FYP Timeline

- **Phase 1: Get familiar with hardware (CM32M433R), prepare for thesis proposal**
  - Sept. 19 - Sept. 24
    - Install Nuclei Studio, finish GPIO lamping
    - Enable UART debugging
    - Read user manual 2.2 (register) and 16 (SAC module), mark critical registers' address of AES/OTP
    - Outcome
      - Screenshot of hardware environment tests
      - Notes of the specific address
  - Sept. 25 - Oct. 1
    - Literature supplement
      - A Survey on RISC-V Security Hardware and Architecture, Lu et al.
      - Security & Vulnerability: permission isolation failure, firmware tampering, side-channel attack, interface abuse
      - How to measure the experiment outcome: embedded system security benchmarking, evaluation, security-performance trade-off, side-channel resistance evaluation, RISC-V security metrics, etc.
    - Thesis proposal preparation
      - Background
      - Motivation and significance of the research
      - Home & Abroad research status and analysis
    - Outcome
      - A list of literature (2~3 official documents included)
      - Thesis proposal
  - Oct. 2 - Oct. 7
    - Thesis proposal finishing
      - Main research contents and research plans
      - Schedule and Expected Goals
      - Required Lab Conditions and Funds
      - Expected Difficulties, Technical Challenges and Solutions
    - PPT for thesis proposal
    - Outcome
      - Final version of thesis proposal
      - PPT for defense

- Oct. 8 - Oct. 9 (Buffer)
  - Troubleshooting, in case of abnormal hardware configuration
- **Phase 2: Core module development**
  - Oct. 10 - Oct. 28
    - Hardware AES development
      - Configure RCC to enable SAC clock (RCC_AHBPCLKEN register)
      - Configure SAC_AES_CTRL to CBC mode, read secret key from RNG_DATA register
      - Rate test in DMA mode
    - Software AES development
      - Transplant tiny-AES-c, run at SRAM (0x20000000)
    - Outcome
      - Hardware AES code with SAC register configuration
      - Hardware vs software encryption speed comparation
      - TRNG random number test report (entropy value)
  - Oct. 29 - Nov. 18
    - Hardware firmware verification
      - Use OTP to store basic value of SHA-256
      - Configure FLASH_ECC register for verification
    - Software firmware verification
      - Use Bootloader for block verification (every 2KB page, referring to the size of FLASH page), support version rollback
    - Outcome
      - Secure boot code (include OTP write-down)
      - Test video of firmware tampering (error reported by ECC)
  - Nov. 19 - Dec. 10
    - Hardware side-channel protection
      - Use TRNG to generate 8-bit mask (enable RNG_CTRL, then read from RNG_DATA)
      - Configure SAC_AES_DATA to xor the input data with the mask
    - Software side-channel protection
      - Use pseudo-random number as mask
    - Outcome
      - Mask code with TRNG invocation
      - Report on protecting effect (use oscilloscope to test power consumption fluctuation in no protection, hardware and software protection)
    - *Power consumption measurement depends on the device conditions. If there are no conditions, then I'll mainly rely on literature references and qualitative analysis.*

- Dec. 11 - Jan. 7
  - Hardware interface protection
    - Use PMP to configure SPI register (0x40013000) to read-only in M mode
    - Use GPIO to lock SPI pin (GPIOx_PLOCK_CFG register)
  - Software interface protection
    - Achieve SPI access whitelist (only allow the spi_transfer() invoke)
  - Outcome
    - Interface protection code with PMP configuration code
    - Attack test report (tampering SPI register in U mode)
- Jan. 8 - Jan. 28
  - System integration
    - Unify module interface (secure_mode = 0 (software) / 1 (hardware))
    - 100000 encryption stress test (monitoring FLASH ECC error rate)
    - Record resource usage
  - Outcome
    - The complete version of system firmware with mode switching logic
    - Stress test report
- Jan. 29 - Feb. 28
  - Interim report preparation
    - Organize the module documentation, mark the configuration basis for each register (like SAC_AES_KEY address)
    - Demonstration video, focus on PMP interception process and OTP verification process
  - Outcome
    - Interim report with register configuration log
    - Demonstration video

- **Phase 3: Thesis writing and defence**
  - Mar. 1 - Mar. 15
    - Interim report and rectification
      - Optimize PMP configuration, add L1-level read protection
      - Add negative tests (input malformed AES data, monitor SAC_ERR register)
    - Outcome
      - Interim report defence feedback record
      - Rectification report with FLASH_RDP configuration and modification log
  - Mar. 16 - Apr. 12
    - Thesis writing

        - System design : explain the working principle of SAC referring to Chapter 16 of user manual, attach the PMP area division diagram
        - Testing: use tables to compare hardware/software index (AES encryption rate, etc.), mark the data source
    - Outcome
        - Thesis draft
        - Performance comparison table
- Apr. 13 - Apr. 26
    - Thesis optimization
        - Conclusion: Highlight that "The hardware verification based on CM32M433R OTP + ECC is superior to the software solution"
        - Bibliography formatting: Add 2 official documents (China Mobile IoT. CM32M4xxR User Manual V1.4, 2022)
    - Outcome
        - Final version of thesis with originality declaration
        - List of references
- Apr. 27 - May. 5
    - Defense preparation
        - Defense PPT, highlight the advantages of the hardware solution in terms of rate and resources, and refer to the parameters from the data manual
        - Debugging demonstration environment (development board + test script, triggering PMP interception with one click)
    - Outcome
        - Thesis defense PPT
        - Result archive package (code, documents, videos)

以上内容整理于 幕布文档