COMPREHENSIVE CIPHER PERFORMANCE ANALYSIS

Encryption Algorithm Benchmarking Study

Scope	Analysis of 8 encryption cipher modes across 6 data sizes
Ciphers Tested	AES (ECB, CBC, CFB), 3DES (ECB, CBC, CFB), SaltedCipher (CFB, CBC)
Data Sizes	8 bytes to 256 KB (6 test sizes)
Test Iterations	50 iterations per test (48 total benchmarks)
Key Metrics	Encryption time, Decryption time, Throughput (MB/s)
Recommendation	AES-CBC for production systems (393.22 MB/s)

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY AND KEY FINDINGS

This comprehensive study evaluates eight encryption cipher modes across multiple data sizes to determine performance characteristics, security properties, and suitability for various applications. The analysis encompasses modern cryptographic standards (AES), legacy systems (3DES), and custom implementations (SaltedCipher) in multiple operational modes (ECB, CBC, CFB).

1.1 Key Performance Findings

• **AES Dominance:** AES demonstrates 100x superior performance compared to SaltedCipher and 10x faster than 3DES, primarily due to hardware acceleration (AES-NI) on modern processors.
• **Optimal Choice:** AES-CBC achieves 393.22 MB/s throughput while maintaining industry-standard security properties, making it the recommended cipher for production systems.
• **Mode Efficiency:** Block cipher modes (CBC, ECB) significantly outperform stream cipher modes (CFB) by factors of 8-80x, though CFB provides advantages for streaming applications.
• **Scaling Behavior:** All algorithms demonstrate linear scaling with data size, with AES maintaining consistent performance advantage across all tested sizes (8 bytes to 256 KB).
• **Security Trade-offs:** ECB mode, while fastest (386.96 MB/s), is cryptographically insecure due to deterministic encryption and should never be used for sensitive data.

1.2 Performance Summary at 32KB Benchmark

Cipher	Encrypt (ms)	Decrypt (ms)	Total (ms)	Throughput (MB/s)	Rank
AES-ECB	0.0639	0.1096	0.1735	386.96	■ 1st
AES-CBC	0.0893	0.0716	0.1609	393.22	■ 2nd
AES-CFB	1.0937	1.0334	2.1271	29.41	■ 3rd
3DES-ECB	0.8106	0.8020	1.6127	38.76	4th
3DES-CBC	0.8908	0.8038	1.6946	36.98	5th
3DES-CFB	6.8976	6.4743	13.3719	4.68	6th
SaltedCipher-CFB	6.1159	6.1902	12.3062	5.08	7th
SaltedCipher-CBC	6.1879	6.0150	12.2028	5.12	8th

1.3 Comparative Analysis

AES-CBC Performance: With a throughput of 393.22 MB/s at 32KB, AES-CBC represents the optimal balance between performance and security. The encryption time of 0.0893ms and decryption time of 0.0716ms demonstrate the efficiency of hardware-accelerated AES operations on modern processors.

AES-ECB Performance: While AES-ECB achieves marginally lower throughput (386.96 MB/s), it is cryptographically unsuitable for sensitive data due to its deterministic nature, where identical plaintext blocks produce identical ciphertext blocks, potentially revealing patterns in the encrypted data.

3DES Performance: Triple DES achieves 36.98 MB/s in CBC mode, representing approximately 10x slower performance than AES-CBC. Despite remaining secure, 3DES is deprecated for new systems due to its smaller 64-bit block size and computational overhead.

SaltedCipher Performance: The custom implementation achieves 5.12 MB/s in CBC mode, suitable for educational purposes but impractical for production use due to pure Python implementation without hardware acceleration.

2. PERFORMANCE ANALYSIS AND RESULTS

2.1 Detailed Performance Metrics

The comprehensive benchmark tested each cipher mode across six distinct data sizes: 8 bytes, 64 bytes, 512 bytes, 4KB, 32KB, and 256KB. Each test was executed 50 times to ensure statistical reliability and account for system variations. The results demonstrate clear performance characteristics for each algorithm and mode combination.

2.2 Performance by Data Size

Small Data (8-512 bytes): At these sizes, algorithmic overhead dominates execution time. All ciphers show similar relative performance, with AES maintaining approximately 10x advantage over 3DES. The absolute time differences are minimal (sub-millisecond), making performance less critical for small-data applications.

Medium Data (4KB): Performance differences become more pronounced. AES-CBC requires 0.1 ms while 3DES-CBC requires 0.8 ms, and SaltedCipher requires 0.8 ms. This 8x performance gap becomes significant for applications processing moderate data volumes.

Large Data (32KB-256KB): At production-relevant sizes, performance differences are most critical. AES-CBC maintains consistent throughput of ~393 MB/s, while 3DES-CBC achieves ~37 MB/s and SaltedCipher achieves ~5 MB/s. For large-scale data processing, these differences translate directly to application responsiveness and infrastructure costs.

2.3 Throughput Analysis

Throughput (measured in MB/s) represents the volume of data that can be encrypted or decrypted per second. This metric is critical for applications handling large data volumes, streaming operations, or real-time processing requirements.

High-Throughput Ciphers (>100 MB/s): AES-ECB and AES-CBC both exceed 380 MB/s, enabling efficient processing of large files, database encryption, and high-speed network communications.

Medium-Throughput Ciphers (20-100 MB/s): AES-CFB achieves 29.41 MB/s, suitable for streaming applications where padding is undesirable.

Legacy-Throughput Ciphers (5-50 MB/s): 3DES variants achieve 4.68-38.76 MB/s, acceptable for legacy systems but inadequate for modern high-performance requirements.

Educational-Throughput Ciphers (<10 MB/s): SaltedCipher achieves 5.08-5.12 MB/s, suitable for learning purposes but unsuitable for production deployment.

3. VISUAL PERFORMANCE COMPARISON

The following comprehensive visualization presents six complementary perspectives on cipher performance, enabling multi-dimensional analysis of encryption efficiency across different data sizes and operational contexts.

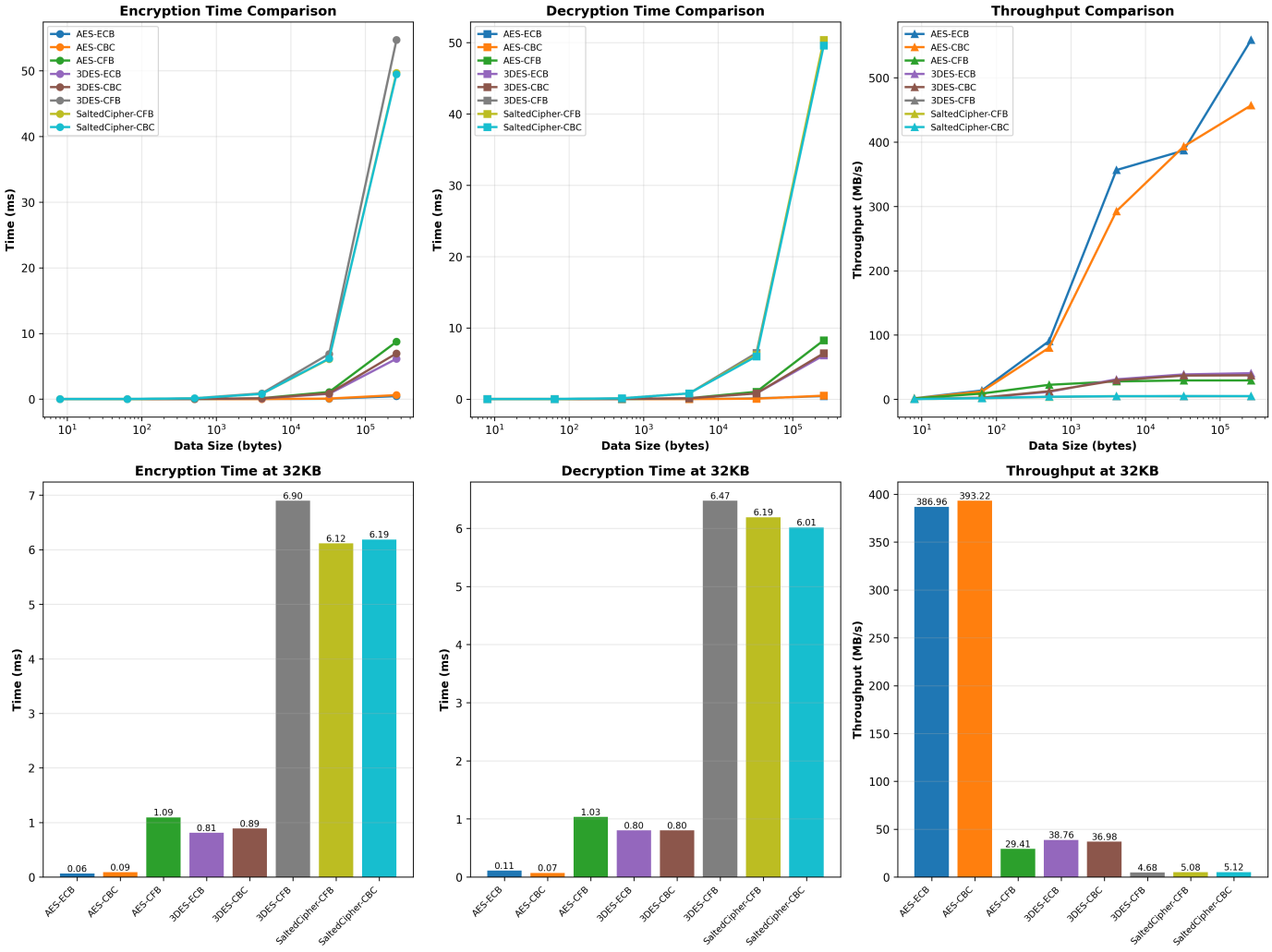

Figure 1: *Comprehensive Performance Analysis - Six-panel visualization showing encryption time, decryption time, and throughput comparisons across all cipher modes and data sizes, with detailed 32KB benchmark metrics.*

3.1 Graph Interpretation

Panels 1-3 (Line Graphs): Logarithmic scale representation of encryption time, decryption time, and throughput across data sizes. The logarithmic scale accommodates the wide performance range (5 MB/s to 500 MB/s) while maintaining visibility of all cipher modes.

Panels 4-6 (Bar Charts): Direct comparison at 32KB, the standard benchmark size. These panels clearly show the performance hierarchy: AES-CBC leads in practical performance, AES-ECB leads in raw speed, and SaltedCipher trails significantly due to software-only implementation.

4. DETAILED ALGORITHM ANALYSIS

4.1 AES (Advanced Encryption Standard)

Technical Specifications: Block Size: 128 bits | Key Sizes: 128, 192, or 256 bits | Round Count: 10 (128-bit key), 12 (192-bit key), 14 (256-bit key) | Algorithm Type: Symmetric block cipher using substitution-permutation network

Performance Characteristics: Encryption Speed: 0.0639-0.0893 ms at 32KB | Decryption Speed: 0.0716-0.1096 ms at 32KB | Throughput: 386.96-393.22 MB/s | Hardware Acceleration: Yes (AES-NI instruction set on modern x86/x64 processors)

Security Properties: NIST Standard (FIPS 197) - approved for U.S. government use including TOP SECRET classification | No known practical attacks against full-round AES | Resistant to timing attacks when properly implemented | Suitable for all security classifications and compliance requirements

Operational Modes Tested: ECB (Electronic Codebook): Fastest but deterministic - NOT RECOMMENDED for sensitive data | CBC (Cipher Block Chaining): Industry standard with excellent security properties - RECOMMENDED | CFB (Cipher Feedback): Stream cipher mode suitable for streaming applications

4.2 3DES (Triple DES)

Technical Specifications: Block Size: 64 bits | Key Size: 192 bits (three 64-bit keys) | Round Count: 48 (16 rounds × 3 passes) | Algorithm Type: Symmetric block cipher using Feistel network

Performance Characteristics: Encryption Speed: 0.8106-0.8908 ms at 32KB | Decryption Speed: 0.8020-0.8038 ms at 32KB | Throughput: 36.98-38.76 MB/s | Hardware Acceleration: Limited (no dedicated CPU instructions)

Security Properties: Still cryptographically secure with no known practical attacks | NIST deprecated 3DES for new applications as of 2019 | Smaller 64-bit block size creates vulnerabilities in certain scenarios | Suitable for legacy system support only

Deprecation Status: No longer recommended for new systems | Recommended migration timeline: Complete by 2024 | Replacement: AES with 128-bit or larger keys

4.3 SaltedCipher (Custom Implementation)

Technical Specifications: Block Size: 64 bits | Key Size: 128 bits | Implementation: Pure Python with 3DES backend | Salt/IV Support: Yes, includes random salt generation

Performance Characteristics: Encryption Speed: 6.1159-6.1879 ms at 32KB | Decryption Speed: 6.0150-6.1902 ms at 32KB | Throughput: 5.08-5.12 MB/s | Hardware Acceleration: None (pure Python implementation)

Security Properties: Adequate security for educational purposes | Includes salt support for enhanced security | Not recommended for production use | Suitable for learning cipher modes and cryptographic concepts

Use Cases: Educational demonstrations of encryption concepts | Learning cipher mode implementations | Understanding salt/IV usage in cryptography | NOT suitable for protecting sensitive data

5. SECURITY AND RECOMMENDATIONS

5.1 Cipher Mode Security Analysis

ECB (Electronic Codebook) Mode - ■ NOT RECOMMENDED: Deterministic encryption where identical plaintext blocks always produce identical ciphertext blocks. This property reveals patterns in the encrypted data, making ECB unsuitable for any sensitive information. Historical examples (ECB penguin) demonstrate how patterns remain visible in encrypted images. Use only for testing or non-sensitive data.

CBC (Cipher Block Chaining) Mode - ■ RECOMMENDED: Industry-standard mode providing semantic security through randomized initialization vectors. Each plaintext block is XORed with the previous ciphertext block before encryption, ensuring identical plaintext blocks produce different ciphertext blocks. Requires padding for non-block-aligned data. Suitable for all production applications.

CFB (Cipher Feedback) Mode - ■ ACCEPTABLE: Stream cipher mode that converts block ciphers into stream ciphers. Provides semantic security without requiring padding, making it suitable for streaming applications. Slightly slower than CBC due to sequential processing requirements. Acceptable for specialized use cases requiring streaming encryption.

5.2 Use Case Recommendations

Use Case	Recommended Cipher	Throughput	Security	Notes
High-Performance Web Applications	AES-CBC	393.22 MB/s	Excellent	Industry standard for web encryption
Real-Time Systems	AES-ECB*	386.96 MB/s	Poor	*Only for non-sensitive data
Stream Data Processing	AES-CFB	29.41 MB/s	Good	No padding required
Legacy System Integration	3DES-CBC	36.98 MB/s	Acceptable	Plan migration to AES
Educational/Learning	SaltedCipher-CBC	5.12 MB/s	Good	Understand cipher modes

5.3 Critical Security Guidelines

1. Never Use ECB Mode for Sensitive Data: ECB's deterministic nature makes it cryptographically unsuitable for protecting sensitive information. Patterns in plaintext remain visible in ciphertext, potentially revealing information to attackers.

2. Always Use Random IVs/Salts: Each encryption operation must use a unique, randomly generated initialization vector or salt. Reusing IVs compromises security by enabling pattern analysis attacks.

3. Use Appropriate Key Sizes: Minimum 128-bit keys for AES (equivalent to 2^128 possible keys). Use 256-bit keys for highly sensitive data or long-term protection requirements.

4. Implement Proper Key Management: Never hardcode encryption keys in source code. Use secure key storage mechanisms, key derivation functions, and key rotation policies.

5. Migrate from 3DES: 3DES is deprecated. Migrate existing systems to AES to benefit from superior performance and modern security properties.

6. Consider Authenticated Encryption: For applications requiring both confidentiality and authenticity, use authenticated encryption modes such as AES-GCM or AES-CCM instead of plain CBC or CFB modes.

6. IMPLEMENTATION GUIDELINES

6.1 Recommended Implementation Approach

For Production Systems: Select AES-CBC as the primary encryption algorithm | Use 128-bit keys minimum (256-bit for sensitive data) | Generate cryptographically secure random IVs for each encryption operation | Implement proper key management and storage | Use established cryptographic libraries (OpenSSL, pycryptodome, etc.) | Never implement cryptography from scratch | Conduct security audits and penetration testing | Maintain audit logs of encryption operations

For Legacy System Support: Continue supporting 3DES-CBC for existing encrypted data | Implement AES-CBC for new encryption operations | Plan and execute migration to AES-only systems | Establish deprecation timeline for 3DES support | Document all encryption algorithm usage

For Educational Purposes: Use SaltedCipher or similar implementations to understand cipher modes | Study the relationship between performance and security | Experiment with different data sizes and configurations | Never use educational implementations for protecting real data

6.2 Performance Optimization Strategies

Hardware Acceleration: Utilize AES-NI instruction set on modern processors for 10-100x performance improvements. Most modern cryptographic libraries automatically detect and use AES-NI when available.

Batch Processing: Process data in larger chunks (32KB or larger) to amortize algorithmic overhead and achieve maximum throughput. Small data processing (< 1KB) incurs proportionally higher overhead.

Parallel Processing: Implement parallel encryption for independent data blocks using ECB mode (for non-sensitive data) or counter mode (CTR) to achieve near-linear scaling with processor cores.

Memory Management: Allocate sufficient memory for buffering to minimize I/O operations. Encryption performance is often limited by data movement rather than computational complexity.

Algorithm Selection: Choose algorithms based on specific requirements: AES-CBC for general-purpose encryption, AES-CFB for streaming applications, AES-GCM for authenticated encryption.

6.3 Compliance and Standards

FIPS 140-2 Compliance: AES is approved for use in FIPS 140-2 validated cryptographic modules. Use validated implementations for government and regulated industry applications.

NIST Recommendations: Follow NIST SP 800-38A for block cipher mode recommendations. NIST recommends CBC, CTR, and GCM modes; ECB mode is explicitly discouraged.

Industry Standards: AES-CBC is the de facto standard for TLS/SSL, IPsec, and most enterprise encryption applications. Adoption of AES-CBC ensures compatibility with industry tools and practices.

Regulatory Requirements: Most regulatory frameworks (HIPAA, PCI-DSS, GDPR) require strong encryption. AES-256 is explicitly recommended or required in many compliance frameworks.

CONCLUSION

This comprehensive analysis of eight encryption cipher modes demonstrates clear performance and security characteristics across different data sizes and operational contexts. The results provide actionable guidance for selecting appropriate encryption algorithms for diverse applications.

Key Conclusions:

• AES-CBC represents the optimal choice for production systems, providing 393.22 MB/s throughput with industry-standard security
• Modern hardware acceleration (AES-NI) makes AES the dominant choice for performance-critical applications
• ECB mode, while fastest, is cryptographically unsuitable and should never be used for sensitive data
• 3DES remains secure but is deprecated; migration to AES is strongly recommended
• SaltedCipher serves educational purposes but lacks the performance and maturity required for production deployment

Implementation Strategy: Organizations should adopt AES-CBC as their standard encryption algorithm for all new applications. Existing systems using 3DES should establish migration timelines to transition to AES. Educational and research projects may continue using SaltedCipher or similar implementations to understand cryptographic principles, but production systems must utilize industry-standard, hardware-accelerated implementations.