Benchmarking of Lightweight Cryptographic Algorithms on MSP432P401R

Jay Raval

Ryan Carter

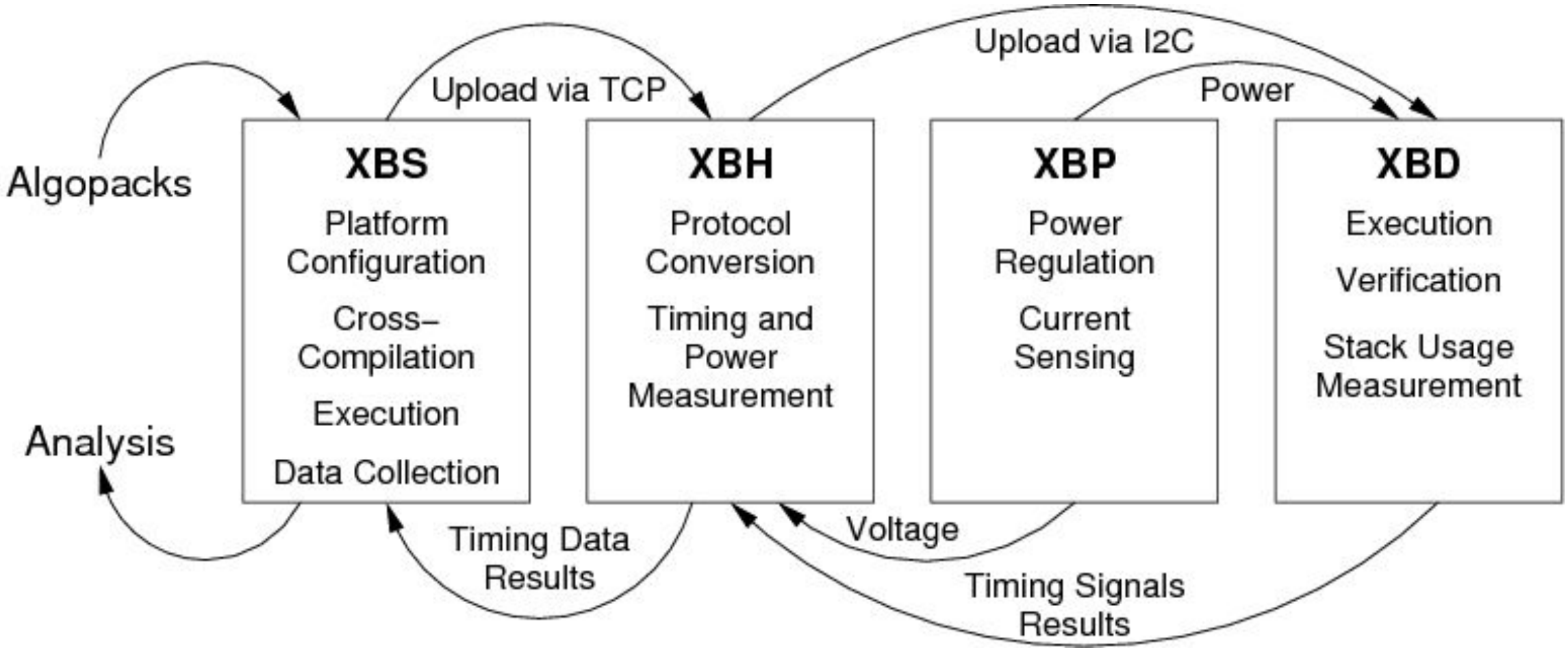
Snehal Patil

XXBX

- eXtended eXternal Benchmarking eXtension
  - Benchmarks AEAD algorithms from CAESAR competition on microcontrollers.
- It is an extension to the XBX platform which was used for benchmarking hashing algorithms for the SHA-3 competition.
  - Added power measurement and support for AEAD algorithms.
- Compatibility with SUPERCOP
  - System for unified performance evaluation related to cryptographic operations and primitives (SUPERCOP)

Motivation

- Growing market of IoT
- A portion of these devices require cryptographic software to protect sensitive data.
- Microcontrollers vary by memory restrictions, supported frequencies, architecture, etc.
- Need extendible framework to benchmark algorithms on microcontrollers


Upload via I2C
Upload via TCP
Power
Algopacks
XBS
Platform Configuration
Cross–Compilation
Execution
Data Collection
XBH
Protocol Conversion
Timing and Power Measurement
XBP
Power Regulation
Current Sensing
XBD
Execution
Verification
Stack Usage Measurement
Analysis
Voltage
Timing Data Results
Timing Signals Results

Components

- **XBS (XBX Benchmarking System)**
  - Python code that runs on Ubuntu PC
  - Commands XBH to benchmark ciphers and collect measurements
- **XBH (XBX Harness)**
  - EK-TM4C1294XL microcontroller based on ARM M4F
  - Facilitates communication between the XBS and XBD
  - 12-bit ADC for power measurement and timer for measuring the throughput
- **XBD (XBX Device under test)**
  - MSP432P401R microcontroller
  - XXBX evaluates ciphers on this device
- **XBP (XBX Power shim)**
  - Power is measured using a shunt resistor circuit.
  - XBH and XBD are connected through the XBP

Environment Setup Progress

- Each Team Member completed the following steps:
  - Installed Ubuntu 16.04.
  - Cloned the XBX repo and dependencies from GitHub.
  - Prepared build environment
- Challenges:
  - XBX setup uses crosstool to compile arm-tiva_c-eabi-gcc and other gnu tools
  - Crosstool fails to get required libraries
  - Workaround: removing tiva_c dependencies

XBH and XBS Progress

- Compiled XBH software and programmed microcontroller
  - Compile and install OpenOCD in the xbx/tools/openocd/directory
  - At the gdb prompt, connect to the openocd gdb server: target extended remote: 3333
  - Select the file for programming via gdb command: file build/xbh.axf
  - Program the device using gdb command “load”
  - Use “screen” to capture the debug output of the XBH
    - sudo screen /dev/ ttyACM0 115200
- Connection established between XBS and XBH

XBD Progress

- Two approaches taken simultaneously by team members:
  - Evaluate and understand working XBD software for MSP430F5529 microcontroller
  - Start developing XBD software for MSP432P401R microcontroller

XBD Progress Approach 1

- As test results section of the benchmarking platform consists of implementation of all the algorithms on gcc and their comparison with output generated from running the same algorithms on code composer studio using MSP430FF529. We will be implementing, compiling and debugging the algorithms in the Code Composer Studio IDE configured to work on MSP 432 environment.

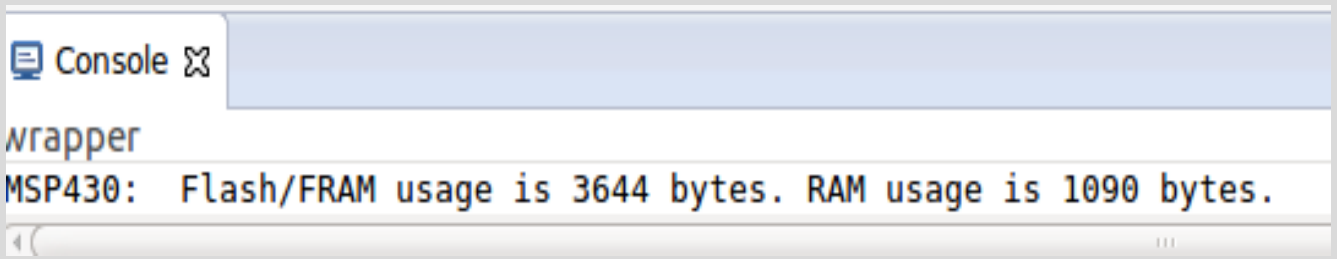
- Each of the algorithms from CAESAR Round 3 needs to be run on the MSP432 microcontroller with a generalized wrapper function built with some memory tweaks within the code to work with the limited memory capability of the MSP432.

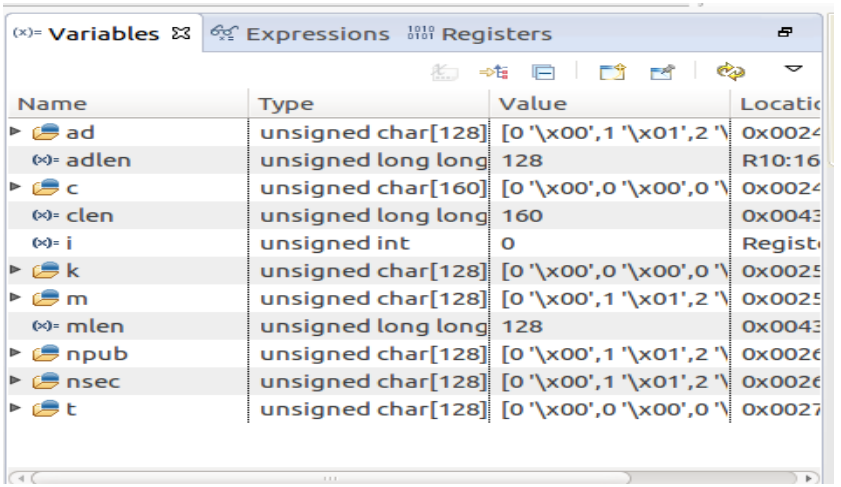
- The wrapper function calls of encrypt and decrypt of all algorithms use the same parameters according to the rules of CAESAR, a common wrapper function could be used to implement those functions.

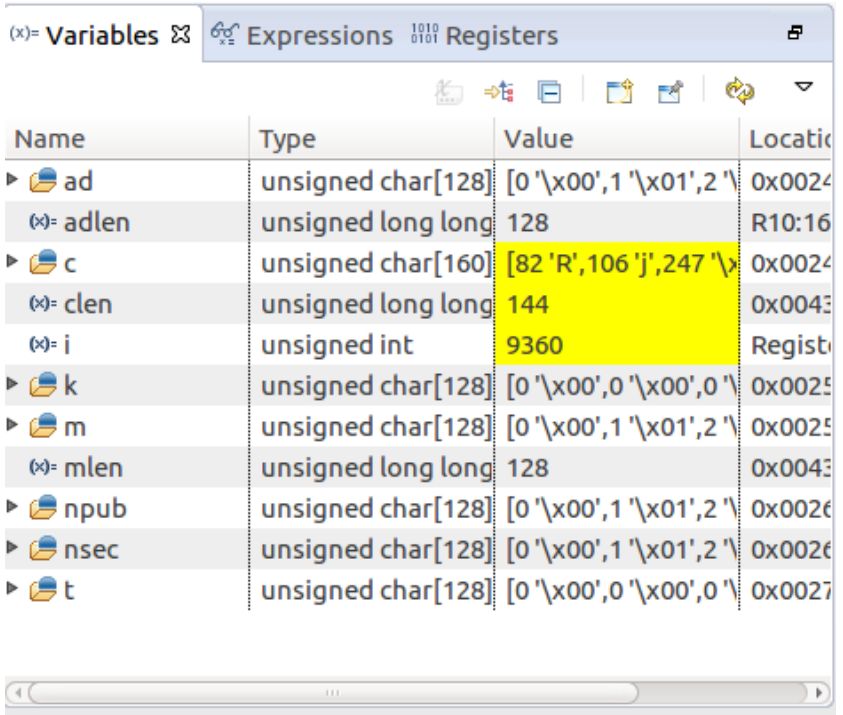
- The encryption and decryption function shall be called by a wrapper function that invokes the crypto_aead_encrypt() and crypto_aead_decrypt() function with the suitable arguments.
- Currently we are working on compiling the wrapper function and testing the round three algorithm on code composer studio.

Output of the CCS for Acorn

- The output of the Code Composer Studio while debugging the wrapper function for algorithm Acorn is as follows:
- M= plaintext c= cipher text.
- The RAM usage as shown in the console:







XBD Progress Approach 2

- Prepared build environment and Makefile
- Compiled test software and programmed the microcontroller successfully
  - Programming of device via Texas Instrument's (TI) gdb_agent_console
  - Start arm-none-eabi-gdb and connect to TI's gdb server

Remaining Work

- Implementation of CAESAR Round 3 candidates and debugging them using Code Composer Studio (CCS).
- Complete XBD transition to MSP432
- Incorporate MSP432 hardware AES encryption module