**Research Task / Overview**

If they can’t see it, they can’t attack it...

- Investigate disaggregation of critical security algorithms into Asymmetric Multi-Processor System on a Chip (ASMP SoC)
- Determine ASMP SoC requirements
- Identify COTS ASMP SoC board(s) for proof-of-concept demonstration
- Develop research roadmap for future work

**Analysis of Prior Art**

Coprocessors have historically been used to improve performance

- Examples: I/O, floating-point, graphics, and cryptographic coprocessors
  - Ryan Cowart, David Coe, Jeffrey Kulick, and Aleksandar Milenkovic, “An Implementation and Experimental Evaluation of Hardware Accelerated Ciphers in All-Programmable SoCs” in ACM SE ’17: SouthEast Conference

Coprocessors are now employed to secure systems

- Trusted Platform Module (TPM) for secure generation/storage of cryptographic keys
- IBM 4758 crypto coprocessor added anti-tamper protections
  - “Extracting a 3DES key from an IBM 4758,” http://www.cl.cam.ac.uk/~rnc1/desrack/ibm4758.html
- Altera, Microsemi SoC Corp, and others investigated use of FPGAs to help secure the boot process
- DARPA System Security Integrated Through Hardware and Firmware Program (SSITH) seeks to mitigate common hardware vulnerabilities
  - Linton Salmon, “System Security Integrated Through Hardware and Firmware (SSITH),” DARPA Proposers Day Overview, April 21, 2017
  - 3+ year, $50M program
  - Participants required to use RISC-V soft-core processor
  - Restrictions on area and performance impacts increase by phase

**Goals & Objectives**

Objective

- Enhance the security of complex cyber-physical systems
- Our Strategy
  - Reduce the attack surface by deploying protection mechanisms into components that are not visible to the attacker

**Methodology**

- Use an Asymmetric Multi-Processor System on a Chip (ASMP SoC) to create regions of isolated, trusted hardware
- Disaggregate the most critical security algorithms from the system under attack and deploy them to the isolated trusted hardware
- Allow these most critical security algorithms to execute unimpeded by attacks launched against the protected system

**Future Research**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Approach</th>
<th>Threat Mitigated</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Passive memory inspection by ASMP</td>
<td>Memory corruption, code injection</td>
</tr>
<tr>
<td>2</td>
<td>Control-flow verification</td>
<td>Return-oriented programming</td>
</tr>
<tr>
<td>3</td>
<td>Introduction of trusted zones</td>
<td>Threats against OS-level abstractions</td>
</tr>
<tr>
<td>4</td>
<td>Hypervisor assisted ASMP</td>
<td>Threats against system-level abstractions</td>
</tr>
</tbody>
</table>

**Contacts/References**

Dr. David J. Coe (coed@uah.edu)
Dr. Jeffrey H. Kulick (kulickj@uah.edu)
Dr. Aleksandar Milenkovic (milenka@uah.edu)