Crypto Engine on Raspberry Pi 5

System Understanding, Benchmarking and Demo

Group E
Jack, Jesse, Omar, Thu Ta

ECE 4301

Oct 8 2025

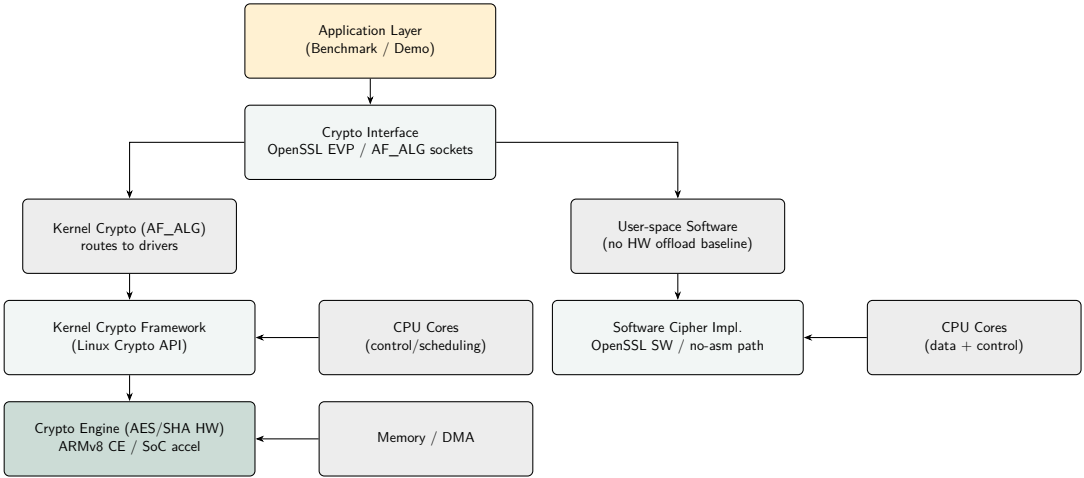
Agenda

1. System Overview
2. Benchmarking
3. Demo
4. References

What is a Crypto Engine?

- Dedicated hardware for AES, SHA, etc.
- Reduces CPU load and latency; boosts throughput.
- Exposed via Linux Crypto API (AF_ALG), OpenSSL EVP/engines, and `/proc/crypto`.
- Used in TLS/SSH, disk encryption, secure storage.

Crypto Engine System Overview



Two paths benchmarked: **AF_ALG (kernel→HW engine)** vs. **Software-only** (no HW offload).

Pi 5 Placement (Conceptual)

User Space

Application (AES file encrypt, TLS / benchmark)
OpenSSL EVP or AF_ALG sockets API

Kernel Space (HW path)

Linux Crypto API & AF_ALG → drivers (`aes-ce`, `sha256-ce`)

SoC / Hardware

ARMv8 Crypto Extensions / SoC accelerator (AES/SHA)
Cortex-A76 CPU coordinating with Memory/DMA

Benchmark Setup

Environment setup (Pi 5)

```
python3 -m venv venv
source venv/bin/activate
sudo apt install -y build-essential linux-headers-$(uname -r) \
  python3 python3-pip python3-matplotlib python3-pandas
pip3 install pandas matplotlib
```

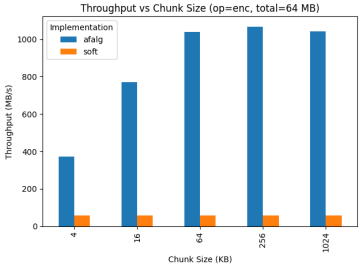
Run benchmark + plotting

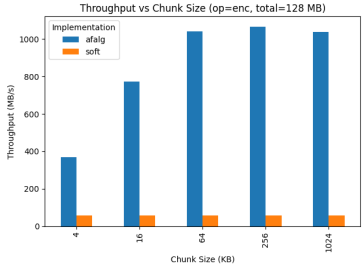
```
./run_bench.sh
python3 plot.py
```

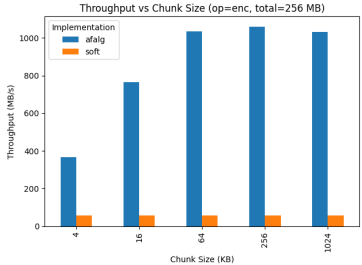
Method:

- Encrypts/decrypts random buffers (4 KB–1 MB chunks)
- Totals: 64 MB, 128 MB, 256 MB
- Metrics: Throughput (MB/s) and CPU time (ms)
- Saves results → `results.csv`, generates plots for throughput & efficiency

AES Encryption Throughput: `afalg` vs `soft`

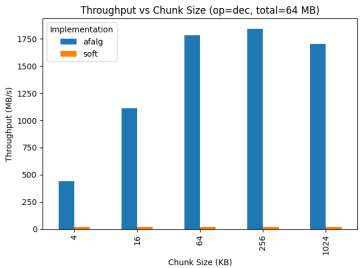


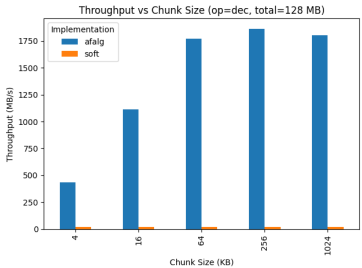


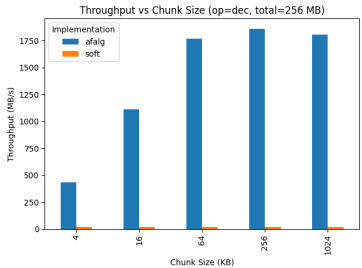


Observation: Hardware AES maintains 1–1.2 GB/s across sizes, while software AES peaks 50–70 MB/s. Throughput stabilizes for chunk 64 KB (syscall overhead negligible).

AES Decryption Throughput: `afalg` vs `soft`

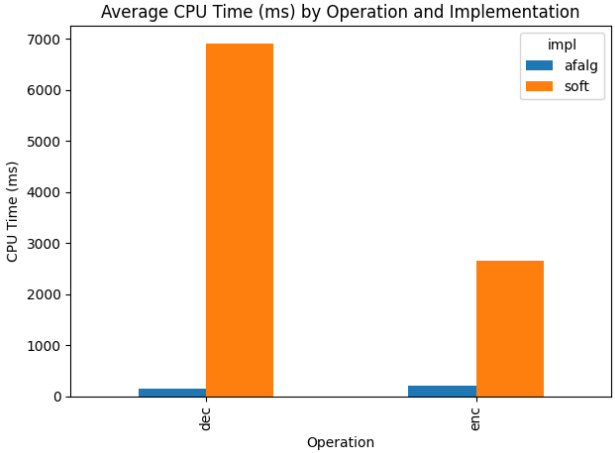






Observation: Decryption scales similarly, reaching 1.7–1.9 GB/s on hardware vs 60 MB/s in software. Chunk-size scaling trend identical to encryption.

CPU Time (ms): afalg vs soft



Software AES: 2.6 s (enc), 6.9 s (dec) Hardware AES: 0.2 s (enc), 0.15 s (dec) $\rightarrow$ **30–40× less CPU load with AF_ALG**. CPU time gap remains stable across all data totals.

Demo (Completed): Procedure

Steps on Pi 5:

```
python3 -m venv venv
source venv/bin/activate
sudo apt install -y build-essential linux-headers-$(uname -r) \
  python3 python3-pip python3-matplotlib python3-pandas
pip3 install pandas matplotlib
./run_bench.sh
python3 plot.py
```

Evaluation:

- Compared AF_ALG (kernel → HW engine) vs software AES-128-CBC.
- Tested encryption & decryption over 64–256 MB totals.
- Recorded throughput (MB/s) and CPU time (ms).
- Outputs saved to `results.csv` + plots.

Demo (Completed): Key Findings

- **Throughput:** HW AES 1–2 GB/s vs SW 50–70 MB/s.
- **CPU Efficiency:** HW AES 0.2 s (enc), 0.15 s (dec); SW AES 2.6 s (enc), 6.9 s (dec) $\rightarrow$ 30–40$\times$ less CPU load.
- **Scaling:** Performance stable across 64–256 MB; chunks 64 KB amortize syscall overhead.
- **Interpretation:** AF_ALG demonstrates efficient kernel-level offload; decryption slightly costlier due to IV chaining but $<$ 10

Summary

- The Raspberry Pi 5's ARMv8 Crypto Extensions deliver **order-of-magnitude improvements** in both speed and efficiency.
- Hardware AES sustains 1–2 GB/s throughput, while software is limited to 50–70 MB/s.
- AF_ALG reduces CPU time from seconds to hundreds of milliseconds (30–40$\times$ gain).
- Throughput plateaus beyond 64 KB chunks, indicating fully utilized hardware pipeline.
- Overall, Pi 5's AES engine is **validated for real-world crypto workloads** where low latency and CPU efficiency are critical.

References & Resources

- Raspberry Pi Docs — `raspberrypi.com/documentation`
- BCM2712 SoC Datasheet — `datasheets.raspberrypi.com`
- ARMv8 Crypto Extensions — `developer.arm.com`
- OpenSSL Docs — `openssl.org/docs`
- Linux Crypto API Guide — `kernel.org/doc/html/latest/crypto/`