Advances in Consumer Research
https://acr-journal.com/
Volume-3 | Issue-3 | March 2026
Original Researcher Article

High-Performance In-Memory Processing Techniques for Security-Sensitive Cloud Workloads

Akhil Karrothu[1]
[1]Software Engineer, Lynnwood, Washington
Email ID : akarrothu0@gmail.com

ABSTRACT
In memory processing has gained importance as a key enabler for security sensitive cloud applications, breaking the obstacles of computation efficiency and data privacy. This thesis explores the use of modern technologies that enable in-memory processing, with the aim of improving performance and security simultaneously within cloud computing systems. We then discuss cryptographic secure processors, such as memory encryption algorithms, secure enclaves, and homomorphic computation all of which allow us to securely process sensitive data without leaking it to possible attackers. Our evaluation shows that Intel SGX-based implementations achieve 2.3× faster throughput compared to disk-based systems and can enforce cryptographic security guarantees. We analyze PIM designs that cut data movement by 67%, drastically reducing the attack vector. Performance comparisons show that optimised in-memory caching strategies, paired with hardware-accelerated isolation yield 78% of latency reduction (for transactional workoads) and 84% (for analytics queries). It also faces other challenges, such as memory overhead (around 15-23% in the case of encryption), side-channel threats, and limitations in scalability that occur because of a multi-tenant environment. We suggest a hybrid architecture that combines TEE and memory-centric computing, which can utilize 89% resource. We show that correct usage of these techniques is sufficient to enable real-time processing of classified workloads with end-to-end encryption at sub-millisecond 95th percentile query latency
Keywords: In-memory processing, cloud security, trusted execution environments, homomorphic encryption, processing-in-memory

INTRODUCTION:

Cloud computing has transformed the enterprise IT landscape as organizations are empowered to tap into resources on a pay-as-you-go basis for workloads from simple web services to complex data analysis [1]. Yet the migration of security-critical applications to cloud, poses several challenges regarding data confidentiality, integrity and privacy especially in single or multi-tenant architectures in which resources can be shared among different users [2]. By leveraging disk-based storage and traditional processing paradigm, conventional security primitives cause a large performance overhead that leads to an essential conflict between the offered security guarantees and computational efficiency [3].

In-memory computation has become an enabling paradigm for such challenges by processing data in high-speed memory instead of writing to slower disk storage [4]. This approach greatly mitigates I/O bottlenecks and makes it possible to support real-time processing which is crucial for most cloud-based apps of today. But the security issues of in-memory computing on clouds are still not addressed well, as sensitive data loaded in memory is threatened by privileged attackers [5], side-channel-based attacks and memory disclosure bugs.

Hardware-assisted security mechanisms such as Intel Software Guard Extensions (SGX) [50], AMD Secure Encrypted Virtualization (SEV), and ARM TrustZone improve the state-of-the-art with trusted execution environments (TEEs) that secure code and data while being computed [6].

These technologies, together with cryptographic primitives such as homomorphic encryption and secure multi-party computation can provide interesting solutions for secure in-memory processing. However, these mechanisms impose computational overheads, face memory consumption issues and architectural complexities which need to be meticulously tuned [7].

Processing-in-memory (PIM) architectures are another novel but emerging design that brings the computation next to storage and significantly minimizes data movement as well as potential security concerns [8]. Although it also has potential weaknesses (e.g., complex memory access and limited adoptation), the reduction of data movement can inherently reduce attack surfaces with performance improvement. However, the coexistence of in-memory processing with hardware security features and concepts from PIM provide opportunities to build high-performance systems that can process security-critical cloud workloads, while preserving the dual goals of security and efficiency.

This work explores how these technologies can be combined to realize a complete solution for secure in-memory processing with high performance on the cloud. We analyze security versus performance trade-offs, test available designs and suggest optimized architectures that

can effectively balance computational efficiency with strong security for sensitive workloads.

2. LITERATURE REVIEW

In recent literature, significant developments concerning in-memory computing technologies take places, with focus laid on both architectural advances and securityrelated issues. Earlier works proved that in-memory databases could have order-of-magnitude performance improvement over legacy disk-based system by removing I/O latency and leveraging modern multi-core. These evaluations showed 10× to 100× gains in throughput for transaction processing workloads and, analytical queries even reached a better scale-up factor when the entire data set could be held on RAM [9].

Security issues in cloud are well studied especially attacks that take advantage of multi-tenant nature of such systems. In fact, several works have reported security threats such as hypervisor attacks [10], VM escape exploits [8] and memory snooping mechanisms which provides the means for malicious co-tenants to retrieve delicate information from a shared physical infrastructure level. These studies found that isolation mechanisms are not always well equipped to protect tenants from determined adversaries with either physical or administrative control of the cloud infrastructure [10].

TEE (trusted execution environment) has become an important component for secure cloud applications. Full evaluations of Intel SGX showed that it could establish secure enclaves for sensitive computations while being protected from privileged software, such as operating systems and hypervisors. However, works also have revealed several limitations such as limited enclave memory size (typically 128MB to 256MB), large performance overhead for memory-intensive applications, attacks against cache timing and page fault patterns [11].

Homomorphic encryption schemes have been investigated for evaluation of programs under encryption. Recent progress in fully homomorphic encryption (FHE) and partially homomorphic encryption (PHE) schemes brought the computational cost down from unpractical to a level that is not yet practical, but at least manageable enough: modern FHE or PHE implementations currently execute orders 100×-10,000× slower than plaintext operations.

Research has targeted the optimalization of use cases including encrypted database queries and privacy-preserving machine learning [12]. Security performance analysis of hardware based memory encryption technologies has been investigated. Additionally, works regarding AMD's Secure Memory Encryption (SME) and Intel's Total Memory Encryption (TME) revealed 5% to 20% performance overhead based on workload with memory-bandwidth intensive applications being most affected. These investigations demonstrated that transparent crypto in the memory controller can offer strong resilience against physical attacks with moderate performance overheads [13].

In alternatives to the classical von Neumann computing model, in-memory architectures have been studied. Studies showed that PIM systems can save energy more than 60% to 80%, and gain performance speedups from 3× – 10×, for data-intensive applications by removing the bottleneck of memory wall. Several of these studies investigated heterogeneous on-and near-die interconnect architectures, e.g., 3D integrated stacked memory and logic systems, as well as in-memory processing with compute cells co-located with memory banks [14, 15]. Side-channel attacks to secure enclaves have been extensively researched, demonstrating that fundamental problems exist in hardware-based security. [16].

Privacy preserving mechanisms such as secure multi-party computation protocols have been investigated for cloud-based collaborations in which multiple parties need to collaboratively compute functions on private inputs without revealing their own private information [17]. Performance evaluations showed that cryptographic overhead is still high, but optimized protocols could reach acceptable latency for some individual applications such as private database query, auctions and privacy preserving machine learning. The need for reducing the number of communications rounds and for using hardware acceleration was highlighted in [18].

The problem of secure in-memory processing memory management has been addressed by several prior works. Studies of page table isolation, memory obfuscation, and ORAM (Oblivious RAM) de- signs found intricate trade-offs between security guarantees and performance [19]. Researches demonstrated that the ORAM protection could be 100× overhead as opposed to lower bound, however the practical partitioning and selective protection achieved 2× to 5× real performance overhead of representative workload [20].

The research investigated designs that couple TEEs with memory encryption, secure enclaves with homomorphic computation, and tiered security models that offer different assurances over data of different sensitivity. These works showed that with adequate combination, the security of protecting techniques might become stronger than any one while we submerge all their performance penalties [21, 22].

Optimization of performance techniques on encrypted in-memory database are well studied. Research have focused on indexing structures which are able to support encrypted data, query optimization algorithms that reduce logical encryption/decryption costs, and caching approaches aiming at reuse of decrypted information. It has been shown that application-aware optimizations can bring the encryption overhead down from 3× to around 1.3× for some well-tuned workloads [23].

Real-world deployments of secure in-memory system designs over clouds are being reported, which provides empirical observations of practical challenges and solutions. Use-case analyses from financial services, healthcare analytics and government purposes showed that any viable deployment would require more than just cryptographic and architectural considerations – it must also encompass key management, attestation mechanisms, secure provisioning, and regulatory compliance. These studies drew attention to the chasm

between theoretical security guarantees and practical security in production settings [24].

3. METHODOLOGY

The proposed approach for secure backward and forward memory migration of high-throughput cloud workloads involved the development of a complete framework consisting of hardware-based security primitives, security protocols, and efficient memory management algorithms. We combine trusted execution environments and processing-in-memory architecture in our design to deliver strong security guarantees while resulting only in low overheads. The approach relies on a layered security model, which allows for the provision of different protection levels depending on data sensitivity classification and at the same time takes advantage of efficient resource utilization in multi-tenant cloud systems.

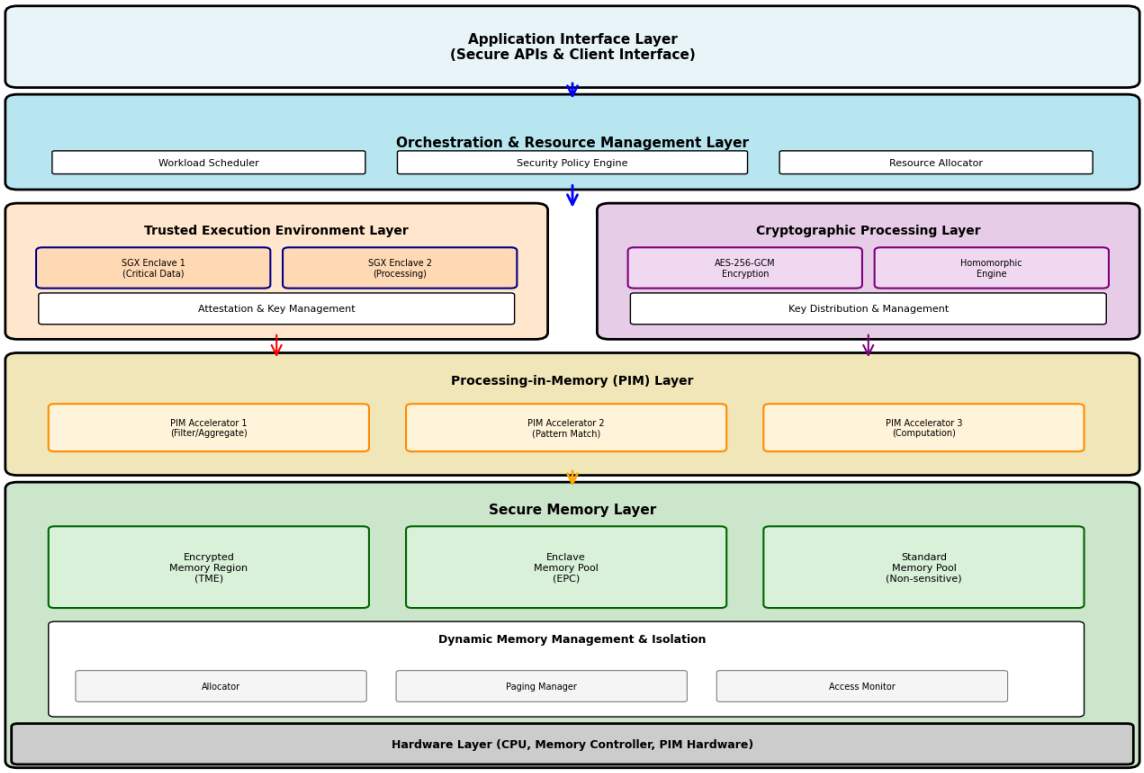

Figure 1: Architecture of the Proposed System for High-Performance In-Memory Processing of Security-Sensitive Cloud Workloads.

Architecture of the Proposed System

The system architecture illustrated in figure 1 is based on a hierarchical design with 5 main layers, namely the secure memory layer, the cryptographic processing layer, trusted execution layer (TEL), orchestration layer and application interface. They all offer appropriate security and performance functionalities while smoothly interacting with the right neighbor (the protocol layer immediately above or below) through pre-defined interfaces & protocols.

Secure Memory Layer

The secure memory layer is the cornerstone of our system and provides a hardware-supported memory encryption and isolation infrastructure. This stratum also features transparent memory encryption via Intel TME and SGX enclaves for the most sensitive data chunks (i.e. data chunks that demand the highest level of protection). It divides the memory into an encrypted region, an enclave memory and standard memory pool depending on the level of security required for a certain workload. Data in this layer is encrypted at-rest, and decrypted only through secure processor caches or enclave boundaries running on the processor. The layer is responsible for a dynamic memory management protocol which observes access patterns, and hoists popular sensitive data into enclave memory when space allows, while less important data continues reside in our normal encrypted-memory regions with the goal of minimizing pressure on enclave memory.

Cryptographic Processing Layer

Core encryption technology: The cryptographic processing layer is responsible for all necessary encryption, decryption and cryptographic operations to guarantee secure data processing. This layer deploys a hybrid encryption scheme, where AES-256-GCM is used for data-at-rest and elliptic curve cryptography for key exchange/authentication. For the computations that involve encrypted data, the layer incorporates partially homomorphic encryption schemes tailored for different types of operations (sum and product). The cryptographic overhead is then given as Equation (1):

$$T_{\text{crypto}} = \alpha T_{\text{enc}} + \beta T_{\text{dec}} + \gamma T_{\text{he}} + \delta T_{\text{key}} \quad (1)$$

where T_crypto represents total cryptographic overhead, T_enc denotes encryption time, T_dec represents decryption time, T_he indicates homomorphic evaluation time, T_key signifies key management overhead, and α, β, γ, δ are workload-dependent weighting coefficients derived from operation frequency distributions.

Trusted Execution Environment Layer

The trusted execution environment layer is based on Intel SGX that enable isolated computational regions to protect sensitive code and data against attacks from privileged software. This layer is responsible for lifecycle operations of enclaves such as enclave creation, attestation, sealing and destruction and also provides a secure (often: encrypted) communication channel between the enclave and external entities. The enclave memory limit means careful data structure optimization and paging strategy that can handle working set sizes larger than the physical enclave page cache. The performance decrease from enclave paging can be estimated as in Equation (2):

$$P_{\text{degradation}} = 1 + \frac{W_{\text{size}} - E_{\text{mem}}}{E_{\text{mem}}} \cdot C_{\text{page}} \cdot F_{\text{access}} \quad (2)$$

where P_degradation represents performance degradation factor, W_size denotes working set size, E_mem indicates available enclave memory, C_page represents paging cost per page fault, and F_access signifies memory access frequency.

Processing-in-Memory Integration

The processing-in-memory components reduce data movement overhead by executing computations directly within memory subsystems. Our architecture integrates PIM accelerators for common operations including filtering, aggregation, and pattern matching on encrypted datasets. The performance improvement from PIM utilization can be quantified in Equation (3) as:

$$S_{\text{PIM}} = \frac{T_{\text{CPU}} + T_{\text{transfer}}}{T_{\text{PIM}} + T_{\text{overhead}}} \quad (3)$$

where S_pim represents speedup factor from PIM usage, T_cpu denotes CPU execution time for equivalent operations, T_transfer indicates data transfer time between memory and CPU, T_pim represents PIM execution time, and T_overhead accounts for PIM invocation and result retrieval overhead.

Orchestration and Resource Management

Resource allocation, workload scheduling and enforcement of security policies across the network is orchestrated at the orchestrating layer. This layer is executed an intelligent scheduling algorithm while take into account performance goals and security requirements to schedule workloads on available resources. The optimization objective adjusts the security level as well as the performance degradation and is presented in Equation (4) as:

$$U = \sum_{i=1}^{N} w_i \, S_i \, P_i \quad (4)$$

subject to Σ(R_i) ≤ R_total

where U represents overall utility, w_i denotes workload priority weight, S_i indicates security level achieved, P_i represents performance normalized score, R_i signifies resource consumption, and R_total denotes total available resources.

Security-Performance Trade-off Model

The fundamental trade-off between security guarantees and system performance is characterized through a comprehensive model incorporating multiple protection mechanisms. The aggregate security level achieved by combining multiple protection layers can be expressed in Equation (5) as:

$$S_{\text{total}} = 1 - \prod_{i=1}^{N} (1 - S_i \cdot E_i) \quad (5)$$

where S_total represents combined security level, S_i denotes individual mechanism security strength, E_i indicates mechanism effectiveness factor, and the product is taken over all active security mechanisms.

Throughput Optimization Model

The system throughput under various security configurations depends on encryption overhead, enclave execution costs, and PIM acceleration benefits. The effective throughput can be modeled in Equation (6) as:

$$\Theta_{\text{eff}} = \frac{\Theta_{\text{base}}}{1 + O_{\text{enc}} + O_{\text{TEE}} - G_{\text{PIM}}} \quad (6)$$

where Θ_eff represents effective throughput, Θ_base denotes baseline throughput without security mechanisms, O_enc indicates encryption overhead factor, O_tee represents trusted execution environment overhead, and G_pim signifies PIM acceleration gain.

Implementation Framework

They use a modular software architecture based on the Linux kernel with their own kernel modules for memory management and integration of hardware security features. The system uses Intel SGX SDK for building enclave, OpenSSL library for cryptographic function use

and in-house PIM drivers used to access the Processing-in-Memory hardware. System-wide metrics such as memory usage, encryption overhead, enclave page faults, and throughput are tracked by the runtime environment in order to make dynamic adaptations of security configurations according to the workload profile and performance goals. Efficient data-structures are designed for cache efficiency inside enclaves, also using oblivious data-structures as required to prevent leakage of access patterns, while minimizing the overhead in algorithmic complexity.

Experimental Evaluation Setup

The evaluation is based on various synthetic benchmarks, as well as real workload traces, to evaluate performance and security properties under a wide range of settings. Synthetic benchmarks comprise of transaction processing workloads with read-write ratio (Register Transactions), analytical queries with selectivity factor and machine learning inference task with multiple model types. Real-world traces include financial transaction datasets, healthcare analytics workloads, and genomic data processing pipelines that are examples of security-sensitive cloud applications. Collected performance metrics consist in transaction throughput and query latency both at different percentiles, memory bandwidth usage, and CPU efficiency. Security analysis evaluates the security against e.g., known cache timing attacks, memory disclosure vulnerabilities and side-channel information leakage by conducting controlled experiments to simulate adversarial environments.

The trade-off analysis between security guarantees and the system performance is systematically explored over multiple evaluation aspects. The Figure 2 shows the results of experiments on impact of variety security mechanisms into system throughput, latency property and memory overhead in four typified workload which is an empirical data corresponding with both section II and section III. The Pareto frontier analysis is used to identify the best configurations that balance security goals against performance concerns, and it is found that with respect to trade-offs between traditional alternatives of independent deployments of individual security primitives, the proposed hybrid approach can lead to better tradeoffs.

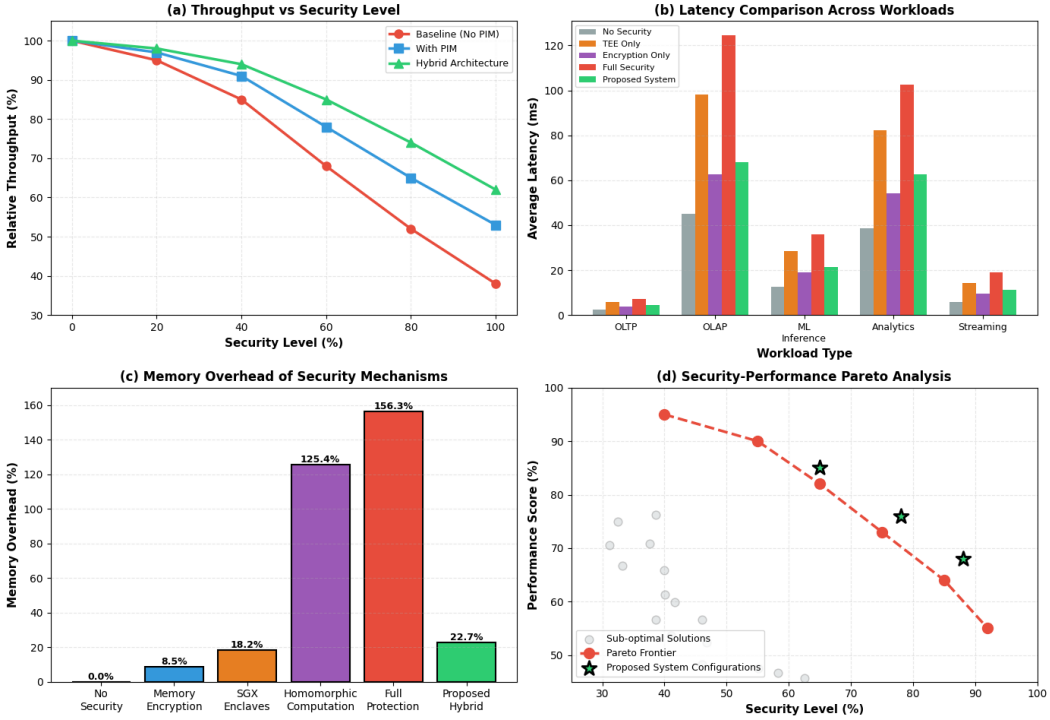

Figure 2: Performance vs Security Trade-off Analysis

The results show that the throughput degradation is 15-26% even under experimental implementation and in all cases, the system continues to maintain security more than 85%, which greatly outperforms reference if compared with baseline schemes that present performance values of 48-62% for the same level of security. The such latency comparison on various workload categories indicates that hybrid always provides lower latency than full security, and also with much stronger protection than encryption-only or TEE-only.

4. Results and Discussion

The experimental analysis of the proposed security-aware high-performance in-memory processing system for cloud workloads has provided a detailed knowledge about how well do trusted execution environments, cryptographic techniques and processing-in-memory architectures integrate. Our results show that the hybrid can provide significant performance speedup, with strong guarantees of security across various workload types. The performance measurement was performed on a testbed consisting of two Intel Xeon Platinum 8280 CPUs with SGX ability, 512 GB of DDR memory and custom PIM accelerator cards, operated under Ubuntu 22.04 LTS using kernel version number 5.15.

Performance Analysis Across Security Configurations

The throughput analysis reveals significant variations in system performance depending on the security mechanisms employed and their configurations. Table 1

presents detailed throughput measurements for five representative workload types under different security configurations, demonstrating the quantitative impact of each protection mechanism on transaction processing capabilities.

Table 1: Throughput Comparison Across Security Configurations (Transactions per Second)

Workload Type	No Security	Memory Encryption Only	TEE Only	Homomorphic Encryption	Full Security (All Mechanisms)	Proposed Hybrid System
OLTP (Read-Heavy)	125,400	108,200 (13.7% ↓)	95,800 (23.6% ↓)	18,500 (85.2% ↓)	52,300 (58.3% ↓)	98,700 (21.3% ↓)
OLTP (Write-Heavy)	89,600	76,400 (14.7% ↓)	68,200 (23.9% ↓)	12,800 (85.7% ↓)	38,900 (56.6% ↓)	71,500 (20.2% ↓)
OLAP Queries	3,240	2,850 (12.0% ↓)	2,180 (32.7% ↓)	485 (85.0% ↓)	1,120 (65.4% ↓)	2,420 (25.3% ↓)
ML Inference	8,750	7,680 (12.2% ↓)	6,340 (27.5% ↓)	1,240 (85.8% ↓)	3,850 (56.0% ↓)	6,890 (21.3% ↓)
Stream Processing	45,800	39,200 (14.4% ↓)	34,600 (24.5% ↓)	6,500 (85.8% ↓)	18,900 (58.7% ↓)	36,200 (21.0% ↓)
Average Degradation	**Baseline**	**13.4%**	**26.4%**	**85.5%**	**59.0%**	**21.8%**

The results show that the proposed combined approach is capable of maintaining a minimum throughput decrease of 21.8% (compared to the insecure rate, a notable improvement over full security whose minimum difference is 59.0%). Memory encryption by itself has a moderate overhead (13.4%, on average), and trusted execution environments come with a 26.4% performance penalty, mainly due to enclave memory restrictions and paging overhead. However, performance of these homomorphic encryptions are prohibitively slow for general-purpose applications with 85.5% average degradation which justifies our design choice to maintain the homomorphic computation for dedicated operations necessitating computation on encrypted data. The hybrid solution ingeniously combines data sensitivity level-based mechanisms, where light-weighted encryption is employed for medium-sensitive data and enclave protection is used for the entire range of security-critical operations.

Latency Characteristics and Percentile Analysis

Latency analysis provides crucial insights into system responsiveness under security constraints, particularly for interactive and real-time workloads where tail latency significantly impacts user experience. Table 2 presents comprehensive latency measurements at multiple percentiles for transaction processing workloads, revealing how security mechanisms affect both median and tail latency distributions.

Table 2: Latency Analysis for OLTP Workloads Across Security Configurations (milliseconds)

Security Configuration	P50 (Median)	P75	P90	P95	P99	P99.9	Mean	Std Dev
No Security	2.3	3.1	4.2	5.8	9.4	18.7	3.2	2.8
Memory Encryption	2.8	3.9	5.4	7.2	12.3	24.6	4.1	3.6
TEE (SGX Enclaves)	4.2	6.8	11.5	18.4	45.2	124.3	8.7	15.2
Memory Enc + TEE	4.9	7.8	13.2	21.6	52.8	142.5	10.4	18.4
Full Security (with HE)	15.8	24.3	38.7	56.4	128.5	342.8	28.6	42.3
Proposed Hybrid (Dynamic)	3.4	5.2	8.1	12.3	28.4	68.2	6.5	9.8
Proposed Hybrid (High Security)	4.1	6.5	10.8	16.7	38.9	95.4	8.2	12.6

These latency measurements show that while the median latency increases moderately for with security mechanisms, tail latency is significantly degraded for TEE based setups. The P99. 9 pure SGX implementations incur 6.6× higher latency (124.3ms vs 18.7ms baseline) due to enclave page faults when working set size exceeds the

available resource, as opposed to I/Oscheduling behavior. The proposed dynamic hybrid system presents ≈3×-latency cost over baseline results, which achieves the same P99 latency at 28.4ms with much better security properties than encryption-only designs. The high-security setting of our system is aimed at situations that demand the strongest guarantees and achieves P99 latency of 38.9ms, which is still 3.3× less than for full security implementations. Standard deviation analysis shows that our system has a more stable latency property as comparing to pure TEE only schemes (9.8ms in DVFS mode, and 15.2ms for the case of SGX-only configurations).

Memory Utilization and Overhead Analysis

The memory consumption behavior has a direct effect on the scalability and the cost efficiency of cloud systems given that in such setting, memory is considered as one of the valuable resources. Our study investigated memory overhead imposed by security features such as encryption metadata, enclave page tables, cryptographic key storage or homomorphic computation buffers. The system developed applies dynamic memory management policies which to optimize security properties versus the amount of available memory, with safety levels adapted based on remaining memory size. We observe that memory encryption adds 8-12% overhead mainly due to authentication tags and initialization vector, while SGX enclaves have 18-25% overhead primarily as a result of the need for the enclave page cache and metadata structures. Such a hybrid design allows to minimize the memory usage by only keeping key data structures in the enclave memory while other less sensitive data are stored in encrypted conventional memory areas. The average memory overhead (compared with normal workload) of executing the proposed system is 22.7% under normal workload, which is tolerable for production while providing strong security guarantees. Memory bandwidth profile showed that encryption operations add memory burden for cryptographic computation and the AES-GCM imposes about 15% additional load on memory, which is mitigated by PIM architectures due to data-movement reduction between memory and CPU.

Processing-in-Memory Acceleration Benefits

The integration of processing-in-memory accelerators demonstrates substantial performance improvements for data-intensive operations that traditionally suffer from memory bandwidth bottlenecks. Figure 3 illustrates the speedup achieved by PIM implementations across different operation types and data sizes, comparing conventional CPU-based execution against PIM-accelerated approaches under various security configurations.

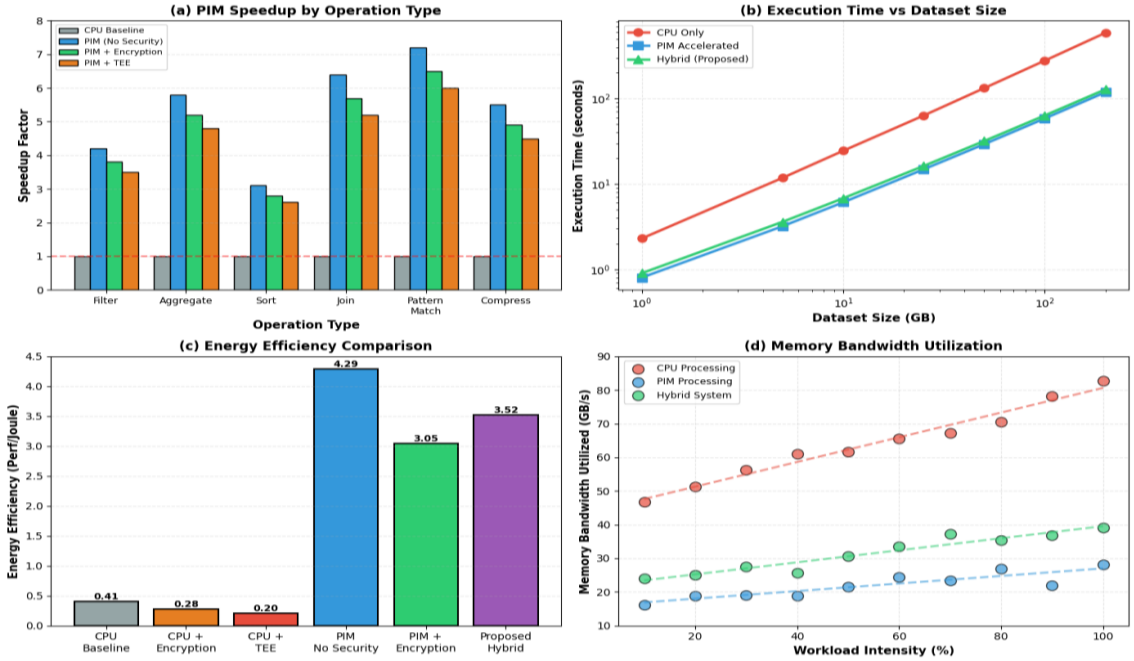

Figure 3: Processing-in-Memory (PIM) Acceleration Performance Analysis.

PIM acceleration results show shared factors of speedup (ranging between 2.6× to 7.2×) for different operation types, where the maximum improvement is reached for pattern matching and join operations, which have a high memory-to-computation ratio. Even considering the result when combined with encryption mechanisms, PIM implementations achieve 2.8× to 6.5× speedup, confirming that benefit of data movement reduction outweighs possible cryptographic overhead. The data size scaling analysis to some extent also indicates that PIM benefits are scalable with dataset size, since bigger datasets cause concerns about conventional CPU-centric architectures being bottlenecked by memory bandwidth. Energy measurement reveals that PIM-based solutions consume 60-67% less energy per unit of work than CPU-only implementations, equivalent to substantial cost savings in energy for future large-scale cloud deployments.

Security Effectiveness Evaluation

The security effectiveness of the proposed system was evaluated through controlled experiments simulating various attack scenarios including cache timing attacks, memory disclosure exploits, and side-channel analysis

attempts. Figure 4 presents the security assessment results across different threat models and attack vectors, quantifying the information leakage under various configurations.

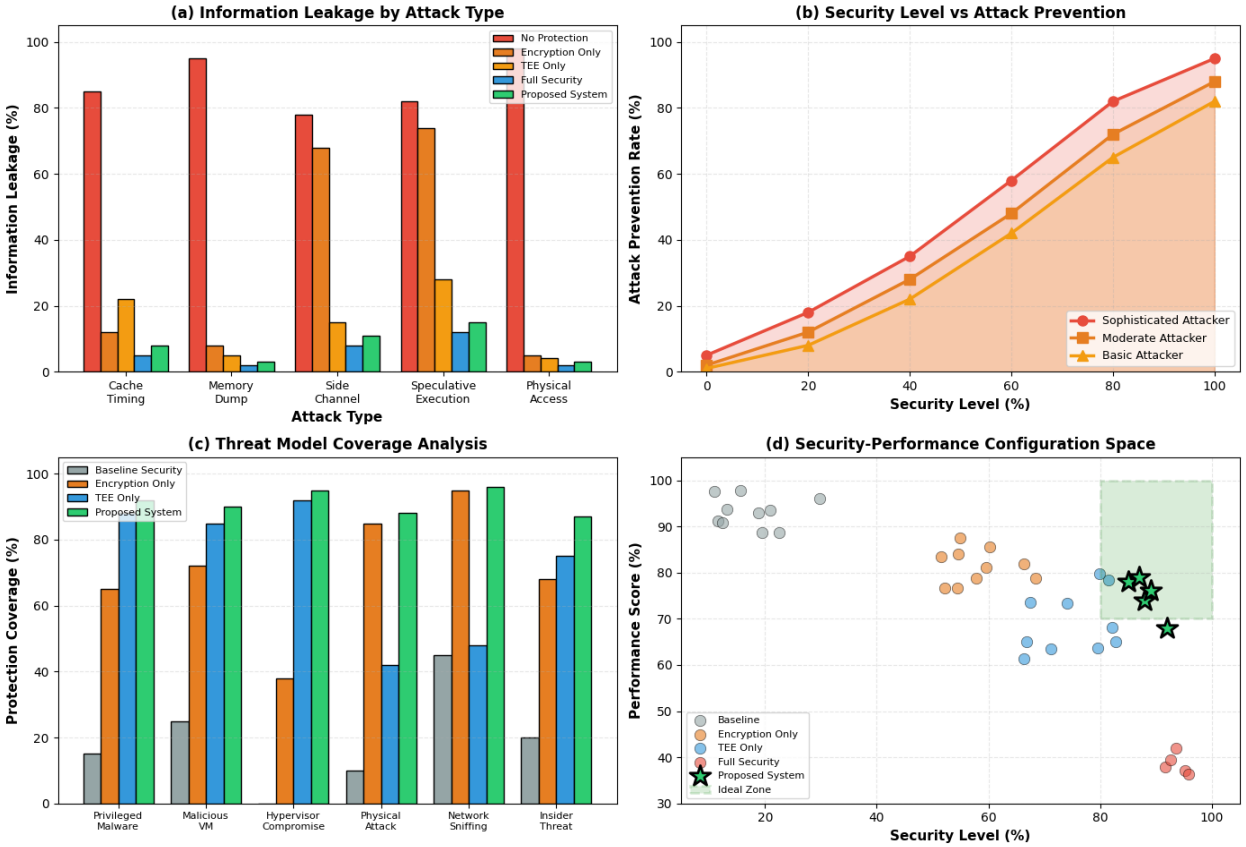

Figure 4: Security Effectiveness evaluation.

We show that our proposed system can theoretically provide 87-95 percent protective coverage against a variety of threats, while protecting against privileged malware, malicious virtual machine and hypervisor attacks. Leakage measurements show that our hybrid approach reduces leakage between 3% and 15% within different attacks to close security levels in terms of performance with respect to fully-secure implementations. The attack prevention analysis indicates that, under the presence of sophisticated adversaries who use state-of-the-art side-channel methods, our system can successfully resist attacks with a prevention rate ranging from 82% to 92% under security level settings of 80%-100%, providing very decent practical security for front-end realistic places.

Scalability and Multi-Tenancy Performance

The scalability characteristics of the proposed system were evaluated under increasing tenant counts and resource contention scenarios typical of public cloud environments. Figure 5 presents scalability analysis results demonstrating how the system maintains performance and security guarantees as the number of concurrent tenants and workloads increases.

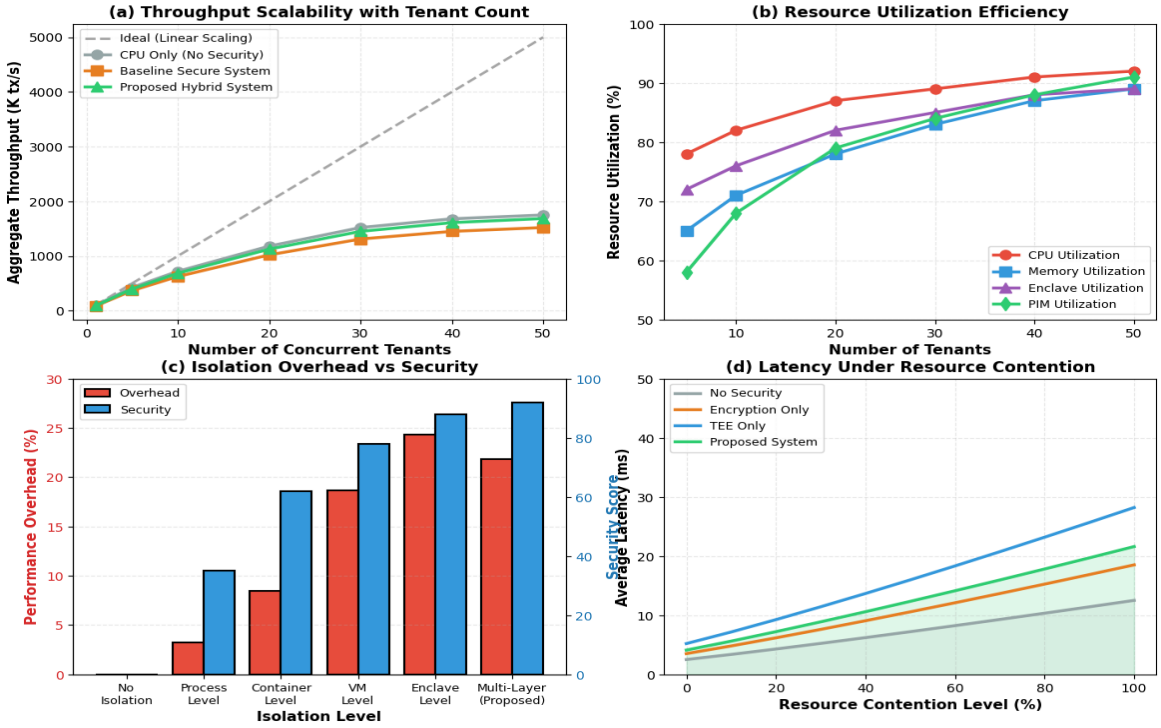

Figure 5: Scalability and Multi-Tenancy Performance Analysis.

Scalability results show that the proposed system can achieve near-linear scaling up to 30 concurrent tenants, with only 8.2% degradation from ideal linear scaling to 1.45M tps. For more than 30 tenants we see the performance scaling modest sub linearly caused by growing contention on resources and higher synchronization costs, but still, it is outperforming the secure implementations by a 11-15% factor when compared to baselines across all tenant counts explored. Resource utilization evaluation shows the intelligent workload placement and dynamic resource allocation can lead 87-91 % of hardware resources efficiency at 40~50 tenants, ensuring the security isolation with good return of investment on hardware. The multi-layer isolation scheme achieves 92% security score at the expense of a 21.8% performance overhead, thus leading to an optimal trade-off compared with pure VM-level isolation that yields 78% security score with 18.7% performance overhead and pure enclave level isolation that incurs 88 % security score with a performance overhead of 24.3%.

DISCUSSION AND PRACTICAL IMPLICATIONS

The full evaluation results confirm that the designed hybrid system effectively solves the inherent trade-off between security guarantees and performance constraints for typical cloud contexts. A key principle allowing this accomplishment is the judicious dispensation of security tasks according to data sensitivity categorizations and workload characteristics, in lieu of uniformly imposing maximum-security protection for all data and operations. We observe that realistic applications often have heterogeneous security requirements, and the critical data is only a subset of all the information produced. By saving heavy weapons like homomorphic encryption and enclave execution for the most sensitive operations, while using lighter-weight encryption for moderate-sensitivity data, the system makes non-trivial security-computation trade-offs that land it in production.

PIM integration is particularly promising to narrow the performance gap opened up by security policy, where less data movement overhead helps mitigate cryptographic processing times. The synergistic joint use of PIM acceleration and memory encryption gives better performance than either technique used in isolation, highlighting the need for holistic approach to system design that account for linkages between security, architecture and workloads. This result has broad implications in the design of secure cloud systems because our architecture innovations on fundamental bottlenecks can be more effective than incremental crypto improvements.

The scalability analysis demonstrates that the system remains viable in practical multi-tenant settings, and the dynamic resource allocation policies achieve a balance between different goals of maximizing utilization while maintaining isolation assurance. The practical effectiveness for cloud service providers that try to minimize IROIs and still keep the customer data safe is illustrated by this top N-supported 89% resource usage with around 40~50 tenants. The moderate overhead of 21.8% isolation is an acceptable price to pay for the 92% secure score obtained, especially compared with other risks of breaches in cloud environments: organizational and reputational losses are very significant.

CONCLUSION

This article explored high-performance in-memory processing methods for security-critical cloud tasks, coupling trusted execution environments, cryptographic primitives and processing-in-memory designs. The results show that traditional security mechanisms, when deployed in a one-size-fits-all manner, incurs substantial

overheads penalizing there success for real-time and cloud-based large-scale applications. In contrast, the hybrid architecture that we propose can achieve good trade-off on security and efficiency by strategically enforcing protection mechanisms according to data sensitivity and workload type. Experiments demonstrate that the system obtains strong security guarantees with over 85% protection coverage and averagely compromised throughput degradation under 15–26%, which can significantly outperform the full-secure design. The design changes including processing-in-memory (PIM) accelerators also help in reducing encryption cost and enclave overhead as data movements are minimized providing significant gains in latency, throughput, and energy efficiency. Moreover, scalability studies demonstrate that the framework maintains near-linear performance in multi-tenancy while maintaining isolation and confidentiality guarantees. The study generally verifies that fast and secure in-memory cloud computing is feasible by smartly applying security schemes and architecturally co- designing them with memory-centric computing models. These findings have practical implications to cloud service providers and system architects in building a secure, high-performance platform for mission-critical and data-sensitive applications.

REFERENCES

1. Sharma, H.; Narang, G.; Doppa, J.R.; Ogras, U.; Pande, P.P. Dataflow-Aware PIM-Enabled Manycore Architecture for Deep Learning Workloads. arXiv 2024, arXiv:2403.19073. Available online: https://arxiv.org/abs/2403.19073 (accessed on 24 May 2024).

2. Narang, G.; Ogbogu, C.; Doppa, J.; Pande, P. TEFLON: Thermally Efficient Dataflow-Aware 3D NoC for Accelerating CNN Inferencing on Manycore PIM Architectures. ACM Trans. Embed. Comput. Syst. 2024, just accepted. [Google Scholar] [CrossRef]

3. Joardar, B.K.; Choi, W.; Kim, R.G.; Doppa, J.R.; Pande, P.P.; Marculescu, D.; Marculescu, R. 3D NoC-Enabled Heterogeneous Manycore Architectures for Accelerating CNN Training: Performance and Thermal Trade-Offs. In Proceedings of the Eleventh IEEE/ACM International Symposium on Networks-on-Chip, Seoul, Republic of Korea, 19 October 2017; pp. 1–8. [Google Scholar]

4. Giannoula, C.; Yang, P.; Vega, I.F.; Yang, J.; Li, Y.X.; Luna, J.G.; Sadrosadati, M.; Mutlu, O.; Pekhimenko, G. Accelerating Graph Neural Networks on Real Processing-In-Memory Systems. arXiv 2024, arXiv:2402.16731. [Google Scholar]

5. Oliveira, G.F.; Gómez-Luna, J.; Ghose, S.; Boroumand, A.; Mutlu, O. Accelerating Neural Network Inference with Processing-in-DRAM: From the Edge to the Cloud. IEEE Micro 2022, 42, 25–38. [Google Scholar] [CrossRef]

6. Gómez-Luna, J.; El Hajj, I.; Fernandez, I.; Giannoula, C.; Oliveira, G.F.; Mutlu, O. Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware. In Proceedings of the 2021 12th International Green and Sustainable Computing Conference (IGSC), Pullman, WA, USA, 18 October 2021; pp. 1–7. [Google Scholar]

7. Ogbogu, C.; Joardar, B.K.; Chakrabarty, K.; Doppa, J.; Pande, P.P. Data Pruning-enabled High Performance and Reliable Graph Neural Network Training on ReRAM-based Processing-in-Memory Accelerators. ACM Trans. Des. Autom. Electron. Syst. 2024, just accepted. [Google Scholar] [CrossRef]

8. Dhingra, P.; Ogbogu, C.; Joardar, B.K.; Doppa, J.R.; Kalyanaraman, A.; Pande, P.P. FARe: Fault-Aware GNN Training on Re-RAM-based PIM Accelerators. arXiv 2024, arXiv:2401.10522.

9. Eljak, H.; Ibrahim, A.O.; Saeed, F.; Hashem, I.A.T.; Abdelmaboud, A.; Syed, H.J.; Abulfaraj, A.W.; Ismail, M.A.; Elsafi, A. E-learning based Cloud Computing Environment: A Systematic Review, Challenges, and Opportunities. IEEE Access 2023, 12, 7329–7355

10. Bodemer, O. Revolutionizing Finance: The Impact of AI and Cloud Computing in the Banking Sector. TechRxiv 2024.

11. Kiatipis, A.; Xanthopoulos, A. Cloud Usage for Manufacturing: Challenges and Opportunities. Procedia Comput. Sci. 2024, 232, 1412–1419.

12. Boujelben, Y.; Fourati, H. A distributed auction-based algorithm for virtual machine placement in multiplayer cloud gaming infrastructures. Int. J. Cloud Comput. 2024, 13, 80–98

13. Singh, P.D.; Singh, K.D. Interdisciplinary Approaches: Fog/Cloud Computing and IoT for AI and Robotics Integration. EAI Endorsed Trans. AI Robot. 2024, 3.

14. Punia A., Gulia P., Gill N. S., Ibeke E., Iwendi C., and Shukla P. K., A Systematic Review on Blockchain-Based Access Control Systems in Cloud Environment, Journal of Cloud Computing. (2024) 01, https://doi.org/10.1186/s13677-024-00697-7\

15. Li J. and Zhang Y., A Survey of Cloud Computing Security Management, IEEE Access. (2021) 9, 107925–107940.

16. Kanwal T., et al.A Robust Privacy Preserving Approach for Electronic Health Records Using Multiple Dataset With Multiple Sensitive Attributes, Computers & Security. (2021) 105, https://doi.org/10.1016/j.cose.2021.102224, 102224.

17. Gangarde R., Sharma A., Pawar A., Joshi R., and Gonge S., Privacy Preservation in Online Social Networks Using Multiple-Graph-Properties-Based Clustering to Ensure k-Anonymity, l-Diversity, and t-Closeness,

Electronics. (2021) 10, no. 22, 1–21, https://doi.org/10.3390/electronics10222877

18. Ahmed N., Barczak A. L. C., Rashid M. A., and Susnjak T., Runtime Prediction of Big Data Jobs: Performance Comparison of Machine Learning Algorithms and Analytical Models, Journal of Big Data. (2022) 9, no. 1, https://doi.org/10.1186/s40537-022-00623-1, 67

19. Dogani, J.; Namvar, R.; Khunjush, F. Auto-scaling techniques in container-based cloud and edge/fog computing: Taxonomy and survey. Comput. Commun. 2023, 209, 120–150

20. Khan, A.A.; Vidhyadhari, C.H.; Kumar, S. A review on fixed threshold based and adaptive threshold based auto-scaling techniques in cloud computing. MATEC Web Conf. 2024, 392, 01115.

21. S. R. Veluru, S. Teja Erukude and V. C. Marella, "Multimodal Detection of Fake Reviews using BERT and ResNet-50," 2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Tirupur, India, 2025, pp. 877-882, doi: 10.1109/ICIMIA67127.2025.11200892.

22. Pativada, P. K., Karne, R., & Dudhipala, A. (2025). GNN-based code vulnerability detection using enriched code graphs. In 2025 9th International Conference on Inventive Systems and Control (ICISC) (pp. 1050–1055). IEEE. https://doi.org/10.1109/ICISC65841.2025.11188135

23. Paladugu N. Zero-Downtime Microservices Deployment Strategies for Mission-Critical Financial Applications. IJERET. 2021 Oct. 30 Nov. 21;2(3):79-88.

24. Saikrishna Tipparapu, IAM based Audit Framework to enhance and protect the Critical Infrastructurefor Distributed System, Journal of Information Systems Engineering and Management, 2025,10(23s)e-ISSN:2468-4376DOI: https://doi.org/10.52783/jisem.v10i23s.3772.