# Photon-Beetle Algorithm Using Raspberry Pi 5

Manuel Alvarado
Caleb Jala-Guinto
Isabel Warth

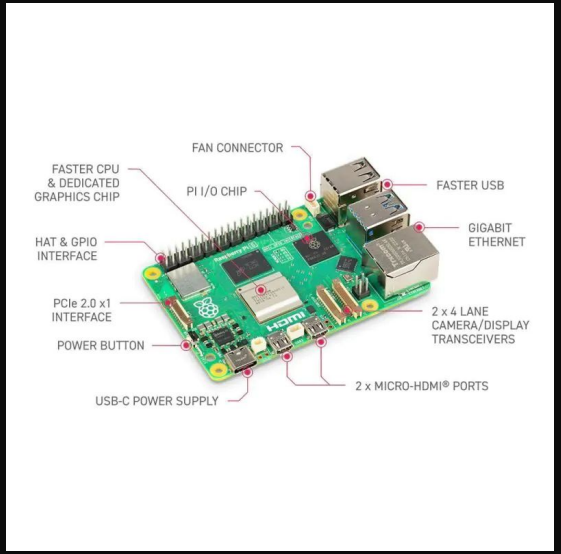
# Project Vision & Overview

**Vision:**

- Investigate and implement the **Photon-Beetle lightweight authenticated encryption algorithm** on the same embedded hardware architecture used in prior secure pipeline projects (e.g., Raspberry Pi 5).
- Evaluate **efficiency, energy, and security trade-offs** compared to standard ciphers like AES-128.
- Support future integration in **IoT and edge-computing** devices where low-power cryptography is essential.

**Key Objectives:**

- Implement Photon-Beetle AEAD (Authenticated Encryption with Associated Data).
- Benchmark runtime, energy, and memory footprint.
- Validate correctness against known test vectors.
- Document performance vs. AES baseline.



# Deliverables & Milestones

**Concrete Deliverables:**

- ✅ **Code:** Photon-Beetle encryption/decryption implementation in C/Rust.
- ✅ **Demo:** Live test or recorded demo showing message encryption/decryption.
- ✅ **Data:** Benchmark tables (execution time, power, memory).
- ✅ **Docs:** Technical report & GitHub README with build/run instructions.

| Week | Task | Output |
|---|---|---|
| **1** | Study Photon-Beetle paper & reference code | Understanding + design doc |
| **2** | Implement core algorithm | Working prototype |
| **3** | Integrate testbench + collect metrics | Benchmark data |
| **4** | Optimize, finalize report, create demo | Final deliverables |

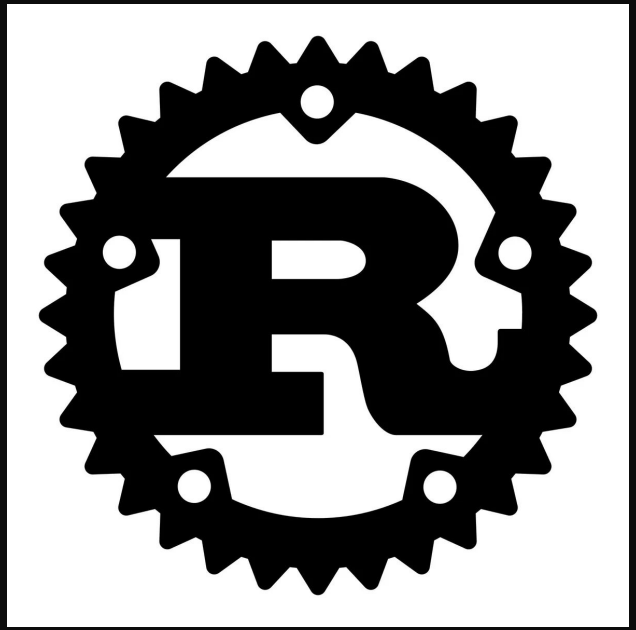
# Toolchain & Hardware Plan

**Hardware:**

- Raspberry Pi 5 (or equivalent ARM SBC)
- Optional: Power monitor (USB tester / INA219 sensor) for energy metrics

**Software Toolchain:**

- Language: **Rust**
- Compiler: rustc / gcc / clang
- Benchmarking tools: perf, time, custom timers
- Version control: **GitHub / GitLab**
- Documentation: **Markdown + LaTeX for final report**
- Optional: Python for plotting benchmark results



**Expected Outcome:**

- Efficient Photon-Beetle implementation suitable for low-power embedded systems, with reproducible performance data and clear documentation.