Benchmarking lightweight cryptographic algorithms using XXBX platform

Jay Raval
Matthew Carter
Snehal Patil
Need of Cryptography on low power devices

• With the current boom of IOT devices, there arises a question of how secure the devices are for malicious attacks.
• A section of these devices require cryptographic software to protect sensitive data.
• In recent years deployment of cryptography on lightweight low power consuming devices has emerged.
• Increase in key sizes has led to a tradeoff between efficiency and hardware cost.
What is XXBX?

• eXtended eXternal B enchmarking eXtension is a benchmarking tool for testing the AEAD algorithms that are competing in the CAESAR competition.

• It is an extension to the XBX platform which was used for benchmarking hashing algorithms for the SHA-3 competition.

• Power measurement and support for AEAD algorithms were the two limitations of the XBX platform which have been taken care of in the XXBX platform.

• Both XBX & XXBX are compatible with SUPERCOP’s algopacks.
XXBX Lineup

• XBS – I give commands and collect results!
• XBH - I help the XBX system communicate
• XBP – I amplify power
• XBD – I am the test subject (i.e. microcontroller)
XBS (XBX Benchmarking System) requirements

- Python 3
- Ubuntu 16.04.
- arm-none-eabi-gdb debugger
- SUPERCOP Primitives

What is SUPERCOP?

- System for Unified Performance Evaluation Related to Cryptographic Operations and Primitives
- SUPERCOP computes the performance of symmetric and asymmetric-key encryption systems and hash functions.
- More: https://bench.cr.yp.to/supercop.html
XBH (XBX Harness)

• The XBH supported is EK-TM4C1294XL based on ARM M4F. It is a microcontroller that facilitates communication between the XBS and XBD. It is termed “Harness” device since the various testing devices are connected on it.

• It uses its 12-bit ADC for power measurement and its timer for measuring the throughput.
XBD (XBX Device under test)

- It is the microcontroller under test which uses its bootloader to communicate with the XBH and the algorithm is implemented on it.
- The device to be tested is MSP432P401R.
- In the meanwhile, the device MSP430F5529 which has been implemented on the XXBX platform was reviewed and used to understand the entire setup and various commands such as:
  - `ls -la /dev/ttyACM0` and `cat /dev/ttyACM0`
XBP (XBX Power shim)

- Power is measured using a shunt resistor circuit.
- It is placed above the XBH and below the XBD.
XXBX in action..
Software Implementation

• The analysis section of the benchmarking platform consists of implementation of all the algorithms on the test device and their comparison with output generated from running the same algorithms on Code Composer Studio v7.

• 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 encryption and decryption of all algorithms use the same parameters according to the rules of CAESAR, a common wrapper function will be used to implement those functions.
Output of the CCS using for SIMON-JAMBU128/128

Plaintext

Encrypted

Decrypted
**Implemented algorithms**

- ACORN-128
- ASCON-128
- ASCON-128a
- twine80n6t4clocv3
- aes128n12t8silcv3
- aes128n12t8clocv3
- Present80n6t4silcv3
- Led80n6t4silcv3
- SIMON-JAMBU96/96
- SIMON-JAMBU64/96
- SIMON-JAMBU128/128
- NORX32-4-1
- NORX32-6-1

**Errors while implementing**

- Ketje Jr
- Ketje Sr
- Ketje Minor
I^2C COMMUNICATION

• Prepared build environment and Makefile
• Compiled XBD software and programmed the microcontroller successfully
  • Currently providing empty functions for some hardware specific features of XBD
• Built I^2C circuit and added software handling
I^{2}C
I²C

• Captured waveforms with Analog Discovery tool by Digilent

• 0x584244 – XBD
  • First part of XBH command structure
### I²C

<table>
<thead>
<tr>
<th>Name</th>
<th>Pin</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>I²C</td>
<td>DIO 0</td>
<td>Stop</td>
</tr>
<tr>
<td>Clock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>DIO 1</td>
<td></td>
</tr>
</tbody>
</table>

- **2048 samples at 4.761 MHz**
- **2017-11-06 21:15:21.540**

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.1e-07 Start</td>
</tr>
<tr>
<td>2</td>
<td>1.68e-06 h75 WR</td>
</tr>
<tr>
<td>3</td>
<td>2.688e-05 ACK</td>
</tr>
<tr>
<td>4</td>
<td>3.003e-05 h58</td>
</tr>
<tr>
<td>5</td>
<td>5.523e-05 ACK</td>
</tr>
<tr>
<td>6</td>
<td>5.838e-05 h42</td>
</tr>
<tr>
<td>7</td>
<td>9.744e-05 ACK</td>
</tr>
<tr>
<td>8</td>
<td>0.00010059 h44</td>
</tr>
<tr>
<td>9</td>
<td>0.0001911 ACK</td>
</tr>
<tr>
<td>10</td>
<td>0.00019425 h30</td>
</tr>
<tr>
<td>11</td>
<td>0.00028476 ACK</td>
</tr>
</tbody>
</table>

---

ECE 511 Microprocessors 17
Remaining Work

• Complete XBD transition to MSP432
  • Timing calibration
  • Flash programming
  • Cipher execution
• Incorporate MSP432 hardware AES encryption module.
• Also implement high performance and defense in depth types of algorithms from Caesar round 3.