```
================================================================
==============
🔐 VPR RESEARCH: HYBRID CRYPTO-VECTORIZED ENGINE
================================================================
==============
📊 Configuration:
   Target: 1000 primes
   Workers: 4
================================================================
==============

   [ 100/1000]  10% | Speed: 18.3 primes/sec | Found: 100
   [ 200/1000]  20% | Speed: 19.3 primes/sec | Found: 200
   [ 300/1000]  30% | Speed: 19.3 primes/sec | Found: 300
   [ 400/1000]  40% | Speed: 19.3 primes/sec | Found: 400
   [ 500/1000]  50% | Speed: 19.2 primes/sec | Found: 500
   [ 600/1000]  60% | Speed: 19.3 primes/sec | Found: 600
   [ 700/1000]  70% | Speed: 19.6 primes/sec | Found: 700
   [ 800/1000]  80% | Speed: 19.4 primes/sec | Found: 800
   [ 900/1000]  90% | Speed: 19.2 primes/sec | Found: 900
   [1000/1000] 100% | Speed: 19.0 primes/sec | Found: 1000

================================================================
==============
📊 FINAL HYBRID AUDIT REPORT
================================================================
==============
✅ Generated: 1000/1000 primes (Saved to primes.txt)
⏱️  Total Processing Time: 52.58s
⚡ True Average Speed: 19.0 primes/sec

🧬 Total gmpy2.is_prime() Calls: 64021
📉 Average Heavy Tests per Prime: 64.0
📁 Peak (Max) Memory Usage: 31.03 MB
================================================================
==============



================================================================
==============
🔐 VPR RESEARCH: HYBRID CRYPTO-VECTORIZED ENGINE
================================================================
==============
📊 Configuration:
   Target: 1000 primes
   Workers: 4
================================================================
==============

   [ 100/1000]  10% | Speed: 20.3 primes/sec | Found: 100
   [ 200/1000]  20% | Speed: 19.6 primes/sec | Found: 200
```

```
   [ 300/1000]  30% | Speed: 19.9 primes/sec | Found: 300
   [ 400/1000]  40% | Speed: 20.0 primes/sec | Found: 400
   [ 500/1000]  50% | Speed: 20.5 primes/sec | Found: 500
   [ 600/1000]  60% | Speed: 20.8 primes/sec | Found: 600
   [ 700/1000]  70% | Speed: 20.5 primes/sec | Found: 700
   [ 800/1000]  80% | Speed: 20.4 primes/sec | Found: 800
   [ 900/1000]  90% | Speed: 20.2 primes/sec | Found: 900
   [1000/1000] 100% | Speed: 20.3 primes/sec | Found: 1000

================================================================
==============
📊 FINAL HYBRID AUDIT REPORT
================================================================
==============
✅ Generated: 1000/1000 primes (Saved to primes.txt)
⏱️  Total Processing Time: 49.41s
⚡ True Average Speed: 20.2 primes/sec

🧬 Total gmpy2.is_prime() Calls: 61996
📉 Average Heavy Tests per Prime: 62.0
📦 Peak (Max) Memory Usage: 31.07 MB
================================================================
==============


================================================================
==============
🔬 BENCHMARK: VECTORIZED SPECTRAL vs STANDARD RANDOM APPROACH
================================================================
==============
📊 Configuration:
   Target: 1000 primes each | Length: 2048 bits | Workers: 4
================================================================
==============

⚡ Running VECTORIZED SPECTRAL method...
⚡ Running STANDARD random method...

================================================================
==============
📊 FINAL AUDITED BREAKDOWN
================================================================
==============
 SPECTRAL Speed: 20.3 primes/sec | Total Time: 49.35s
 STANDARD Speed: 12.4 primes/sec | Total Time: 80.43s
----------------------------------------------------------------
--------------
 🎯 VERDICT: SPECTRAL IS 1.63x FASTER!
 🧬 SPECTRAL ELIMINATED 80.7% OF Miller-Rabin CALLS!
================================================================
==============
```

```
================================================================================
🔒 INDEPENDENT CRYPTOGRAPHIC AUDIT (2026-07-08 20:34:38)
Analyzed objects: 1000 prime numbers | Expected length: 2048 bits
================================================================================

📊 PHASE 1: State Space Analysis
   Unique numbers: 1000/1000 (100.00%)
   Collisions (duplicates): 0 (0.00%)
   Local Min-Entropy: 9.9658 bits (Guesswork complexity)
   ℹ️  Note: This is log2(N) with 0 collisions — confirms only the absence
       of exact duplicates in the sample, does not measure the generator's true entropy.

🗜️ PHASE 2: Compressibility Test (Hidden Dependencies)
   Raw data size: 256,000 bytes
   Compressed size: 256,086 bytes
   Compression ratio: 100.03%

🧬 PHASE 3: NIST Statistical Bit Balance (Global Stream)
   Total bits analyzed: 2,048,000
   Distribution: Ones: 50.05% | Zeros: 49.95%
   NIST Monobit p-value: 0.1205020642
   ℹ️  Note: This is only 1 of 15 tests in NIST SP800-22 and operates on
       the concatenated stream — local positional biases may be masked.

📉 PHASE 4a: Autocorrelation (Global Stream, includes boundaries between numbers)
   Adjacent bit correlation: 50.06% (Expected: ~50%)

📉 PHASE 4b: Autocorrelation (per-number, lag-1)
   Average autocorrelation: 50.04% (Expected: ~50%)
   Range across numbers: 46.26% — 53.59%

📍 PHASE 5: Positional (Columnar) Bias Test
   Checking 2048 bit positions against 1000 samples
   Flagging threshold: outside 40%–60%

   ℹ️  Positions always '1' (expected for MSB by design): [0, 2047] (2 total)
   ✅ All (non-masked) positions within expected range 40%–60%

================================================================================
🎯 FINAL SECURITY ASSESSMENT
================================================================================
```

```
   Cryptographic tests passed: 6/6
   (includes the new positional test — this is the test most
likely
    to detect limited entropy space in specific bit positions)

   🟢 VERDICT: NUMBERS APPEAR CRYPTOGRAPHICALLY SECURE
   Passed all tests, including positional bit-level analysis.
================================================================
==============

ℹ️  This script is an independent diagnostic tool, not a substitute
for
   the full official NIST SP800-22 test suite on independent
samples, nor
   for batch-GCD checks between generated primes (recommended when
issuing
   many keys from the same generator).



================================================================
==============
🔬 BENCHMARK: VECTORIZED SPECTRAL vs STANDARD RANDOM APPROACH
================================================================
==============
📊 Configuration:
   Target: 1000 primes each | Length: 2048 bits | Workers: 4
================================================================
==============

⚡ Running VECTORIZED SPECTRAL method...
⚡ Running STANDARD random method...

================================================================
==============
📊 FINAL AUDITED BREAKDOWN
================================================================
==============
 SPECTRAL Speed: 21.0 primes/sec | Total Time: 47.64s
 STANDARD Speed: 11.7 primes/sec | Total Time: 85.57s
----------------------------------------------------------------
--------------
 🎯 VERDICT: SPECTRAL IS 1.80x FASTER!
 🧬 SPECTRAL ELIMINATED 82.4% OF Miller-Rabin CALLS!
================================================================
==============



================================================================
==============
🔐 VPR RESEARCH: HYBRID CRYPTO-VECTORIZED ENGINE
```

```
================================================================
==============
📊 Configuration:

   Workers: 4
================================================================
==============

   [ 100/1000]  10% | Speed: 20.8 primes/sec | Found: 100
   [ 200/1000]  20% | Speed: 19.6 primes/sec | Found: 200
   [ 300/1000]  30% | Speed: 20.0 primes/sec | Found: 300
   [ 400/1000]  40% | Speed: 20.9 primes/sec | Found: 400
   [ 500/1000]  50% | Speed: 20.7 primes/sec | Found: 500
   [ 600/1000]  60% | Speed: 20.7 primes/sec | Found: 600
   [ 700/1000]  70% | Speed: 20.6 primes/sec | Found: 700
   [ 800/1000]  80% | Speed: 20.6 primes/sec | Found: 800
   [ 900/1000]  90% | Speed: 20.6 primes/sec | Found: 900
   [1000/1000] 100% | Speed: 20.3 primes/sec | Found: 1000

================================================================
==============
📊 FINAL HYBRID AUDIT REPORT
================================================================
==============
✅ Generated: 1000/1000 primes (Saved to primes.txt)
⏱️  Total Processing Time: 49.23s
⚡ True Average Speed: 20.3 primes/sec

🧬 Total gmpy2.is_prime() Calls: 60925
📉 Average Heavy Tests per Prime: 60.9
💾 Peak (Max) Memory Usage: 31.07 MB
================================================================
==============



================================================================
==============
🔒 INDEPENDENT CRYPTOGRAPHIC AUDIT (2026-07-08 21:05:45)
Analyzed objects: 1000 prime numbers | Expected length: 2048 bits
================================================================
==============

📊 PHASE 1: State Space Analysis
   Unique numbers: 1000/1000 (100.00%)
   Collisions (duplicates): 0 (0.00%)
   Local Min-Entropy: 9.9658 bits (Guesswork complexity)
   ℹ️  Note: This is log2(N) with 0 collisions — confirms only the
absence
       of exact duplicates in the sample, does not measure the
generator's true entropy.

🗜️ PHASE 2: Compressibility Test (Hidden Dependencies)
```

```
   Raw data size: 256,000 bytes
   Compressed size: 256,086 bytes
   Compression ratio: 100.03%

🧬 PHASE 3: NIST Statistical Bit Balance (Global Stream)
   Total bits analyzed: 2,048,000
   Distribution: Ones: 50.05% | Zeros: 49.95%
   NIST Monobit p-value: 0.1815327296
   ℹ️  Note: This is only 1 of 15 tests in NIST SP800-22 and
operates on
       the concatenated stream — local positional biases may be
masked.

📉 PHASE 4a: Autocorrelation (Global Stream, includes boundaries
between numbers)
   Adjacent bit correlation: 50.06% (Expected: ~50%)

📉 PHASE 4b: Autocorrelation (per-number, lag-1)
   Average autocorrelation: 50.04% (Expected: ~50%)
   Range across numbers: 46.46% – 54.18%

📍 PHASE 5: Positional (Columnar) Bias Test
   Checking 2048 bit positions against 1000 samples
   Flagging threshold: outside 40%–60%

   ℹ️  Positions always '1' (expected for MSB by design): [0, 2047]
(2 total)
   ✅ All (non-masked) positions within expected range 40%–60%

================================================================
==============
🎯 FINAL SECURITY ASSESSMENT
================================================================
==============
   Cryptographic tests passed: 6/6
   (includes the new positional test — this is the test most
likely
    to detect limited entropy space in specific bit positions)

   🟢 VERDICT: NUMBERS APPEAR CRYPTOGRAPHICALLY SECURE
   Passed all tests, including positional bit-level analysis.
================================================================
==============

ℹ️  This script is an independent diagnostic tool, not a substitute
for
   the full official NIST SP800-22 test suite on independent
samples, nor
   for batch-GCD checks between generated primes (recommended when
issuing
   many keys from the same generator).
```

🔐 FAIR BENCHMARK: FULL RSA KEY GENERATION (2048 vs 4096)
==========
==========
n=1000 keys per test

Both methods generate COMPLETE RSA keys:
  • My method: p, q, n, φ(n), e, d, CRT (dP, dQ, qInv)
  • OpenSSL: Complete RSA key (openssl genrsa)

==========
==========
📊 TESTING: RSA-2048 (2×1024-bit primes)
==========
==========

PART 1: SINGLE-THREADED COMPARISON (2048-bit keys)

▶ My METHOD (2048) - SINGLE-THREADED (1 core, sequential)
  200/1000...
  400/1000...
  600/1000...
  800/1000...
  1000/1000...
  Average time/key: 0.0298s | 33.56 keys/sec

▶ OpenSSL (2048) - SINGLE-THREADED (1 core, sequential)
  200/1000...
  400/1000...
  600/1000...
  800/1000...
  1000/1000...
  Average time/key: 0.1654s | 6.04 keys/sec

PART 2: 4-CORE PARALLEL COMPARISON (2048-bit keys)

▶ My METHOD (2048) - PARALLEL (4 cores)
  200/1000...
  400/1000...
  600/1000...
  800/1000...
  1000/1000...
  Total time: 10.41s | 96.04 keys/sec

▶ OpenSSL (2048) - PARALLEL (4 cores)
  200/1000...
  400/1000...
  600/1000...
  800/1000...

1000/1000...
Total time: 55.40s | 18.05 keys/sec

==========
TESTING: RSA-4096 (2×2048-bit primes)
==========

PART 1: SINGLE-THREADED COMPARISON (4096-bit keys)

▶ My METHOD (4096) - SINGLE-THREADED (1 core, sequential)
200/1000...
400/1000...
600/1000...
800/1000...
1000/1000...
Average time/key: 0.2367s | 4.22 keys/sec

▶ OpenSSL (4096) - SINGLE-THREADED (1 core, sequential)
200/1000...
400/1000...
600/1000...
800/1000...
1000/1000...
Average time/key: 1.1176s | 0.89 keys/sec

PART 2: 4-CORE PARALLEL COMPARISON (4096-bit keys)

▶ My METHOD (4096) - PARALLEL (4 cores)
200/1000...
400/1000...
600/1000...
800/1000...
1000/1000...
Total time: 86.90s | 11.51 keys/sec

▶ OpenSSL (4096) - PARALLEL (4 cores)
200/1000...
400/1000...
600/1000...
800/1000...
1000/1000...
Total time: 351.06s | 2.85 keys/sec

==========
FINAL FAIR SUMMARY - COMPLETE RSA KEYS

==========
==========

RSA-2048 (2×1024-bit primes):

Scenario	My (keys/s)	OpenSSL (keys/s)	Ratio
Single-threaded (1 core)	33.56	6.04	5.55x
Parallel (4 cores)	96.04	18.05	5.32x

RSA-4096 (2×2048-bit primes):

Scenario	My (keys/s)	OpenSSL (keys/s)	Ratio
Single-threaded (1 core)	4.22	0.89	4.72x
Parallel (4 cores)	11.51	2.85	4.04x

==========
==========
📊 DETAILED COMPARISON: WHAT EACH METHOD COMPUTES
==========
==========

My Method (FULL RSA):
- p (prime, N-bits)
- q (prime, N-bits)
- n = p × q
- φ(n) = (p-1) × (q-1)
- e = 65537 (standard)
- d = e^{-1} mod φ(n) ← COMPUTED
- dP = d mod (p-1) ← COMPUTED
- dQ = d mod (q-1) ← COMPUTED
- qInv = q^{-1} mod p ← COMPUTED

OpenSSL (FULL RSA):
- p (prime, N-bits)
- q (prime, N-bits)
- n = p × q
- φ(n) = (p-1) × (q-1)
- e = 65537 (standard)
- d = e^{-1} mod φ(n)
- CRT parameters (dP, dQ, qInv)
- PEM encoding
- Key validation

Both methods compute the complete RSA mathematical key components (p, q, n, φ, d, CRT parameters).

==========
==========
📊 PERFORMANCE COMPARISON: RSA-2048 vs RSA-4096
==========
==========

Metric	RSA-2048	RSA-4096	2048/4096
My method (1-core)	33.56	4.22	7.94x
My method (4-core)	96.04	11.51	8.35x
OpenSSL (1-core)	6.04	0.89	6.76x
OpenSSL (4-core)	18.05	2.85	6.34x
My advantage (1-core)	5.55x	4.72x	1.18x
My advantage (4-core)	5.32x	4.04x	1.32x

==========

🔎 FINAL CONCLUSION

==========

✅ Both methods generate COMPLETE RSA keys
✅ Both include: p, q, n, φ(n), e, d, CRT parameters
✅ All computations are FORCED (no optimization)
✅ This is an APPLE-TO-APPLE comparison

Results Summary:

Metric	RSA-2048	RSA-4096	Advantage
My (1-core)	33.56 keys/s	4.22 keys/s	7.94x faster
My (4-core)	96.04 keys/s	11.51 keys/s	8.35x faster
OpenSSL (1-core)	6.04 keys/s	0.89 keys/s	6.76x faster
OpenSSL (4-core)	18.05 keys/s	2.85 keys/s	6.34x faster
My advantage (1-core)	5.55x	4.72x	
My advantage (4-core)	5.32x	4.04x	

✅ My method is faster than OpenSSL for BOTH key sizes:
- RSA-2048: 5.55x faster (1-core), 5.32x faster (4-core)
- RSA-4096: 4.72x faster (1-core), 4.04x faster (4-core)

✅ My method maintains significant advantage even for larger keys.

🔐 VPR RESEARCH: HYBRID CRYPTO-VECTORIZED ENGINE
==========
📊 Configuration:
   Target: 50000 primes
   Workers: 4
==========

   [5000/50000] 10% | Speed: 22.4 primes/sec | Found: 5000
   [10000/50000] 20% | Speed: 23.0 primes/sec | Found: 10000
   [15000/50000] 30% | Speed: 23.2 primes/sec | Found: 15000
   [20000/50000] 40% | Speed: 23.2 primes/sec | Found: 20000
   [25000/50000] 50% | Speed: 23.3 primes/sec | Found: 25000
   [30000/50000] 60% | Speed: 23.3 primes/sec | Found: 30000
   [35000/50000] 70% | Speed: 23.4 primes/sec | Found: 35000
   [40000/50000] 80% | Speed: 23.4 primes/sec | Found: 40000
   [45000/50000] 90% | Speed: 23.4 primes/sec | Found: 45000
   [50000/50000] 100% | Speed: 23.4 primes/sec | Found: 50000

==========
📊 FINAL HYBRID AUDIT REPORT
==========
✅ Generated: 50000/50000 primes (Saved to primes.txt)
⏱️ Total Processing Time: 2132.95s
⚡ True Average Speed: 23.4 primes/sec
🎯 Total Interval Coordinates Screened: 257,735,000
🧬 Total gmpy2.is_prime() Calls: 3135871
📉 Average Heavy Tests per Prime: 62.7
📁 Peak (Max) Memory Usage: 161.52 MB
==========

🔒 INDEPENDENT CRYPTOGRAPHIC AUDIT (2026-07-10 14:26:39)
Analyzed objects: 50000 prime numbers | Expected length: 2048 bits
==========

📊 PHASE 1: State Space Analysis
   Unique numbers: 50000/50000 (100.00%)
   Collisions (duplicates): 0 (0.00%)
   Local Min-Entropy: 15.6096 bits (Guesswork complexity)
   ℹ️ Note: This is log2(N) with 0 collisions — confirms only the absence
       of exact duplicates in the sample, does not measure the generator's true entropy.

🗜️ PHASE 2: Compressibility Test (Hidden Dependencies)

```
   Raw data size: 12,800,000 bytes
   Compressed size: 12,803,911 bytes
   Compression ratio: 100.03%

🧬 PHASE 3: NIST Statistical Bit Balance (Global Stream)
   Total bits analyzed: 102,300,000
   Distribution: Ones: 50.00% | Zeros: 50.00%
   NIST Monobit p-value: 0.3147501440
   ℹ️  Note: This is only 1 of 15 tests in NIST SP800-22 and
operates on
       the concatenated stream — local positional biases may be
masked.

📉 PHASE 4a: Autocorrelation (Global Stream, includes boundaries
between numbers)
   Adjacent bit correlation: 50.00% (Expected: ~50%)

📉 PHASE 4b: Autocorrelation (per-number, lag-1)
   Average autocorrelation: 50.00% (Expected: ~50%)
   Range across numbers: 45.68% – 54.86%

📍 PHASE 5: Positional (Columnar) Bias Test
   Checking 2048 bit positions against 50000 samples
   Flagging threshold: outside 40%–60%

   ℹ️  Positions always '1' (expected for MSB by design): [0, 2047]
(2 total)
   ✅ All (non-masked) positions within expected range 40%–60%

================================================================
==============
🎯 FINAL SECURITY ASSESSMENT
================================================================
==============
   Cryptographic tests passed: 6/6
   (includes the new positional test — this is the test most
likely
    to detect limited entropy space in specific bit positions)

   🟢 VERDICT: NUMBERS APPEAR CRYPTOGRAPHICALLY SECURE
   Passed all tests, including positional bit-level analysis.
```