ECE 5520 Secure Hardware Design
Spring 2017

Instructor:	Patrick Schaumont
Office:	Durham 337
E-mail:	schaum@vt.edu
Office Hours:	by appointment; please send me an email
Class Time:	TR 9:30A-10:45A
Class Location:	RAND 208
Class Website:	http://canvas.vt.edu

1 Objectives

Cryptography plays a fundamental role in the practical implementation of information security, needed in a vast range of applications from safe banking using electronic credit cards, over user privacy in medical record storage, up to protecting a nation state from electronic theft and cyber-attacks. With the Internet of Things, cryptographic implementations will be found over an ever wider range of computing architectures and performance budgets.

The central question addressed in this course is *how we can build efficient and secure cryptography in hardware?* One aspect of this study is the efficient implementation to meet the performance and cost requirements of computing platforms from embedded computing devices up to serverlevel computers. This includes the study of specialized architecture elements that form the trusted computing base. A second aspect is the analysis of implementation attacks, which are a particular concern when attackers have knowledge of, or access to the lowlevel implementation of cryptographic operations in computing devices.

Topics covered in the course include the implementation of Finite Field Arithmetic, Random Number Generation, Lightweight Cryptography, High-performance Cryptography, FPGA Security, Side Channel Analysis, Fault Analysis, Physical Unclonable Functions, and Secure Design Flows.

Having successfully completed this course, students will be able to:

- Describe hardware primitives for finite-field arithmetic.

- Explain the design concepts of cryptographic modules including block ciphers, stream ciphers, hash functions, and public-key ciphers.
- Design a hardware implementation for a given cryptographic function based on a high-level specification.
- Describe various methods for reverse engineering of secure hardware, and provide corresponding countermeasures for each of these techniques.
- Explain the principles of trusted computing.

2 Prerequisites

- ECE 4514 (Digital Design II). Students who have not taken this course have to contact the instructor before proceedings.
- Another useful course, not yet listed as an official prerequisite, is ECE 5560 (Fundamentals of Information Security).

3 Text and References

- The class material will come from several text books and research publications. There will be a source reference with each lecture. The key references (text books and seminal publications) will be available online.
- The proceedings of the yearly workshop on Cryptographic Hardware and Embedded Systems (CHES) carry state-of-the-art results relevant to secure hardware design. CHES was held each year from 1999.
  URL: http://dblp.uni-trier.de/db/conf/ches/index.html
- Journal publications and conference publications can be retrieved through several portals, depending on the publisher.
  - IEEExplore for papers published by IEEE.
    URL: http://ieeexplore.ieee.org/
  - ACM Portal for papers published by ACM.
    URL: http://portal.acm.org

- The IACR catalog for papers published by IACR (International Association for Cryptologic Research).
  URL: https://www.iacr.org/publications
- A crypto-oriented Springer proceedings catalog is maintained as well by J. Muir at Carleton. However, it was not updated after 2013. URL: http://www.ccsl.carleton.ca/ jamuir/crypto_springer.php
- LNCS for papers published by Springer.
  URL: http://www.springerlink.com/content/105633/

4 Course Work

As a student in *Secure Hardware Design*, you will be involved in research and design aspects of cryptographic hardware components. The format of the course is as follows.

- There will be a **regular lecture series** taught by the instructor to introduce major topics in secure hardware design.

  We will cover implementations for finite-field arithmetic, building blocks for symmetric-key and public-key cryptography, and random number generation. We will also discuss implementation attacks (passive and active) and countermeasures for them. Finally, we discuss platform implementation issues: optimizations for high-performance and 'lightweight' cryptography, and secure design flows.

- There will be a **student class presentation** series to present selected research papers on the topics discussed in the regular lecture series. The research papers will be assigned by the instructor and are listed further in the syllabus. All students (in groups of 2 students) will need to take part in a presentation of about 30 minutes. Both the presenters as well as the audience will be graded, in the following manner. At the start of each presentation, the instructor will randomly assign one or more students as presentation-evaluator(s). The student evaluator needs to prepare a written summary of the presentation, including an assessment of presentation style, clarity and overall quality. The evaluator will submit the report to the instructor and will be graded on the submitted report.

- There will be a **single research project**, implemented as a group effort. Each group will count at least 2 students. The assignment is a research problem in Cryptographic Engineering (specifications further in this report). The project will be implemented in two phases: (1) Problem Analysis and Solution Proposal; (2) Proof of Concept and Optimization. The objective of this research project is to approach and, if feasible, outperform state-of-the-art documented in contemporary literature.

  Each group will need to document the results for each phase in a report. The first report will be graded per student, and each student will need to submit a report. The second report will be graded for the entire group, and only one report per group is needed. The second report will be reviewed as a conference submission (ie. the instructors will look for external reviewers, who are experts in this field).

  Both reports represent a large portion of the grade for this course. The reports need to be of the highest quality. Do not underestimate the weight of this portion on your overall course grade.

- There will be a **final exam**, open book, at the end of the course. The exam is individually graded and will test your overall understanding of the course material. The exam is held on the official University exam date for this course.

5 Grading

Grades are assigned throughout the semester, based on the Course Work delivered during the course project, the presentations, and the class engagement. Your letter grade is determined by a mix of group work and individual effort.

Item Item	Graded per group	Graded per student	Grade % Grade %
In-class Presentation		X	15 %
Presentation Evaluation Report		X	15 %
Project Report 1		X	25 %
Project Report 2	X		25 %
Final Exam (Open Book)		X	20 %

6 Honor Code Policy

Adherence to the Virginia Tech Honor Code is expected in all phases of this course. Any work that you submit for a grade must be your own, or that of the group in case of the Project. In particular, proper citation of external source code material is essential.

See http://ghs.grads.vt.edu for information about the Graduate Honor System.

Abuses of the Honor Code Policy include for example cheating on Quizzes and plagiarism in presentations and project work. Such violations will be reported to the Office of the Honor System.

7 Special Needs

- Reasonable accommodations are available for students who have documentation of a disability from a qualified professional. Students should work through Services for Students with Disabilities (SSD) in 152 Henderson Hall. Any student with accommodations through the SSD Office should contact the instructor during the first two weeks of the semester.

- If participation in some part of this class conflicts with your observation of specific religious holidays during the semester, please contact the instructor during the first two weeks of class to make alternative arrangements.

- If you miss class due to illness, especially in the case of an exam or some deadline, see a professional in Schiffert Health Center. If deemed appropriate, documentation of your illness will be sent to the Deans Office for distribution to the instruction.

- If you experience a personal or family emergency that necessitates missing class, contact the Dean of Students at 231-3787 or see them in 152 Henderson Hall.

8 Tentative Schedule

Week	Date	Lecture	Topic	Presentation	Report
1	17 Jan	L	Introduction		
	19 Jan	L	Finite Field Arithmetic		
2	24 Jan	L	Modular Multiplication		
	26 Jan	L			
3	31 Jan	L	Random Number Generation		
	2 Feb	L			
4	7 Feb	L	Lightweight Crypto: Ciphers		
	9 Feb	L		P1	
5	16 Feb	L	Lightweight Crypto: Hash		
	18 Feb	L		P2	
6	23 Feb	L	High Performance Crypto		
	25 Feb	L		P3	
7	28 Feb	L	FPGA Security		R1
	2 Mar	L		P4	
8	7 Mar		Spring Break		
	9 Mar		Spring Break		
9	14 Mar	L	Side-Channel Analysis		
	16 Mar	L		P5	
10	21 Mar	L	Side-Channel Countermeasures		
	23 Mar	L		P6	
11	28 Mar	T	Guest Lecture (TBD)		
	30 Mar	T			
12	4 Apr	L	Fault Analysis		
	6 Apr	L		P7	
13	11 Apr	L	Fault Countermeasures		
	13 Apr	L		P8	R2
14	18 Apr	L	Physical Unclonable Functions		
	20 Apr	L			
15	25 Apr	L	Secure Design Flow		
	27 Apr	L			
16	2 May	T	TBD		

L lectures are *regular* lectures.
T lectures are lectures on travel days for the instructor.
P_i indicate student presentation schedule.
R_i indicate project report due dates.

9 Research Presentations

The following is the list of papers that will be used to drive the research presentations. The topic of each paper is in sync with the topic of the lectures at the moment of the presentation. Your team will select one paper, study it, and prepare a presentation for it. Presentation guidelines will be posted on the course website. Instructions to express your presentation preference will be provided by email.

P1: Lightweight Block Ciphers

- Julia Borghoff, Anne Canteaut, Tim Gneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Sren S. Thomsen, Tolga Yalin: *PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract.* ASIACRYPT 2012: 208-225

P2: Lightweight Hash Functions

- Jian Guo, Thomas Peyrin, Axel Poschmann: *The PHOTON Family of Lightweight Hash Functions.* CRYPTO 2011: 222-239

P3: High Performance Crypto

- Sanu Mathew, Farhana Sheikh, Michael E. Kounavis, Shay Gueron, Amit Agarwal, Steven Hsu, Himanshu Kaul, Mark Anders, Ram Krishnamurthy: *53 Gbps Native $GF(2^4)^2$ Composite-Field AES-Encrypt/Decrypt Accelerator for Content-Protection in 45 nm High-Performance Microprocessors.* J. Solid-State Circuits 46(4): 767-776 (2011)

P4: FPGA Security

- Stephen Trimberger, Jason Moore: *FPGA Security: Motivations, Features, and Applications.* Proceedings of the IEEE 102(8): 1248-1265 (2014)

P5: Side-Channel Analysis

- Amir Moradi, Tobias Schneider: *Improved Side-Channel Analysis Attacks on Xilinx Bitstream Encryption of 5, 6, and 7 Series.* COSADE 2016: 71-87

P6: Side-Channel Countermeasures

- Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, Vincent Rijmen: *A More Efficient AES Threshold Implementation.* AFRICACRYPT 2014: 267-284

P7: Fault Analysis

- Yang Li, Kazuo Sakiyama, Shigeto Gomisawa, Toshinori Fukunaga, Junko Takahashi, Kazuo Ohta: *Fault Sensitivity Analysis.* CHES 2010: 320-334

P8: Fault Countermeasures

- Akashi Satoh, Takeshi Sugawara, Naofumi Homma, Takafumi Aoki: *High-Performance Concurrent Error Detection Scheme for AES Hardware.* CHES 2008: 100-112

10 Research Project Topics

In this course, you will need to solve a research problem in Secure Hardware Design. The research problem will be broadly defined based on a research topic and one or more recent publications on this topic as a starting point. You will have several different topics to choose from. Your task will be to perform background research on the selected topic, to identify possible improvements, to define a plan to achieve those improvements, and to complete that plan by the end of the course.

To evaluate your progress, you will have to report two times on your research project. You will need to provide a written report before 5:00PM on 28 February, and on 13 April. Each of these reports will be graded separately. The instructor may invite you for a meeting to discuss further details. This research project accounts for 50% of your grade in this class. Thus, do not underestimate the weight of this project towards your overall grade!

The two phases of the problem are defined next.

Phase 1: Problem analysis. In this phase you will perform background research for a given research problem. This involves locating, reading, and

summarizing existing literature, identifying open issues in this research, identifying possible solutions to it, and defining a strategy to implement the solution. The majority of the proposed research topics involve hardware design. A minority of them will require low-level, architecture-specific software development or hardware analysis. Some projects requires hardware prototyping, and you will need to familiarize yourself with tools. The instructor may be able to help out with access to design tools and hardware; however, you are responsible for identifying the proper target platform.

Phase 2: Proof of concept. In the second phase you will develop a prototype implementation. For hardware-oriented designs, this will be a prototype based on FPGA or ASIC technology. FPGA design leads to actual prototypes, and ASIC will be based on technology mapping.

10.1 Getting Started

To complete the first phase of the project, you will proceed as follows.

1. Read through the topics below and select three which you would like to work on.

2. Find a team mate among the students of ECE 5520. You will be implementing this project in groups of maximal 2 students.

3. Send your team configuration and your three favorite topics to the instructor at schaum@vt.edu. Do this as soon as possible, but definitely before 3 February.

4. Read through the papers referenced in the description and engage in a critical discussion on this paper with your team mate. Read secondary papers, for example those listed as references in the provided paper. Look through the Literature Sources listed above to identify related work. In short, analyze and understand the research problem addressed by the provided paper.

5. The instructor will confirm the team and assign one topic to work on.

6. Identify an improvement, an enhancement, an alternative solution to the research problem addressed by the provided papers. Be original

and creative! Your improvement must involve hardware design unless this is explicitly ruled out in the topic description.

7. Write up your conclusions of this phase in a report. The format of the report must follow LNCS format.

8. Your report will be graded on completeness, correctness, accuracy, innovation, originality. Make sure to follow the rules of scientific writing: use proper figures, formatting, references (very important!), citations, and so on. You will receive your report grade, along with some feedback, two weeks after you have turned it in. The instructor may make arrangements with you to discuss your report during office hours; this will depend on the available time and the research topic. Such a discussion will serve to clarify your work and to provide further guidance.

10.2 Topics

The list of project topics, along with some initial pointers to get you started, is enumerated below. Feel free to email the instructor to exchange thoughts and to refine the project idea. With solid justification, it's possible to propose your own research topic as well. However, you will need to clearly argue the relevance of the proposed topic to this course.

As you will see from the list of papers, all of the related work is less than five years old. That means that your project needs to aim for state-of-the-art. While refining your project, you will need to ensure that your proposed solution is competitive to any existing solutions, and you will need to motivate its advantages over related work. Recall that the final project report will be reviewed as a conference paper.

Topic	Subject	Target
1	Low-Energy Cryptography	MCU, ASIC
2	IoT Lightweight Protocol	FPGA
3	Lightweight Entropy Extraction	FPGA, ASIC
4	True Random Number Generation Self-test	FPGA, ASIC
5	Hardware-supported Software Integrity	FPGA
6	Threshold Implementation of Block Cipher	FPGA, ASIC
7	Oblivious RAM Controller	FPGA

10.2.1 Low-Energy Cryptography

The objective of this topic is to study techniques that will minimize the *energy* consumption (not *power*) of a cryptographic protocol (such as for authentication). You can orient your implementation either to embedded software, or else to hardware. If you target embedded software, you will need to exploit the low-power mechanisms of an embedded micro-controller. If you target hardware, you will need to study ASIC design techniques that are amendable to low-energy dissipation.

References:

- Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, Francesco Regazzoni: *Midori: A Block Cipher for Low Energy.* ASIACRYPT (2) 2015: 411-436
- Pieter Maene, Ingrid Verbauwhede: *Single-Cycle Implementations of Block Ciphers.* LightSec 2015: 131-147
- Wenfeng Zhao, Yajun Ha, Massimo Alioto: *AES architectures for minimum-energy operation and silicon demonstration in 65nm with lowest energy per encryption.* ISCAS 2015: 2349-2352

10.2.2 IoT Lightweight Protocol

The Internet of Things will use constrained platforms with reduced computational and communication capabilities. You will evaluate the implementation cost of a communication protocol that could be used in lieu of a full-fledged internet security protocol such as TLS. You will consider cost factors such as area, latency, power consumption. The target is an all-hardware implementation (no software).

References:

- Mike Hamburg: *The STROBE protocol framework.* Cryptology ePrint Archive, Report 2017/003, 2017
- Michael Hutter, Jrgen Schilling, Peter Schwabe, Wolfgang Wieser: *NaCl's Crypto_box in Hardware.* CHES 2015: 81-101
- Markku-Juhani O. Saarinen: *Beyond Modes: Building a Secure Record Protocol from a Cryptographic Sponge Permutation.* CT-RSA 2014: 270-285

10.2.3 Lightweight Entropy Extraction

Physical Unclonable Functions (discussed in week 14) are used to extract stable cryptographic keys from physical objects and electronic components (such as SRAM cells). This requires additional postprocessing called entropy extraction. In current designs, these entropy extractors tend to be rather complicated and large (compared to the PUF circuit). In the project, you will study techniques to reduce the complexity and the implementation cost of these entropy extractors.

References:

- Roel Maes, Anthony Van Herrewege, Ingrid Verbauwhede: *PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator*. CHES 2012: 302-319
- Roel Maes, Vincent van der Leest, Erik van der Sluis, Frans Willems: *Secure Key Generation from Biased PUFs*. CHES 2015: 517-534
- Charles Herder, Meng-Day (Mandel) Yu, Farinaz Koushanfar, Srinivas Devadas: *Physical Unclonable Functions and Applications: A Tutorial.* Proceedings of the IEEE 102(8): 1126-1141 (2014)

10.2.4 True Random Number Generation Built-in Self Test

True Random Number Generators work by quantizing a physical noise source. To ensure high-quality random number streams, a random-number test can be used to analyze the statistical properties of the number stream. Typical test batteries such as NIST and DIEHARD are rather complex, and not suited for embedded hardware implementation. You will design a self-test engine for a TRNG that can indicate, with high confidence, if a TRNG is operating correctly or not.

References:

- Bohan Yang, Vladimir Rozic, Nele Mentens, Wim Dehaene, Ingrid Verbauwhede: *TOTAL: TRNG on-the-fly testing for attack detection using Lightweight hardware.* DATE 2016: 127-132
- Viktor Fischer, David Lubicz: *Embedded Evaluation of Randomness in Oscillator Based Elementary TRNG.* CHES 2014: 527-543

- Carlos Tokunaga, David Blaauw, Trevor N. Mudge: *True Random Number Generator with a Metastability-Based Quality Control.* ISSCC 2007: 404-611

10.2.5 Hardware-supported Software Integrity

Embedded software is subject to various tampering attacks (buffer overflow, fuzzing, etc). Two recent techniques - data flow integrity and control flow integrity - aim to verify the correctness of the software execution flow at runtime, using either hardware or else software techniques. In this project, you will study these software integrity techniques, and evaluate how to support them using hardware enhancements in a microprocessor. This project will start from an existing microprocessor, such as Z-80, MSP430, LEON3, RISC-V, .. and modify it so as to demonstrate the proposed software integrity technique. The recommendation is to focus initially on control flow integrity, but depending on the chosen route, data flow integrity can be studied as well.

References:

- Lucas Davi, Matthias Hanreich, Debayan Paul, Ahmad-Reza Sadeghi, Patrick Koeberl, Dean Sullivan, Orlando Arias, Yier Jin: *HAFIX: hardware-assisted flow integrity extension.* DAC 2015: 74:1-74:6

- Job Noorman, Pieter Agten, Wilfried Daniels, Raoul Strackx, Anthony Van Herrewege, Christophe Huygens, Bart Preneel, Ingrid Verbauwhede, Frank Piessens: *Sancus: Low-cost Trustworthy Extensible Networked Devices with a Zero-software Trusted Computing Base.* USENIX Security Symposium 2013: 479-494

- Nick Christoulakis, George Christou, Elias Athanasopoulos, Sotiris Ioannidis: *HCFI: Hardware-enforced Control-Flow Integrity.* CODASPY 2016: 38-49

10.2.6 Threshold Implementation for Block Cipher

Modern side-channel resistant implementations of block ciphers use so-called *threshold implementations*, which ensure that the side-channel leakage of an implementation is unrelated to the actual secrets being processed. In this project, you will construct and test a threshold implementation for a block cipher to be selected from the open literature of block ciphers. The threshold

implementation needs to be novel - i.e. you cannot re-implement an existing threshold design. You can, of course, study existing implementations for ideas.

References:

- Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, Vincent Rijmen: *A More Efficient AES Threshold Implementation.* AFRICACRYPT 2014: 267-284
- Aria Shahverdi, Mostafa Taha, Thomas Eisenbarth: *Silent Simon: A Threshold Implementation under 100 Slices.* IACR Cryptology ePrint Archive 2015: 172 (2015)
- Thomas De Cnudde, Svetla Nikova: *More Efficient Private Circuits II through Threshold Implementations.* FDTC 2016: 114-124

10.2.7 Oblivious RAM Controller

Oblivious RAM is an important concept for privacy friendly computing. The idea is that a microprocessor will organize its RAM acesses in such a way that the order and locations being accessed do not reveal anything on the activities of the internal program. While Oblivious RAM started out as a conceptual (theoretical) solution, recent progress has led to practical memory-access controllers. The objective of this project is to study the oblivious RAM concept in the context of a microcontroller, and build a prototype. You will analyze the cost and performance of ORAM on the selected micro-architecture.

References:

- Srinivas Devadas, Marten van Dijk, Christopher W. Fletcher, Ling Ren, Elaine Shi, Daniel Wichs: *Onion ORAM: A Constant Bandwidth Blowup Oblivious RAM.* TCC (A2) 2016: 145-174
- Christopher W. Fletcher, Ling Ren, Albert Kwon, Marten van Dijk, Emil Stefanov, Dimitrios N. Serpanos, Srinivas Devadas: *A Low-Latency, Low-Area Hardware Oblivious RAM Controller.* FCCM 2015: 215-222
- Emil Stefanov, Marten van Dijk, Elaine Shi, Christopher W. Fletcher, Ling Ren, Xiangyao Yu, Srinivas Devadas: *Path ORAM: an extremely simple oblivious RAM protocol.* ACM Conference on Computer and Communications Security 2013: 299-310