# Advanced Encryption Standard New Instructions (AES-NI) Analysis: Security, Performance, and Power Consumption

Eslam G. AbdAllah[1], Yu Rang Kuang[2], and Changcheng Huang[1]
[1]Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada
[2]Quantropi Inc., Ottawa, ON, Canada
[1]{eslamabdallah, huang}@sce.carleton.ca [2]{randy.kuang}@quantropi.com

## ABSTRACT
Advanced Encryption Standard New Instructions (AES-NI), which is a hardware accelerated version of the Advanced Encryption Standard (AES), are a set of instructions that enable fast and secure data encryption and decryption. AES-NI consists of seven instructions and supports all usage and modes of operations of AES.

In this paper, we analyze AES-NI from security, performance, and power consumption perspectives. We compare AES-NI with the traditional AES. Through our experiments, we measure confusion, diffusion, randomness, as well as encryption and decryption speed, and power consumption. Our experiments are carried out on different processor platforms and with different encryption workloads. Our experimental results show that AES-NI achieves up to 13.5x speed over AES at 90% reduced energy consumption over AES. The low energy footprint makes AES-NI a candidate for secure communication for IoT and other lightweight edge devices.

## CCS Concepts
• **Security and privacy→Cryptography→Symmetric cryptography and hash functions→Block and stream ciphers**

## Keywords
AES-NI, security analysis, performance analysis, power consumption analysis


ICCAE 2020, February 14–16, 2020, Sydney, NSW, Australia

ACM ISBN 978-1-4503-7678-5/20/02…$15.00
https://doi.org/10.1145/3384613.3384648

## 1 INTRODUCTION
The goal of cryptography is to protect data against attackers or outside people by preventing their unauthorized use or alteration. One of the most widely used cryptographic technique is the Advanced Encryption Standard (AES). AES is adopted by the U.S. government in 2001. AES is used in various domains to protect different applications such as online transactions, databases, and disk encryption. The main challenge is that traditionally, cryptography is computationally costly to execute. To overcome this challenge, Intel introduced the Advanced Encryption Standard New Instructions (AES-NI) for better performance and security than AES [1, 2].

AES consists of four stages, one permutation and three substitutions. Substitute bytes stage uses an S-box to perform a byte-by-byte substitution of the block. ShiftRows stage is a simple permutation. MixColumns stage performs substitution based on arithmetic over GF($2^8$). AddRoundKey stage, which is a simple bitwise XOR of the current block with part of the expanded key. AES encryption and decryption use nine rounds with the four stages followed by a tenth round of three stages. Both encryption and decryption start with AddRoundKey stage. The four stages are easily reversible. Each stage, except AddRoundKey, has an inverse function that is used in the decryption algorithm. For the AddRoundKey stage, the inverse is performed by XORing the same round key to the block. Only the AddRoundKey stage uses the secret key, while the other three stages achieve confusion, diffusion, and nonlinearity [3].

AES-NI implements complex and costly sub-steps of the AES algorithm in hardware, which accelerates the execution of the AES-based encryption. AES-NI is a set of seven new instructions. Four instructions (AESENC, AESENCLAST, AESDEC, and AESDELAST) achieve high performance for encryption and decryption. Two instructions (AESIMC and AESKEYGENASSIST) support the AES key expansion. The seventh instruction (PCLMULQDQ) is used to perform carryless multiplication. AES-NI is now integrated into many processors such as Intel, latest Scalable Processor Architecture (SPARC), and latest ARM processors [4, 5].

In this paper, we evaluate AES-NI from security, performance, and power consumption perspectives. For security experiments, we measure the diffusion, confusion, and initial vector (IV) change. In this paper, we use the Cipher Block Chaining (CBC) mode with

256 bytes as a key length (AES-256-CBC). We measure the randomness based on entropy and randomness online tester, Dieharder and the National Institute of Standards and Technology Statistical Test Suite (NIST STS). We perform performance and power consumption experiments across multiple platforms. Based on our results, the benefits of AES-NI can be summarized in the following points.

**Performance improvement**. AES-NI provides significant speedup of AES and it can reach up to 13.5x of AES.

**Security improvement**. AES-NI addresses the side channel attacks on AES. AES-NI instructions are able to perform the decryption and encryption in hardware and does not use the software lookup tables. Hence, using AES-NI decreases the risk of side-channel attacks in addition to the performance improvement.

**Power consumption saving**. AES-NI saves up to 90% of the power consumed in the traditional AES.

In the following sections, we present AES-NI security, performance, and power consumption experiments and results. Performance and power consumption results are carried out on different processor platforms and with different encryption workloads.

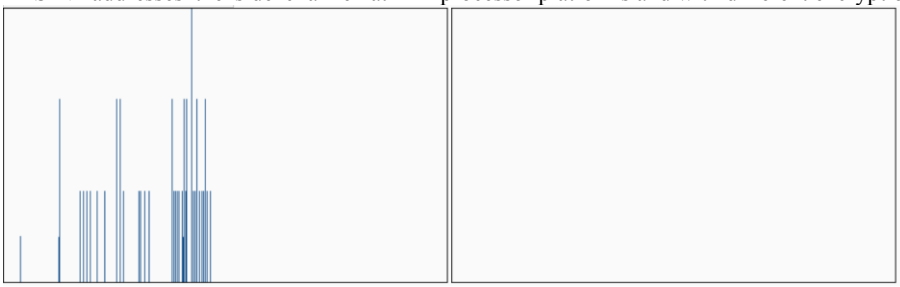


**Figure 1: Biased plaintext entropy and randomness**

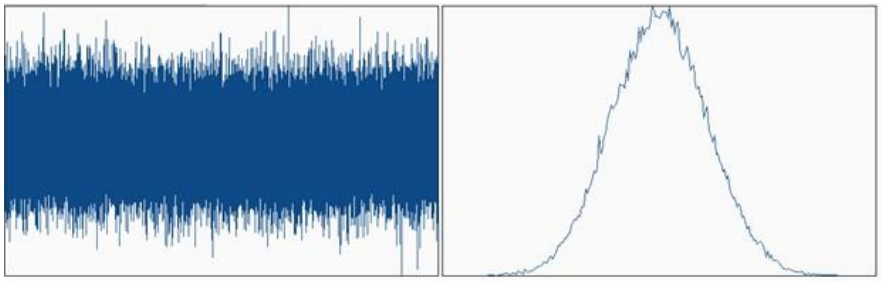


**Figure 2: AES-NI entropy and randomness**

## 2 SECURITY ANALYSIS

In this section, we present our security experiments and results for AES-NI. We measure the diffusion, confusion, and IV change. Diffusion and Confusion are the cryptographic measures where diffusion tries to maximize the effect of each individual plaintext digit over around half of ciphertext digits. On the other hand, the confusion's goal is to make a relationship between the statistics of the ciphertext and the value of the encryption key as complicated as possible. We also measure the randomness based on entropy and randomness online tester, Dieharder and NIST statistical test suite.

**Diffusion**. Means that if we change a single bit of the plaintext, then (statistically) around half of the bits in the ciphertext should change. We change a single bit in the plaintext, while we use the same key and IV. AES-NI achieves close to 50%, as shown in Table 1. The test is performed using the same input. The input size Is 1 KB.

**Confusion.** Means that each binary digit (bit) of the ciphertext should depend on several parts of the key, obscuring the connections between the key and the ciphertext. We change a single bit in the key, while we use the same plaintext and IV. As shown in Table 1, AES-NI achieves close to 50%, which is the ideal target value (coin flip). The test is performed using the same input. The input size is 1KB.

**IV change**. In this test, we change one bit in the IV, while the key and plaintext are the same. Results in Table 1 show similar results for AES-NI. Practically, this is very useful case as users will send messages with random IV with each message. Even the plaintext repeated, the generated ciphertext will be around 50% different. The test is performed using the same input. The input size is 1KB.

**Table 1: Security analysis AES-NI (%)**

| Diffusion | Confusion | IV change |
|---|---|---|
| 49.78 | 49.61 | 49.80 |

**Entropy and randomness test.** This experiment makes use of 16 bits Shannon entropy calculator to test serial correlation of binary files [6]. It uses gnuplot to create the frequency and distribution graphs useful for testing normality. Gnuplot is a command-line program that can generate two- and three-dimensional plots of functions, data, and data fits. The results are used to estimate the strength and quality of random number generators. We use biased plaintext as shown in Figure 1. We repeat the same short message too many times. However, the output as shown in Figure 2, has equal and normal distribution across all values. The graph in Figure 2 shows that AES-NI is near ideal entropy randomness with no bias for certain data values (left) and the output cipher is random and follow a Gaussian normal distribution (right). The test is performed using the same input. The input size is 100MB.

**Dieharder.** The main goal of dieharder is to test random number generators for research and cryptography purposes [7]. The tool is built entirely on top of the Gnu Scientific Library's (GSL) random number generator interface and uses a variety of other GSL tools sort, erfc, incomplete gamma, distribution generators) in its operation. The full suite consists of 114 tests. It has three outputs for each test: pass, weak, and fail.

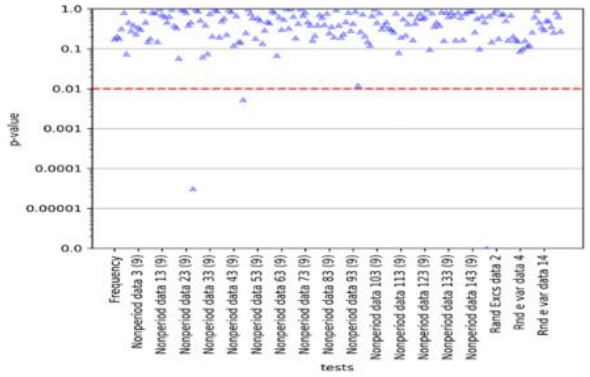
From the test results shown in Table 2, AES-NI can be considered as perfect random number generators or encryption algorithms. The weak test does not describe an attack opportunity, but is a side-effect of the random test process, even the baseline random number generators given in the dieharder tool also have some weak or fail tests. The test is performed using the same input. The input size is 100MB.

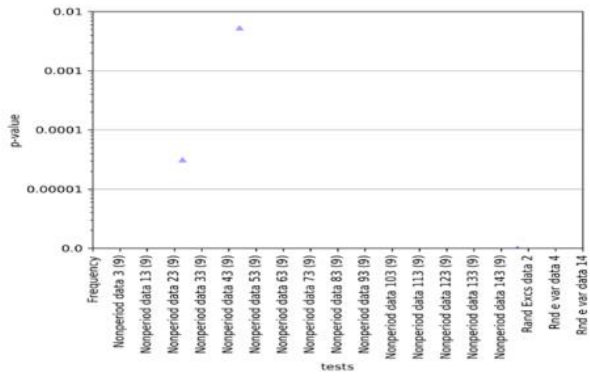
**NIST suite.** A statistical test suite for random and pseudorandom number generators for cryptographic applications [8]. This test suite is widely used in the evaluation and validation of industry standard cryptographic algorithms before they are considered safe for adoption. It consists of fifteen tests: frequency test, test for frequency within a block, runs test, test for the longest run of ones in a block, random binary matrix rank test, discrete fourier transform (spectral) test, non-overlapping (aperiodic) template matching test, overlapping (periodic) template matching test, maurer's universal statistical test, linear complexity test, serial test, approximate entropy test, cumulative sum (cusum) test, random excursions test, and random excursions variant test.

The results presented in Table 2 show that AES passes all tests. Figure 3.a shows the output p-values for each subtest in the fifteen tests. AES-NI has two points that are below the border line as shown in Figure 3.b. Figure 4 shows the generated histogram for AES-NI. The test is performed using the same input. The input size is 100MB.

**Table 2: Dieharder and NIST results for AES-NI**

| Test | Pass | Weak | Fail |
|---|---|---|---|
| Dieharder (114 tests) | 113 | 1 | 0 |
| NIST (15 tests) | 15 | 0 | 0 |




**(a) AES-NI p-values** **(b) AES-NI zoomed p-values**

**Figure 3: NIST statistical test suite: AES-NI p-values and zoomed p-values charts**

# 3 PERFORMANCE ANALYSIS

In the performance related experiments, we measure AES-NI speed in different platforms and with various data sizes: 16 bytes, 64 bytes, 256 bytes, 1 KB, 8 KB, and 16 KB as shown in Table 3 to Table 7. We compare between AES-NI and AES. We measure AES and AES-NI speed using openssl tool [9]. OpenSSL is a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. It is also a general-purpose cryptography library. We test the speed on the following platforms:

- Intel(R) Core(TM) i5-8250U CPU@1.6GHZ OS: Linux v19.04 (Table 3)
- Intel(R) Core(TM) i5-8250U CPU@1.6GHZ OS: Windows 10 (Table 4)
- Intel(R) Pentium Silver J5005 CPU@1.5GHZ OS: Linux v18.10 (Table 5)
- Intel(R) Pentium Silver J5005 CPU@1.5GHZ OS: Windows 10 (Table 6)
- ARMv8 Processor rev 1 (v81) OS: JetPack 4.2 (L4T 32.1) (Table 7)

The results as depicted in the performance analysis tables, show that AES-NI is significantly faster than AES on all platforms. Speed is measured by megabytes per second. AES-NI on Intel(R) Core(TM) i5 processor can encrypt around 920 MB/S. AES-NI is up 13.5 times faster than AES on this Intel processor. AES-NI on ARMv8 processor can encrypt around 355 MB/S. AES-NI is up to 10 times faster than AES on the ARM processor. Results also show that the performance on linux in most cases for AES and AES-NI is better than windows OS for the same CPU.

# 4 POWER CONSUMPTION ANALYSIS

In the power consumption experiments, we measure AES-NI power consumption and compare it with AES. For ARM processors, we use a physical power meter device to measure the power

consumption and for intel processor, we use powerstat tool [10] to measure the consumed battery power.

We test AES and AES-NI power consumption for the following platforms:

- ARMv8 Processor rev 1 (v81) OS: Ubuntu 18.04.2 LTS (Table 8)
- Intel(R) Core(TM) i5-8250U CPU@1.6GHZ OS: Linux v19.04 (Table 9)

Energy is a desirable property for IoT space, where longer lasting devices deployed in the field can create a significant competitive advantage. Also, higher speed, and lower power footprint mean reduced thermal footprint, which is desirable in constraint environments such as Radio frequency Identification (RFID) tags and Internet of Things (IoT) low-energy devices. AES-NI is an energy saving solution. The results as depicted in the power consumption analysis tables, show that AES-NI uses much less power than AES.

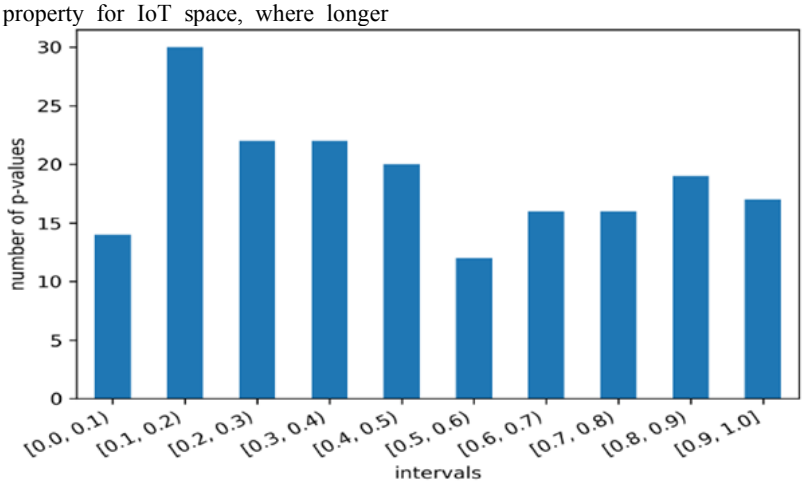



**Figure 4: NIST statistical test suite: AES-NI histogram for all tests**

**Table 3: Performance Comparison with different data sizes**
**Platform: Intel(R) Core(TM) i5-8250U CPU@1.6GHZ OS: Linux v19.04**

| | | 16 bytes | 64 bytes | 265 bytes | 1024 bytes | 8192 bytes | 16384 bytes |
|---|---|---|---|---|---|---|---|
| Speed (MB/S) | AES | 99.68 | 108.33 | 110.27 | 111.00 | 111.29 | 111.30 |
| | AES-NI | 758.91 | 913.88 | 923.13 | 926.47 | 927.36 | 927.03 |
| Ratio | AES-NI/AES | 7.61 | 8.44 | 8.37 | 8.35 | 8.33 | 8.33 |

**Table 4: Performance Comparison with different data sizes**
**Platform: Intel(R) Core(TM) i5-8250U CPU@1.6GHZ OS: Windows 10**

| | | 16 bytes | 64 bytes | 265 bytes | 1024 bytes | 8192 bytes |
|---|---|---|---|---|---|---|
| Speed (MB/S) | AES | 96.45 | 97.41 | 104.93 | 118.11 | 116.92 |
| | AES-NI | 749.32 | 899.34 | 911.20 | 847.41 | 846.94 |
| Ratio | AES-NI/AES | 7.8 | 9.2 | 8.6 | 7.2 | 7.2 |

**Table 5: Performance Comparison with different data sizes**
**Platform: Intel(R) Pentium Silver J5005 CPU@1.5GHZ OS: Linux v18.10**

| | | 16 bytes | 64 bytes | 265 bytes | 1024 bytes | 8192 bytes | 16384 bytes |
|---|---|---|---|---|---|---|---|
| Speed (MB/S) | AES | 49.00 | 51.91 | 53.12 | 120.87 | 121.49 | 121.79 |
| | AES-NI | 325.81 | 603.58 | 696.15 | 756.44 | 775.36 | 746.16 |
| Ratio | AES-NI/AES | 6.65 | 11.63 | 13.11 | 6.26 | 6.38 | 6.13 |

**Table 6: Performance Comparison with different data sizes**
**Platform: Intel(R) Pentium Silver J5005 CPU@1.5GHZ OS: Windows 10**

| | | 16 bytes | 64 bytes | 265 bytes | 1024 bytes | 8192 bytes |
|---|---|---|---|---|---|---|
| Speed (MB/S) | AES | 46.91 | 50.11 | 51.54 | 116.24 | 117.76 |
| | AES-NI | 463.50 | 653.28 | 699.44 | 733.87 | 745.63 |
| Ratio | AES-NI/AES | 9.88 | 13.04 | 13.57 | 6.31 | 6.33 |

**Table 7: Performance Comparison with different data sizes**
**Platform: ARMv8 Processor rev 1 (v81) OS: JetPack 4.2 (L4T 32.1)**

| | | 16 bytes | 64 bytes | 265 bytes | 1024 bytes | 8192 bytes | 16384 bytes |
|---|---|---|---|---|---|---|---|
| Speed (MB/S) | AES | 30.43 | 32.23 | 33.09 | 33.39 | 33.29 | 33.19 |
| | AES-NI | 157.47 | 260.94 | 306.64 | 324.74 | 355.01 | 333.08 |
| Ratio | AES-NI/AES | 5.17 | 8.10 | 9.27 | 9.73 | 10.06 | 10.04 |

AES-NI on ARM v8 processor uses around 2.7 Joule per Gigabyte (J/GB), while AES consumes 29 J/GB. AES-NI saves up to around 90% of consumed power by AES in the ARM processor. AES-NI on Intel Core (TM) i5-8250U processor uses around 2.2 J/GB, while AES consumes 20 J/GB on the intel processor. AES-NI saves up to around 90% of consumed power by AES in the Intel processor.

**Table 8: Power Consumption for ARMv8 Processor rev 1 (v81) OS: Ubuntu 18.04.2 LTS**

| | Baseline | AES | AES-NI |
|---|---|---|---|
| Voltage (V) | 5.16 | 5.16 | 5.16 |
| Current (A) | 0.31 | 0.6 | 0.57 |
| Power (W) | 1.6 | 3.1 | 2.95 |
| Power usage | - | 1.5 | 1.35 |
| Speed (1K data size) (MB/S) | - | 51.76 | 504.89 |
| Energy per GB (J/GB) | - | 28.9 | 2.7 |

**Table 9: Power Consumption for Intel(R) Core(TM) i5-8250U CPU@1.6GHZ OS: Ubuntu 18.04.2 LTS**

| | Baseline | AES | AES-NI |
|---|---|---|---|
| Voltage (V) | - | - | - |
| Current (A) | - | - | - |
| Power (W) | 5.19 | 7.37 | 7.31 |
| Power usage | - | 2.18 | 2.12 |
| Speed (1K data size) (MB/S) | - | 110.07 | 931.52 |
| Energy per GB (J/GB) | - | 20 | 2.2 |

# 4 CONCLUSION

In this paper, we evaluate Advanced Encryption Standard New Instructions (AES-NI) from security, performance and power consumption perspectives. The experiments are performed using the Cipher Block Chaining (CBC) mode with 256 bytes as key length (AES-256-CBC). In security experiments, we measure the diffusion, confusion, and IV change. We also measure the randomness based on entropy and randomness online tester, Dieharder and NIST statistical test suite. Performance and power consumption are tested across multiple platforms. Our results show that AES-NI achieves much better performance and less power consumption than AES. In this paper, we present tools and techniques that can be used to evaluate and analyse different cryptographic techniques.

In Quantropi Inc., we have a private Randomness Transformation and Interpretation (RTI) technique that can be used to securely distribute keys between communicating pairs. The future work stemming from this paper is to evaluate the output randomness of the RTI using the comprehensive tools used in this paper.

## 6 ACKNOWLEDGMENTS

This research has been partially funded by Mitacs Canada.

## 7 REFERENCES

[1] National Institute of Standards and Technology. Advanced Encryption Standard (AES). *Federal Information Processing Standard 197*, online at http://csrc.nist.gov/publications/fips/fips197/fips197.pdf, (2001), last accessed on Dec. 10, 2019.

[2] Akdemir, K., Dixon, M., Feghali, W., Fay, P., Gopal, V., Guilford, J., Ozturk, E., Wolrich, G., and Zohar R. Breakthrough AES performance with Intel AES new instructions, online at https://software.intel.com/sites/default/files/m/d/4/1/d8/10TB24_Breakthrough_AES_Performance_with_Intel_AES_New_Instructions. final.secure.pdf, last accessed on Dec. 10, 2019.

[3] Stalling, W. Cryptography and network security principles and practices. Fourth Edition, *Prentice Hall* (November 2005).

[4] Gueron, S. Intel Advanced Encryption Standard (AES) new instructions set. online at https://software.intel.com/sites/default/files/article/165683/aes-wp-2012-09-22- v01.pdf, (2012), last accessed on Dec. 10, 2019.

[5] Xu, L. Securing the Enterprise with Intel AES-NI, online at https://www.intel.com/content/www/us/en/enterprise-security/enterprise- security-aes-ni-white-paper.html, (2010), last accessed on Dec. 10, 2019.

[6] Entropy and randomness online tester, online at https://servertest.online/entropy, (2019), last accessed on Dec. 10, 2019.

[7] Brown, R. G. Dieharder: a random number test suite, online at https://webhome.phy.duke.edu/~rgb/General/dieharder.php, last accessed on Dec. 10, 2019.

[8] Sýs, M., Říha, Z., and Matyáš V. Algorithm 970: optimizing the NIST statistical test suite and the berlekamp-massey algorithm, *ACM Transactions on Mathematical Software* 43, 3 (2017), 27-37.

[9] OpenSSL cryptography and SSL/TLS toolkit, online at https://www.openssl.org/, last accessed on Dec. 10, 2019.

[10] Powerstat, online at http://manpages.ubuntu.com/manpages/xenial/man8/powerstat.8.html, (2015), last accessed on Dec. 10, 2019.