

ROCHESTER INSTITUTE OF TECHNOLOGY

Course: CMPE-661
Hardware and Software Design for Cryptographic Applications

(Studio 3, Credit 3)

Course description (Bulletin):

The objective of this course is to introduce students to the subject of efficient implementations of cryptographic primitives using software and hardware approaches. The implementation platform will be an All Programmable System-on-Chip (AP SoC) Zynq device containing a general purpose processor and additional reconfigurable fabric for implementations of custom hardware accelerators. In the studio format, students will work on projects that require design of selected cryptographic primitives followed by comparison and contrast of various implementation alternatives, such as software, custom hardware, and hybrid hardware-software. Topics may include binary finite field arithmetic, block ciphers, hash functions, counter mode of operation for block ciphers, public key cryptosystems, hardware/software co-design methodologies with reconfigurable fabric, software development and profiling, high level synthesis, on-chip buses, hardware/software interfaces, and custom hardware accelerators.

To verify broader understanding of the material covered in this course, students are required to conduct a term-long research on published state-of-the-art technology on a course related topic. To achieve desired outcome, students will analyze available literature, prepare a written report, and give a presentation.

Prerequisite(s): Foundations of Digital System Design (CMPE-160/260), and C programming.

Classes:

Monday : 6.30pm- 7.45pm James E Gleason Hall (GLE)-3159

Wednesday : 6.30pm- 7.45pm James E Gleason Hall (GLE)-3159

The course is delivered in a studio lab format to reinforce student learning of the material with hands-on experience.

Instructors:

Michael Kurdziel; e-mail: mtkeec@rit.edu

Marcin Lukowiak; office (GLE) 09-3439; e-mail: mxleec@rit.edu

CE Linux system administrator:

Finetti Kounnavong <fxkeec@rit.edu> - contact him with any issues related to the computer account and software;

Office: (GLE) 09-3415

EE Facilities Manager:

Vincent Antonicelli <vaaeec@rit.edu> - contact him with any issues related to the card swipe lab access;

Office: (GLE) 09-3245

Office hours: Marcin Lukowiak

Wednesdays: 5.00pm - 6.00pm

TAs:

Matthew Krebs <mlk6450@g.rit.edu>

Payton Burak <psb4026@g.rit.edu>

Textbook(s) and/or required materials:

No textbook is required. On-line materials will be posted on mycourses.rit.edu.

References:

Understanding Cryptography: A Textbook for Students and Practitioners, C. Paar, J. Pelzl, Springer 2010, ISBN-978-3-642-04101-3 (free online access for RIT students via library)

The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable SOC, L. Crockett, R. Elliot, M. Enderwitz, R. Stewart, <http://www.zynqbook.com/>

FPGAs for Software Programmers, Editors: D. Koch, F. Hannig, D. Ziener, Springer 2016, ISBN 978-3-319-26408-0 (free online access for RIT students via library)

C language online reference, <https://en.cppreference.com/w/c>

NIST lightweight crypto, <https://csrc.nist.gov/Projects/lightweight-cryptography>

Additional references:

Cryptography Engineering, N. Ferguson, B. Schneier, and T. Kohno, John Wiley & Sons, 2010, ISBN-9780470474242

Cryptographic Engineering, C. Koc (Ed.), Springer, 2009, ISBN: 978-0-387-71816-3

Cryptographic Algorithms on Reconfigurable Hardware, F. Rodriguez-Henriquez, N.A. Saqib, A. Díaz Pérez, C. Koc, Springer, 2006, ISBN-13: 978-0387338835

Practical FPGA Programming in C, D. Pellerin and S. Thibault, Prentice Hall, 2005, ISBN-13: 978-0131543188

Course objectives:

1. To provide knowledge and understanding necessary for design and implementation of cryptographic primitives as software and using reconfigurable resources.
2. To provide knowledge and understanding of design methodologies and techniques required for all programmable SoC platforms.


Course outcomes:

1. Students have successfully customized and implemented an AP SoC based embedded processor system; students understand how to configure linker scripts and build software projects for cryptographic primitives.
2. Students know how to profile software applications to identify performance bottlenecks. Students have successfully optimized a software application to improve performance; students have analyzed cost in terms of the application size.
3. Students have successfully performed high level synthesis from C program to reconfigurable hardware; students have analyzed performance improvement in a hardware/software system.

Grade Weighting:

Assignments & tutorials	: 55%
Homework	: 15%
Research paper presentation	: 10%
Exams	: 20%

Tentative Topics:

	Monday	Wednesday	Deliverables/ Assignments timeline
Week 1 1/17-1/20		Course introduction, general hardware-software design flow for SoC devices, communication interfaces	
Week 2 1/23-1/27	Hands-on#1a: Using Xilinx Software Development Kit (SDK) and Vivado to create simple hardware and software for a Zynq device	Hands-on#1b: Profiling part 1, parallel processing, advanced software design flow with Xilinx Software Development Kit (SDK)	
Week 3 1/30-2/3	Overview of binary finite field arithmetic	Hands-on#2: Binary finite field arithmetic in software	Hands-on#1
Week 4 2/6-2/10	Hands-on#2: Binary finite field arithmetic in software	Crash overview of cryptography, security need, perfect secrecy, threat model, applications of private and public key cryptosystems	
Week 5 2/13-2/17	Crash overview of cryptography	Classical cryptography, stream cipher, block cipher	HW#1
Week 6 2/20-2/24	Classical cryptography, stream cipher, block cipher	The Advanced Encryption Standard (AES)	Hands-on#2
Week 7 2/27-3/3	The Advanced Encryption Standard (AES)	Hands-on#3: Profiling part 2, implementing the AES-128 block cipher in software; modifying the AES-128 code to improve performance, performance gain/cost analysis	HW#2
Week 8 3/6-3/10	Block cipher modes and authenticated encryption: ECB, CBC, OFB, AES-Galois/Counter Mode Term exam		Term exam
Spring break 3/13-3/17			
Week 9 3/20-3/24	Hash functions and their applications		
Week 10 3/27-3/31	Hands-on#4: AES-Galois/Counter Mode AES hardware architectures.		Research paper selection Hands-on#3
Week 11 4/3-4/7	Composite field S-box.		HW#3

Week 12 4/10-4/14	Introduction to hardware/software co-design with Vivado HLS. Using Vivado HLS to port the AES encryption algorithm to a hardware component to be run on the reconfigurable fabric of Zynq device Hands-on#5: Performance gain/hardware cost analysis, exploiting parallelism and pipelining at the software level, coding techniques for parallelism and pipelining		Hands-on#4
Week 13 4/17-4/21	Overview of public key cryptosystems and essential number theory		
Week 14 4/24-4/28	Diffie-Hellman, RSA, digital signatures, cryptosystems based on the discrete logarithm problem		Hands-on#5
Week 15 5/1	Overview of side channel attacks		HW#4
Exams week 5/3-5/10	Final exam Research paper presentations		

Students are responsible to observe announcements sent via email addresses used in <http://mycourses.rit.edu> and online grading/early alert system.

Hands-on reports

Upon completion of each Hands-on assignment, a complementary report is required (i.e. grade of zero will be given regardless of completeness of assignment without accompanying report). Reports are required to be submitted in **PDF ONLY** (no Microsoft word, no text files, etc...). How you get a PDF file is up to you (word or latex). But the submitted version **MUST BE PDF**. The filename of your report must be in the following form: **Assignment-#_abc1234.pdf** where the # is the assignment number and abc1234 is your RIT username (same as your email address). You reports will be submitted, graded, and returned to you with comments electronically. When submitting reports for a team of people, each student must author their own report. Meaning the text descriptions and explanations must be your own (NOT your teammate's). However, you can share (have the same) results, tables, pictures, diagrams, etc. (but the captions and descriptions must be unique). This policy may be updated as needed during the semester. You will be notified via email, with explicit change information if such changes are deemed necessary.

Academic Honesty: Although students are strongly encouraged to talk with each other and with the instructor to learn the course material, each student must individually complete work submitted for grading, including quizzes, exams, graded homework, and lab exercises. *No grading credit is earned for work not completed independently and completely by the individual student.* Copying assignments (including source code where various changes are made to make them “different”) will not be tolerated; all students involved in such copying will receive a grade of zero for copied assignments, regardless of who copied from whom.

RIT syllabus required policies

- **Academic integrity statement:** All conduct in this course is governed by the KGC OE Academic Honesty Policy, RIT Honor Code (P03.0), and RIT Student Academic Integrity Policy (D08.0).
- **Academic adjustments statement:** RIT is committed to providing academic adjustments to students with disabilities. If you would like to request academic adjustments such as testing modifications due to a disability, please contact the Disability Services Office (DSO). Contact information for the DSO and information about how to request adjustments can be found at <http://www.rit.edu/dso>. After you receive academic adjustment approval, it is imperative that you contact the instructor as early as possible to work out whatever arrangement is necessary.
- **Title IX statement:** Title IX violations are taken very seriously at RIT. RIT is committed to investigate complaints of sexual discrimination, sexual harassment, sexual assault, and other sexual misconduct, and to ensure that appropriate action is taken to stop the behavior, prevent its recurrence, and remedy its effects. Title IX rights and resources at RIT can be found at <http://www.rit.edu/fa/compliance/content/title-ix>.