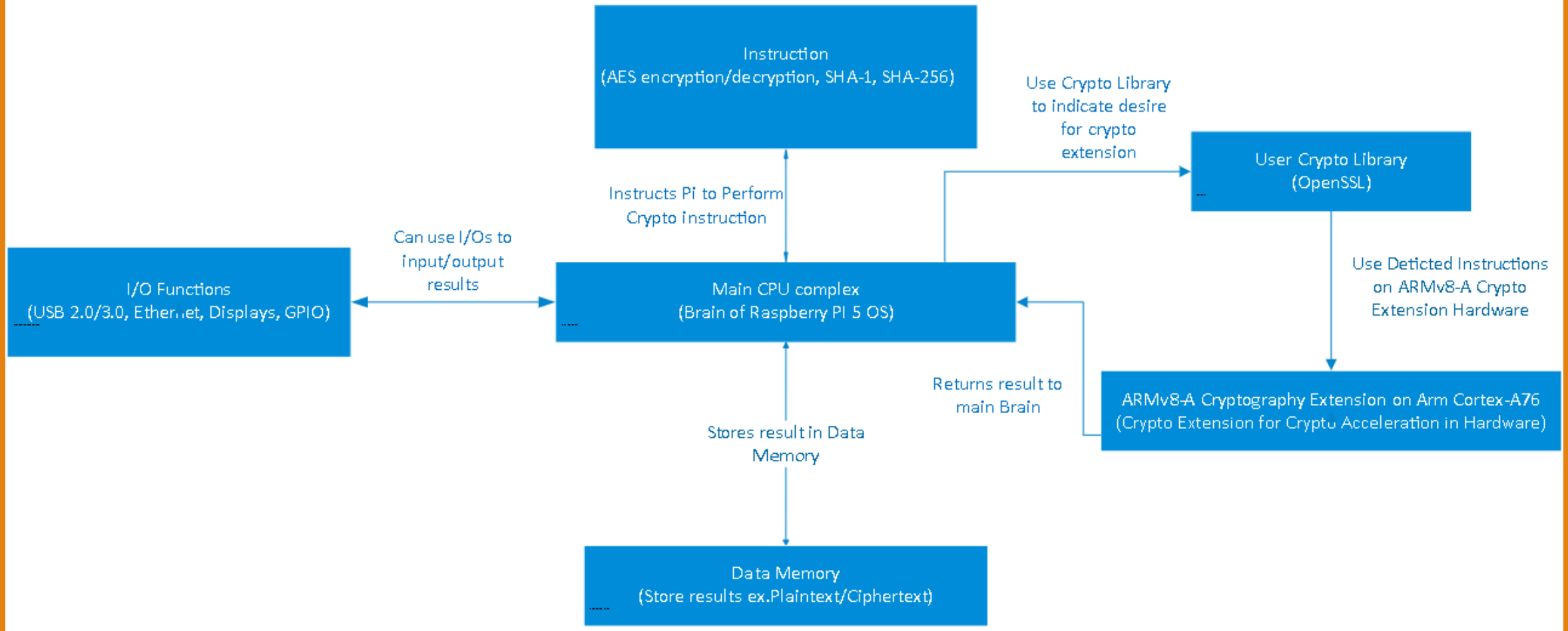
Raspberry PI 5 Crypto Engine

CREATED BY: AARON TRAN, MELVIN CONTRERAS, QUINN BELL

Basic Overview

- Physical Hardware block built into the System on Chip of the Raspberry PI 5 inside the ARM Cortex-A76 CPU on the PI's Broadcom BCM2712
- Utilizes ARMv8-A Cryptographic Extension on ARM Cortex-A76 CPU
- Processes cryptography in hardware crypto extension much faster than the CPU could do in software. The main OS is not involved beyond normal scheduling.
- OpenSSL libraries route the crypto instruction to the hardware engine so that the main OS doesn't have to perform the cryptography instruction.
- Capable of executing AES(Advanced Encryption Standard)/SHA(Secure Hash Algorithm) instructions on the ARM crypto extension
- Supports AES encryption/Decryption and SHA-1 and SHA-256

Raspberry Pi 5 Crypto Extension Block Diagram



Hardware Crypto Acceleration on Pi-5

- ARMv8-A Cryptographic Extension provides hardware acceleration for AES and SHA algorithms.
- Supports AES (128/192/256-bit keys) and SHA-1/SHA-256 operations directly in hardware.
- Reduces CPU load by offloading cryptographic computations to dedicated instructions.
- Enables secure and efficient encryption for SSL/TLS, VPN, and disk encryption.

Interface with OS and Libraries

- Integrated with Linux kernel crypto API for scheduling hardware-accelerated tasks.
- OpenSSL leverages ARMv8-A instructions for AES and SHA operations transparently.
- Applications using OpenSSL automatically benefit from hardware acceleration without code changes.
- Kernel-level drivers manage context switching between CPU and crypto engine.

AES Encryption/Decryption Flow on Pi-5

- AES operates on 128-bit blocks with key sizes of 128, 192, or 256 bits.
- Encryption steps: AddRoundKey → SubBytes → ShiftRows → MixColumns → AddRoundKey (repeated for 10/12/14 rounds).
- Pi-5 hardware executes these transformations using ARMv8-A crypto instructions for speed.
- Key schedule is precomputed and stored securely in memory; hardware applies round keys efficiently.
- Decryption uses inverse operations with the same hardware acceleration.