Cryptography Report

Table of Contents

GITHUB PROJECT

Martin Harrigan

Munzer Elsarafandi
Date:25/11/2025

C00281934

CryptoBench Project Report

1. Introduction & Experimental Setup

This project evaluates the performance of several modern cryptographic algorithms using Python’s cryptography library.

Five categories of operations were benchmarked:
1. Key Pair Generation (RSA, DSA, ECC)
2. Symmetric Encryption & Decryption (AES-GCM, ChaCha20-Poly1305)
3. Asymmetric Encryption & Decryption (RSA-OAEP)
4. Digital Signing (RSA-PSS, DSA, ECDSA)
5. Signature Verification (RSA-PSS, DSA, ECDSA)

Each experiment was executed 10 times, and the first run was discarded to avoid warm-up bias. Results were exported as CSV files and visualized using Matplotlib.

Machine Specifications:
- Desktop PC
- AMD Ryzen 9 5900X CPU
- 32GB DDR4 4000MHz RAM
- NVIDIA RTX 2080 Super
- Windows 10

- Python version 3.12.6
- cryptography version 46.0.3

2. Key Pair Generation Results

Key generation time varies widely between RSA, DSA, and ECC.

Observations:

-RSA key generation becomes significantly slower at higher security levels.

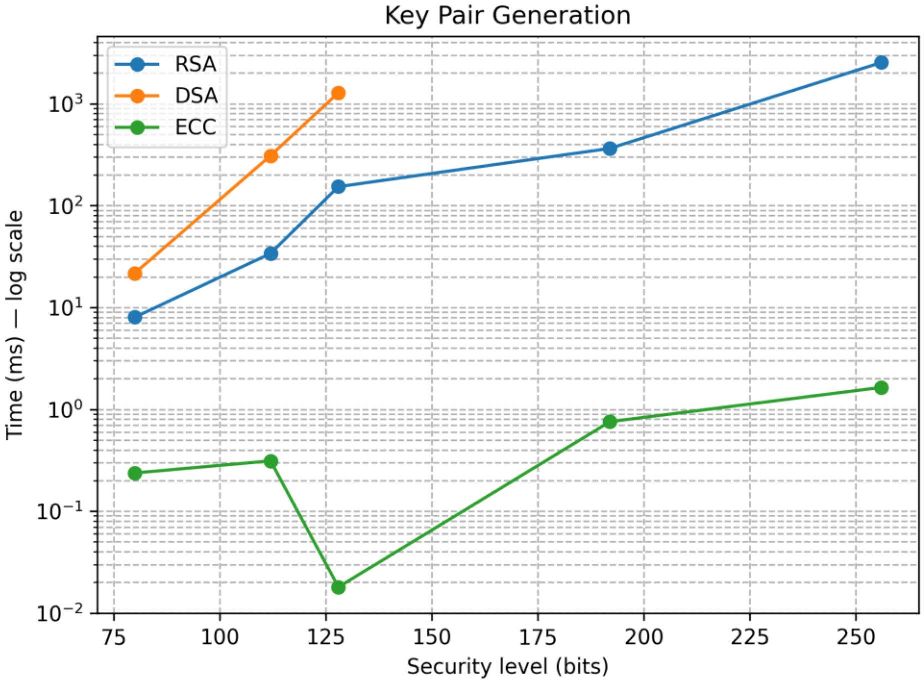
- 1024-bit keys are fast,
- but 4096-bit and 8192-bit keys take drastically longer.

-DSA behaves similarly and becomes very slow at 3072-bit.

-ECC is dramatically faster, generating keys in a fraction of the time, almost instantly compared to RSA/DSA.

Conclusion:

-ECC offers the best performance-to-security ratio.
-RSA and DSA become impractical as key sizes grow beyond 3072 bits.



3. Symmetric Encryption & Decryption Results

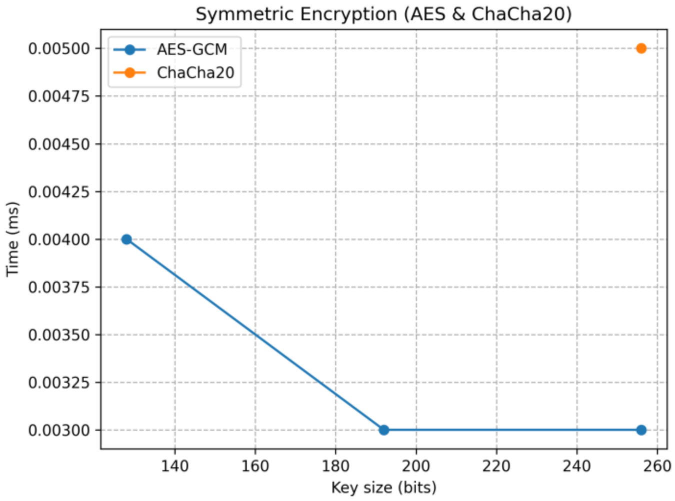
Both AES-GCM and ChaCha20-Poly1305 performed extremely fast at all key sizes.

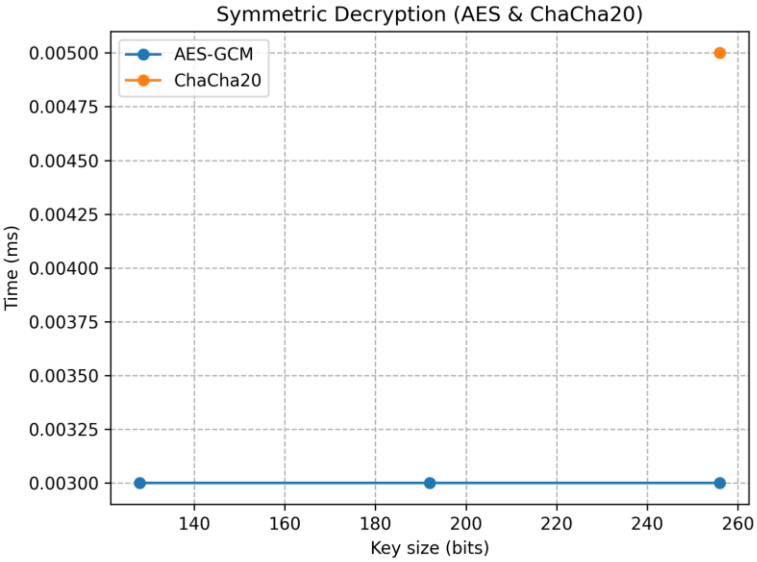
Observations:

- Both algorithms performed extremely fast (microseconds scale).
- AES-GCM was slightly faster at 128-bit and 192-bit.
- ChaCha20 showed stable performance but only supports 256-bit keys.
- Decryption times matched encryption results.

Conclusion:

-Symmetric algorithms are highly efficient and ideal for large data volumes.
-AES-GCM is ideal when hardware acceleration is available, while ChaCha20 is excellent on CPUs without AES instructions.





4. RSA-OAEP Encryption & Decryption Results

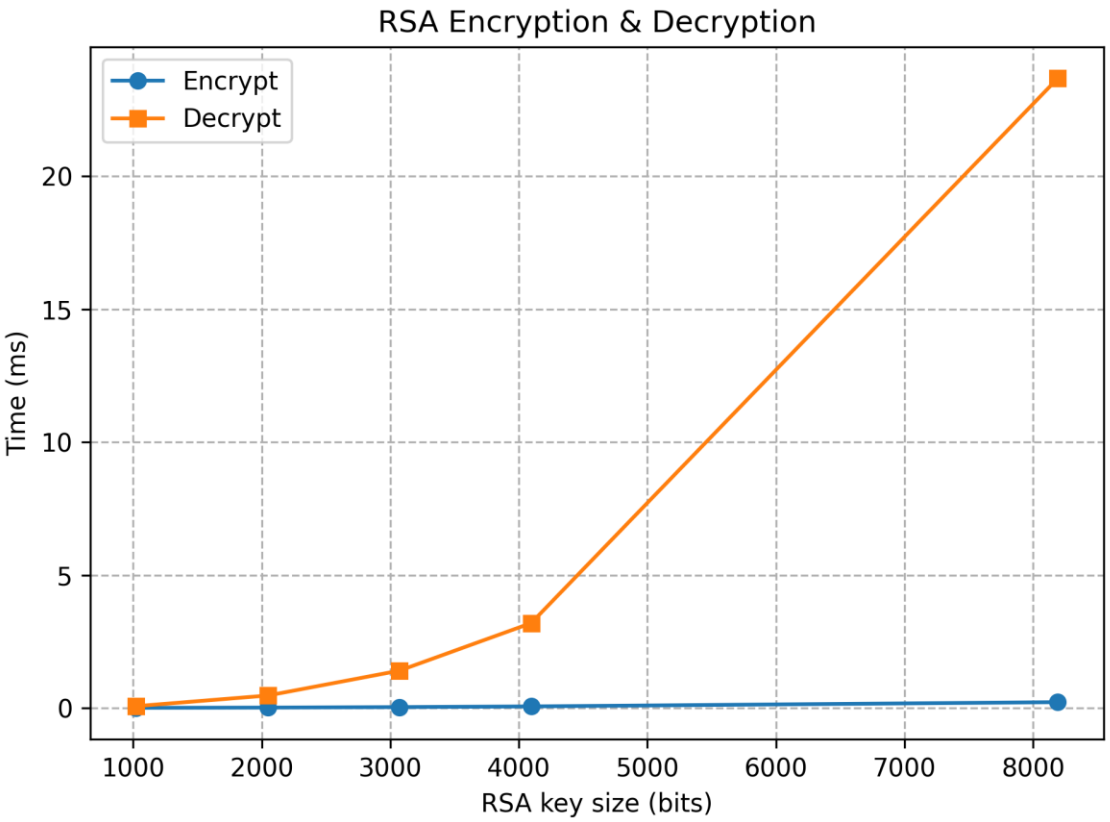
RSA encryption remained fast across all key sizes, but decryption slowed dramatically.

Observations:
- Encryption is fast across all key sizes, even up to 8192 bits.
-Decryption is slow, especially at high key sizes.

- Decryption cost increases non-linearly.
- 8192-bit RSA decryption becomes significantly expensive.

Conclusion:

RSA decryption is the bottleneck.
Large RSA keys significantly impact performance and are not practical for high-load systems.



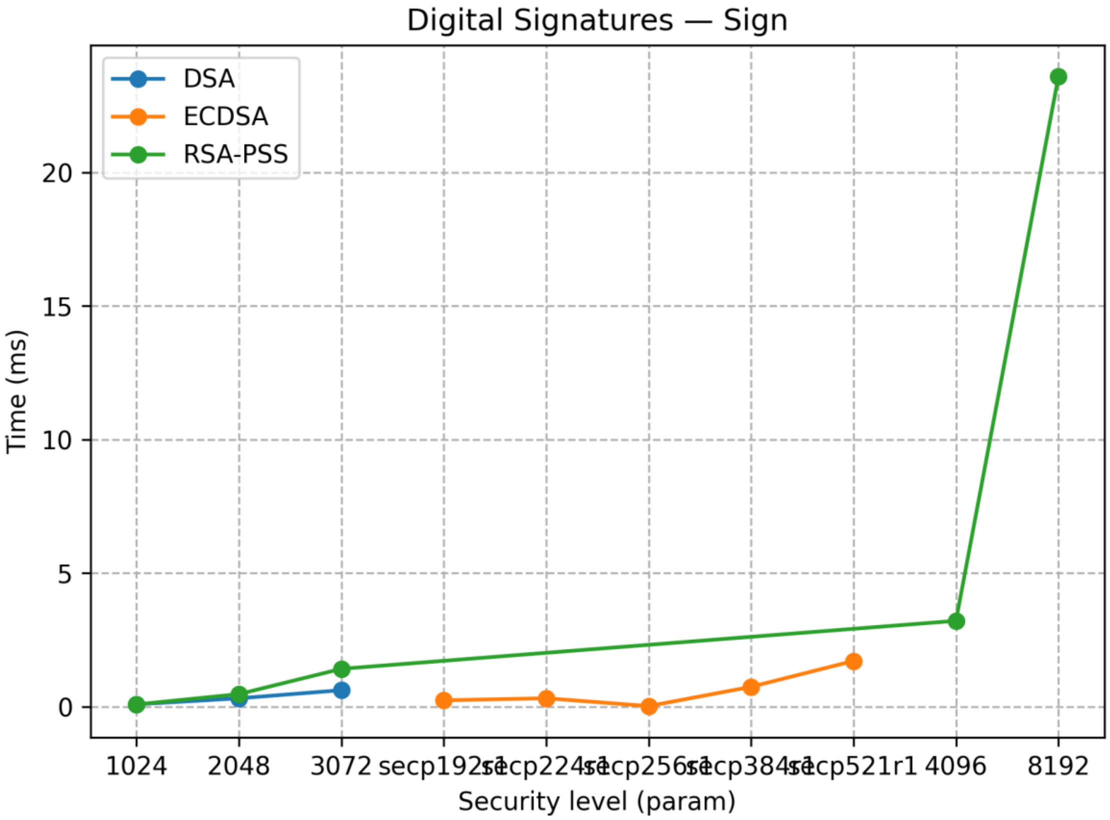
5. Digital Signing Results (RSA-PSS, DSA, ECDSA)

ECDSA was consistently the fastest signing algorithm.

Observations:
- ECDSA is the fastest signing algorithm across all security levels, outperformed RSA-PSS and DSA.
- RSA-PSS was the slowest, especially at 4096-bit and 8192-bit.
- DSA produced moderate speeds but was less efficient than ECC.

Conclusion:

ECDSA offers the best performance for signature generation, especially on modern hardware.



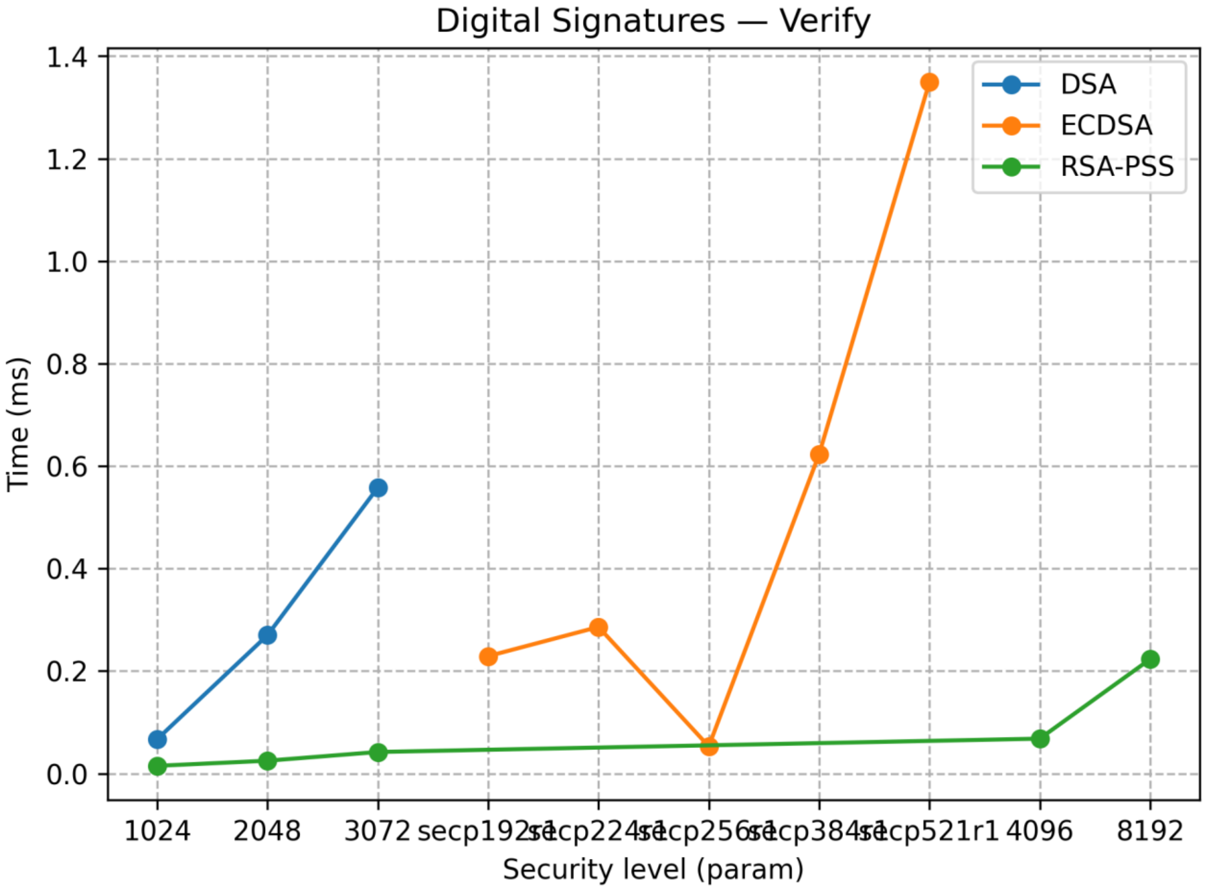
6. Signature Verification Results

Signature verification showed similar behavior to signing.

Observations:
- ECDSA verification is significantly faster than signing.
- RSA-PSS verification is slower at larger key sizes.
- DSA verification shows moderate performance but is less efficient than ECC.

Conclusion:

ECC again provides strong performance advantages.
RSA verification remains usable but expensive at large key sizes.



7. Overall Conclusion

The benchmarking results clearly show the differences between symmetric and asymmetric cryptography.

Summary:

- Symmetric algorithms (AES-GCM, ChaCha20) are extremely fast.
- Asymmetric encryption RSA decryption is expensive at large key sizes, computationally heavy.
- RSA-PSS produces high-overhead signatures, while ECDSA offers excellent performance and strong security.
- ECC is the most efficient algorithm family in key generation, signing, and verification.

No anomalies were observed.

All measured results agree with theoretical expectations and common cryptographic literature.