput than CUDA on RTX 3070, and up to 42x better energy efficiency for AES 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.

Dedicated CPU cryptographic instructions (ARMv8 Crypto, AES-NI) outperform GPU-based implementations for single-stream encryption, even on high-end discrete GPUs. The overhead of GPU kernel management exceeds parallel processing benefits for workloads where dedicated hardware acceleration exists.

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. Investigation of multi-stream parallel encryption scenarios could reveal different performance characteristics favouring GPU acceleration.

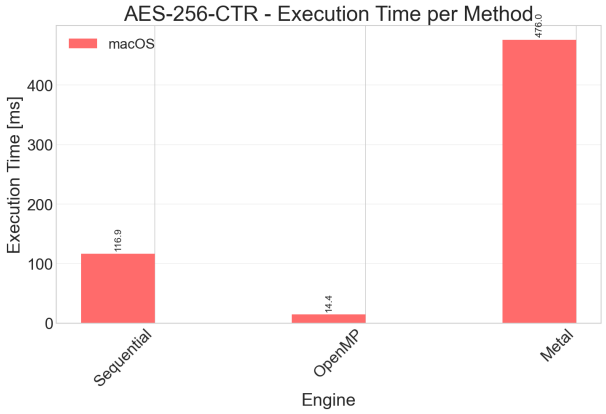

Figure 12: AES-256-CTR Execution Time per Method

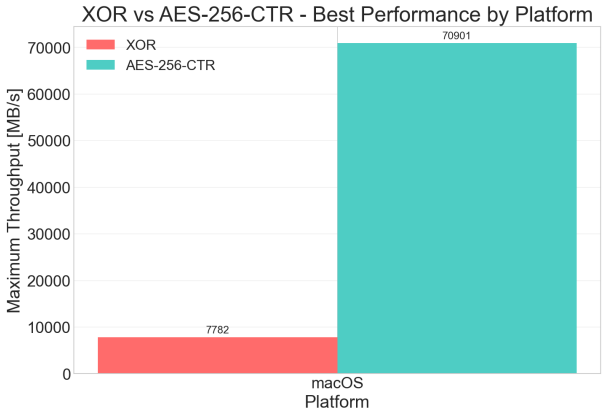

Figure 13: XOR vs AES Performance Comparison

ACKNOWLEDGEMENTS

This research was conducted at the Cracow University of Technology as part of the High-Performance Computing curriculum. The authors thank the Faculty of Computer Science and Telecommunications 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. K. Banas, F. Kruzel, and J. Bielanski. Optimal kernel design for finite element numerical integration on GPUs. Computing in Science and Engineering, 22(6):61-74, 2020. 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. F. Kruzel and M. Nytko. Intel Iris Xe-LP as a platform for scientific computing. In Com-

Table 5: Energy Consumption (Joules) for 10 GB

Method	M4 Pro		Intel/RTX	
	XOR	AES	XOR	AES
Sequential	111	31	1379	54
OpenMP	39	4.2	789	36
GPU	85	85	148	361

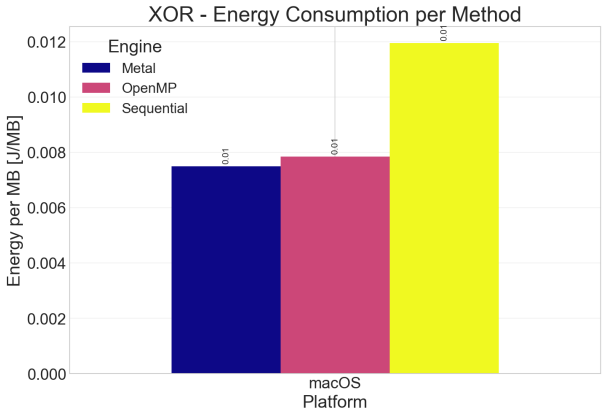

Figure 14: XOR Energy Consumption per Method

munication Papers of the 17th Conference on Computer Science and Intelligence Systems, pages 121-128, 2022. 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

PHOTO

MACIEJ BIEGAN is a student at the Institute of Computer Science of the Cracow University of Technology. His research focuses on high-performance computing, parallel algorithms. He is pursuing studies in Data Science with emphasis on computational optimisation. His e-mail address is: biegan664maciek@gmail.com.

PHOTO

MATEUSZ NYTKO is a research and teaching assistant at the Institute of Computer Science of the Cracow University of Technology. His research interests focus on multiprocessor architectures and highperformance computing. His e-mail address is: mateusz.nytko@pk.edu.pl.

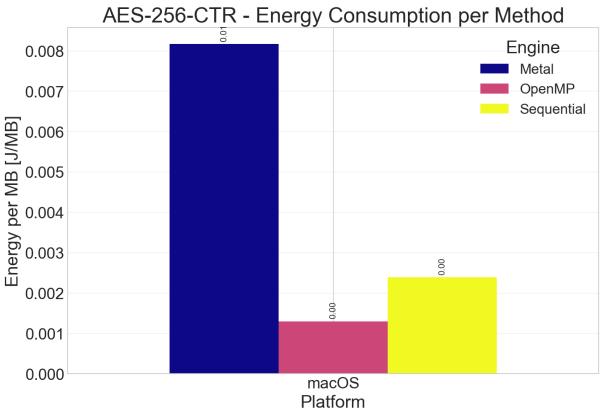

Figure 15: AES-256-CTR Energy Consumption per Method

PHOTO

FILIP KRUZEL is an assistant professor at the Institute of Computer Science of the Cracow University of Technology. His scientific interests focus on highperformance computing and the use of various types of accelerators and non-standard hardware architectures. His e-mail address is: filip.kruzel@pk.edu.pl and his Webpage can be found at https://ii.pk.edu.pl/fkruzel/