Wireless Cybersecurity Education via a Software Defined Radio Laboratory

Cem Sahin*, Danh Nguyen*, James Chacko*, and Kapil R. Dandekar*
*Drexel Wireless Systems Lab, ECE Department, Drexel University
Emails: {cs486, dhn24, jjc652, dandekar}@drexel.edu

***Abstract*—Cybersecurity is one of the fastest growing concerns in the world today. Recent global events show the need for more strict measures to protect the public from cyber attacks, which has triggered an increased demand for cybersecurity professionals. Many academic institutions began offering courses covering current cybersecurity concepts to satisfy this need. Although these courses educate students with proper skills, they lack the connection to current advancements in academic research. A more encompassing curriculum is needed in this rapidly growing field. For the most up-to-date education, we developed a course that takes a student-centric hands-on approach supported with Software Defined Radios to study current cybersecurity research projects in wireless networks. Our course consists of short lectures followed by lab sessions, where students implement security algorithms described by standards or by recent peer-reviewed research articles. Our results indicate that students appreciate the mixture of textbook and research topics being covered in the course and they feel more prepared for any future task in the cybersecurity field. While deviating from the textbook applies additional strain to educators, our paper shows that including current research practices in curriculum development efforts is a good investment towards a better educational outcome.**

Keywords*—*engineering education; curriculum development; data security

I. Introduction

The amount of data that is being transmitted over the Internet is growing rapidly every day. The modern lifestyle is shifting users towards having multiple devices connected to each other and the Internet, which comes with the natural tendency to complete daily tasks online and share more information on social media sites. According to Youtube, one of the most popular video streaming services, "300 hours of video are uploaded every minute" into their servers [1]. As more of our daily lives shift from being face-to-face transactions to online encounters, the vulnerabilities of the Internet are being realized. Global news shows that the information put into the Internet is not as secure as it should be.

With the increased need for Cybersecurity, the demand for professionals in this field is increasing. Many academic institutions started offering courses, certificates, and now even majors in Cybersecurity [2]. We have also started seeing initiatives from government agencies and private companies to raise awareness and attract more interest in Cybersecurity [3]–[5]. PBS NOVA Labs is offering an educational digital game, which introduces Cybersecurity basics, that can be played from any computer with an Internet connection [6].

In addition to offering students more hands-on modules, educators continued to look for additional tools that will support these labs and projects. There has been a growing interest in integrating Software Defined Radios (SDR) with hands-on course modules because of their flexibility. Initially, signal processing courses started taking advantage of them [7] for running labs covering topics such as modulation, encoding, and filtering. As their popularity increased, platforms such as GNURadio Companion [8], and OSSIE [9] emerged to offer visual development tools. These tools allowed students to create a signal, process it, and send it via the SDR. However, they lack key details pertaining to the hardware implementation aspects of SDRs. On the other hand, Computer Engineering and Software Engineering courses focus more on the hardware implementation portion and do not emphasize the signal processing capabilities of SDRs [10]. In this paper, we merge the gap between the SDR hardware implementation, signal processing and radio capabilities, and discuss a multi-disciplinary course that leverages SDRs to introduce wireless cybersecurity to students.

In order to provide up-to-date materials to our students, we developed a student-centric, SDR based wireless security course, where we support traditional textbooks with current research practices. Some of the

978-1-4799-8454-1/15/$31.00 ©2015 IEEE

topics covered in our course are packet detection [11], eavesdropping and man-in-the-middle attacks [12], and encryption techniques including Physical-layer security [13]. During the labs, students are encouraged to implement various attack mitigation techniques on SDRs, which requires them to use hardware design techniques, along with signal processing concepts, and to comment on their findings. Grading is done by checking the validity and strength of their solution.

In [14]–[16], the authors discuss the benefits of offering a student-centric learning experience, which is especially prefered when the course is related to technology. In the recent years, this approach has been augmented by several others to include competition between students [17]–[19]. Our course tries to combine the benefits of both student-centric and game-based learning styles. We developed our labs to give our students as much experience as possible with the current SDR and security topics. We also created a wireless *hacking* competition as their final project.

In this paper, we discuss the use of SDRs in teaching wireless cybersecurity and the effectiveness of bringing research into the classroom in terms of student acceptance and success. We organize this paper as follows: In the next section, our course logistics and methodology is discussed. Section III then goes into details of the specific course modules we developed. In Section IV, we present a case study and investigate the specifics of one of the lab modules. Section V provides a summary of the course outcomes. Our end of term student evaluation results are summarized in Section VI. Then, we conclude this paper in Section VII.

II. Course Design and Methodology

A. Student Demographics

Since our SDR security course was being offered for the first time, a small pilot section was created, where 9 graduate students were enrolled. The decision to limit enrollment to this number was based on available computer resources and classroom space. All students were enrolled in a Masters of Science program within the Electrical and Computer Engineering Department at Drexel University. Out of the 9 students, there was only one female student. One of the students was pursuing his education while also working full-time at a company. No country of origin or race information was collected for the purposes of this paper.

B. Course Logistics and Flow

Our SDR based security course is developed and offered weekly at the graduate-level. We were assigned a 3 hour meeting period. Our classroom was a Digital Signal Processing (DSP) laboratory with built-in equipment racks and available computers for student use. Each work area was designed to accommodate 2 students. There were 4 work areas in total. Desktop computers were preinstalled with the Ubuntu operating system before the beginning of the course. Students were then guided to install any additional programs and/or packages as the course progressed. We ran the course by splitting the students into 3 groups of 2 and a group of 3. Although students had the freedom to choose who to be partners with, we encouraged them to form groups to include at least one person knowledgable in Linux programming.

Drexel University offers a term based system, where each term consists of 10 weeks of in-class sessions followed by a week of final examinations. Table I shows a breakdown of the 10-week term by labs. A similar table was also given to the students as a part of the course syllabus. The table also reminds students of important dates, such as the course drop deadline. Each lab was originally designed to last for two weeks. Small adjustments were made as the term progressed.

TABLE I: Breakdown of the term by lectures and labs

Week	Lectures	Labs	Homework
1	Lecture 1	Lab 1	-
2	Lecture 2	Lab 1	-
-	Course drop deadline	-	-
3	Lecture 3	Lab 2	HW 1
4	Lecture 4	Lab 2	-
5	Lecture 5	Lab 3	HW 2
6	Lecture 6	Lab 4	HW 3
-	Course withdraw deadline	-	-
7	Lecture 7	Lab 4	-
8	Lecture 8	Lab 5	HW 4
9	Lecture 9	Lab 5 / Project	-
10	Lecture 10	Final Project	HW 5
11	Final Project due	Final Competition	-

The course was run as a series of short (usually 30-40 minutes), theory-based lectures followed by the student-centric lab assignment. The lab assignments were designed to leverage the concepts covered in the lecture portion but also to invite the students to indulge into research papers to complete their lab and the homework assignment at the end of the lab directions. The lab assignments were expected to be completed in-class.

Additional homework assignments were given at the end of each lab. Each homework was due before the beginning of the next lab. Students were provided with USRP N210 SDR platforms to work on their lab and homework assignments. They were also given remote desktop access to lab computers for when they would want to have access to the experiment setup with SDRs. Students were not allowed to remove the SDR nodes from the classroom. Throughout the term, two unannounced quizzes were given. These quizzes were mainly focusing on the theory covered in the lecture component of the course. Although a textbook was listed in the syllabus, most of the course material was custom prepared and distributed via the Blackboard Learn course management system. Blackboard Learn was also used to post grades, collect assignments, and make annoucements.

The course was assigned 3 teaching assistants (TAs), who were knowledgable in wireless communications, security, and SDRs. All assistants were present in the classroom during the labs.

C. Student Work Evaluation

Student work was graded regularly. Each week, specific lab checkpoints were communicated to students. In order to ensure students were not falling behind, these checkpoints were strictly enforced. When students reached a checkpoint, they called one of the TAs over to their table and demonstrated the functionality of their setup. They were also asked to clearly explain what they did and how they did it. A successful demonstration yielded full points for that checkpoint. In case something was wrong, the TA tried to help them by leading them towards the correct output without actually giving them the answer. If a checkpoint was not met at the end of the meeting time, partial credit was given. A general grading breakdown is presented in Table II.

TABLE II: Final grade breakdown

Labs & Homework	60%
Quizzes	10%
Final Project	30%

The final project, which will be described later, was a competition based setup, where students were *fighting* against the instructor and the TAs. The grading was performed based on how successful their security implementations were. Further discussion of the final project is discussed below.

Fig. 1: USRP N210 SDR [20]

III. Cybersecurity Lab Modules

Our course was designed to introduce students to security vulnerabilities present in wireless networks. It took a Software-Defined Radio (SDR) implementation approach to demonstrate selected wireless network security challenges. With the use of open-source tools and commercial off-the-shelf SDRs, students gained hands-on experience to analyze wireless security problems and prototype their solutions.

This course met 3 hours a week for 11 weeks and covered five separate laboratory assignments spanning various topics in wireless security. The labs were organized as follows:

A. Lab 1: Introduction to GNU Radio and USRP

This lab introduced students to the concept of software radios and gave specific examples of an SDR design flow. Students were introduced to the course's main hardware platform, the Universal Software Radio Peripheral (USRP) seen in Fig. 1, and its accompanying software design tool, GNU Radio. The lab also included a run-through for setting up the development environment on an Ubuntu 12.04 installation and executing the included example designs. Most importantly, the lab showed a tutorial on implementing a single carrier wireless communication link between two USRP nodes. Students were then asked to extend this tutorial to come up with their own implementations.

B. Lab 2: Digital Communications Hardware Design

This second lab familiarized students with the tools and concepts required for hardware design in the context of digital communication. The USRP platform is extremely customizable and allows for user-defined operations on the hardware side. This lab merges several

different topics from Computer Engineering, Telecommunications, and Software Engineering to create an interdisciplinary bridge, which continues throughout the rest of the labs. Students were instructed to implement the modulation and coding components for a wireless link on the FPGA available on the USRP, rather than processing them on the host side. Some of the concepts introduced in this lab were adopted from current research described in [21] and [22]. The design environment was conducted in Xilinx System Generator, an industry-standard hardware design suite for FPGAs. Through this lab, students gained essential hardware design skills required for prototyping fast (in the order to nanoseconds) turn-around wireless systems with real-time attack and defense capabilities.

C. Lab 3: Basics of Eavesdropping and Encryption

This lab covered signal detection, classification, and encryption techniques. Students were introduced to the basics of passive wireless attacks, including raw signal capture and off-line processing to identify the frame structures of common over-the-air (OTA) wireless protocols including WiFi and 4G WiMAX. Provided with an always-on transmitted signal from the instructors (realized using the USRP), students were asked to sniff the OTA packets and identify their key signatures, such as the source IP address, port number, packet length, etc. The lab mainly illustrated the role of an eavesdropper on an unencrypted wireless network. However, state-of-the-art encryption techniques were also introduced along with the history of cyrptography. This discussion was also complemented with current research practices in Physical-layer security schemes [13]. Students were asked to take additional measures to secure the wireless link using any of the discussed encryption techniques.

D. Lab 4: Real-Time Signal Detection

We continue to follow the multi-disciplinary approach in this lab by introducing students to real-time, hardware-based techniques for signal detection. It covered important signal processing techniques such as cross-correlators, energy detectors, frame filtering, and identification of signal components with low entropy. Students were asked to design, validate, and demonstrate a series of signal detectors. For extra credit, students should allow certain customization capabilities for their detectors, including the ability to search for signals from different standards such as WiFi and WiMAX signals. The lab required FPGA hardware logic design and embedded programming techniques to capture and analyze packets in real time. This lab prepared students towards developing eavesdropping-resilient signal transmissions.

E. Lab 5: Introduction to Jamming

This lab covered the basics of signal jamming, a series of active attacks to render the wireless network unusable. Students were introduced to the different types of jammers and asked to improve upon their design from Lab 4 to realize a reactive jammer [23]. Reactive jamming is a sophisticated form of jamming attack, wherein the attacker listens on the wireless channel and triggers jamming waveforms on positive signal detection. Due to legal considerations concerning the operating of real-time over-the-air jammers, this lab was conducted primarily in simulation. The jamming performance was analyzed and evaluated under variable signal-to-interference (SIR) ratio and different modulation parameters. This lab also provided theoretical background on jamming mitigation techniques, such as frequency hopping and spread spectrum communication [24].

F. Final Competition: Multi-team Melee

By the end of the five labs, students had gained understanding of SDR programming basics and important wireless security principles. The final project consisted of an aggregation and extension of the labs covered. Students were asked to implement an over-the-air eavesdropper using the GNU Radio / USRP framework. The instructors provided a working WiFi 802.11g link using either commercial hardware or SDRs with a fixed bit rate. The students' goals were to listen in on this wireless communication and extract any useful information they could find. In the first phase of the project, transmitted messages were encrypted with a given restricted set of keys, which were provided to students. In the second phase of the project, the keys used in message encryption were drawn from an unrestricted set of keys, and students had to conduct brute-force attacks to guess the correct keys while eavesdropping on the wireless medium. The student team that could extract the most useful information from the provided wireless link would win the competition.

IV. Laboratory Case Study: Real-Time Signal Detection (Lab 4)

In this section, we provide as an example the detailed description of one of the labs covered in our SDR security course. The materials presented here are representative of the scope and outcome expectations of every

other laboratory assignment. We focus our description on Lab 4: Real-time signal detection.

A. Lab Introduction

This lab introduces the concept of real-time signal detection in wireless communications. Signal detection is an important procedure to achieve synchronization between the transmitter and receiver, as well as to gain awareness of an active transmission in the channel. We will focus on two primary methods to detect the presence of a signal: cross-correlation and energy detection.

Wireless communications rely on successfully transmitting signals and correctly detecting them on the receiver end. A thorough understanding of packet detection principles is necessary to study cybersecurity principles. Many of the cyber attacks originate from an eavesdropper, who is trying to snoop on the packets in a wireless network. In this lab, we expect students to gain knowledge on packet detection principles and urge them to think of possible ways to develop algorithms to protect their data against an eavesdropper.

Cross-correlation: A cross-correlator performs template-based matching between the incoming wireless signal and a standardized template to identify the template presence in the signal. The template is usually designed with a very high auto-correlation property, yielding a high coefficient when correlating with itself. This enables us to pinpoint precisely the start of an active transmission frame in fine-grain synchronization.

The cross-correlator can be implemented as a matched filter, which is simply an FIR filter with tap coefficients set to the time-inverse of the template values. Recall that an N-th order FIR filter is implemented with an N-tap convolution:

$$y[n] = \sum_{i=0}^{N-1} h_i x[n-i]$$

In hardware, an FIR filter can be implemented as either a parallel or serial filter. Figures 2 and 3 show the block diagrams of example hardware implementations for both filter types. The serial version of the filter runs at $3x$ sample rate in order to produce outputs at the same rate as the input samples.

Energy Detection: an energy detector computes the short-term energy of incoming signal and tries to detect a rise or fall in the energy levels. Energy detectors are effective in detecting an active transmission without the need to understand the underlying modulation methods and data content.

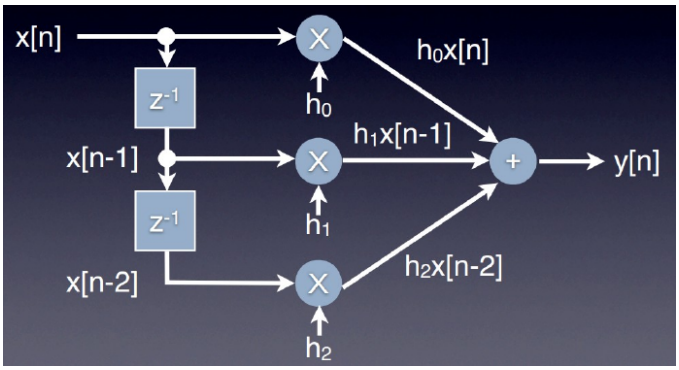

Fig. 2: Parallel 3-Tap FIR Filter Example

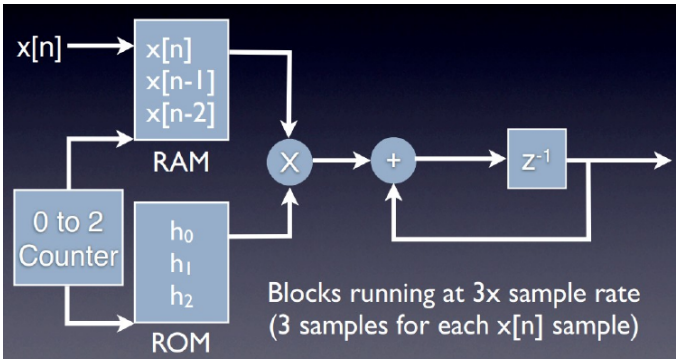

Fig. 3: Serial 3-Tap FIR Filter Example

In hardware, the energy detector can be implemented as an energy sum calculator, which continuously compares the energy level of incoming samples against the recent past to detect an energy rise or fall. Figure 4 shows an example structure of an energy detector. In essence, this hardware block keeps a running sum of N recent energy readings, where N is the desired length of the detector (specified here as 32 samples). At the nth instant, an energy reading $x[n]$ is computed from the incoming pair of I and Q values. The energy sum $y[n]$ is then updated according to the relationship:

$$y[n] = y[n-1] + x[n] - x[n-N]$$

The output of the energy sum calculator is compared to its own previous values after scaling by a user-defined threshold for either energy high or low detection. The thresholds are selected based on the desired probability of detection and false alarm rate.

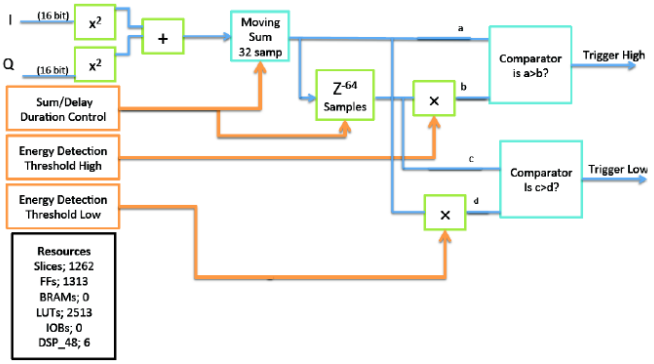

Fig. 4: Example Implementation of an Energy Detector

B. Lab Deliverables

The deliverables for this lab include two hardware implementation components: a single-rate, fully parallel FIR filter with $N = 8$ tap coefficients, and a single-rate, fully serial FIR filter with $N = 13$ coefficients. The serial design may include only one multiplier and one adder, so students are expected to use RAM memory for managing samples. Students may use either a blocked RAM or an addressable shift register (ASR) to implement the sample RAM buffer.

In addition, students need to deliver a hardware implementation of a *cross-correlator* with the 13-value Barker sequence $[1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1]$ as the template. Detailed simulation results in the System Generator are expected in the deliverables.

Students are also encouraged to identify and discuss any possible scheme that would prevent an eavesdropper to be able to employ the basic packet detection principles described above to gain access to the network. Some of the possible ideas include designing a different cross-correlator and/or leveraging the interferance in the network.

V. Course Discussion

Based on students' assignment completion and their grades against the rubrics, an overall final grade was calculated for all enrolled students. The grade distribution was shown in Table II. Once the numeric grades were calculated a standard letter mapping was used to identify a student's final letter grade. At the end of the term, 5 students received grades in the A range, 3 students received grades in the B range, and one student in the C range. This distribution yielded an average of 90.8% with a standard deviation of 10.0%. This high average could be attributed to the extremely high educator (instructor and TAs)-to-student ratio.

VI. End-of-term Student Evaluation Results

After having developed and conducted the Software Defined Radio security course, a brief feedback survey was given focused on the course's reception. In this survey, students were asked to anonymously elaborate on their experience throughout the term. They were allowed to write as little or as much as they needed. Student participation for this survey was voluntary and students were allowed to see their final grades before the submission of the survey. The received testimonials were all positive. The word map seen in Fig. 5 was created to get a general idea of the student feedback, where the most frequently used words appear the biggest. As the frequency of a specific word gets smaller, its font size follows the same trend.

While most classes rely completely on teaching the theory, our class offered students guidelines to help develop and realize current research techniques in a student-centric approach. One of the students mentioned the following in his/her feedback: *"I feel like the class is the glue which combines different pieces of knowledge together. The class is both a challenge and a great opportunity for anyone who wishes to learn more security issues in SDR."* Another student adds to the submitted evaluations by focusing on the course's hands-on approach. He/she says, *"The course involved a lot of hands on learning using [...] hardware interfacing with student-created software in a lab setting. I enjoyed this course very much due to the fact that students were able to prototype designs on the computer and then realize them in live transmissions, comparing actual results, and demonstrating the security concepts."*

Simulations are much more effective in being flexible enough to teach over a range of concepts but true learning comes from seeing the end goal. We realized this need to concentrate the concepts being taught by guiding students to physical implementations. Not only simulations and implementations were important part of the student-centric learning goal; however, we also introduced the latest achievements in the research field during our labs. Our course was described as *"an excellent balance between background theory and hardware implementation"*, where we covered *"[m]odern topics related to wireless security and communication systems [...], which highlighted examples of current technologies using the techniques discussed."*

Fig. 5: The word map created by concatenating all the student reviews.

One of the students concluded his/her review by saying he/she *"would highly recommend this course and similar courses for anyone interested in pursuing a degree in wireless communications, cryptography, or signals intelligence related fields."*

VII. Conclusion

In this paper, we presented our graduate-level Software Defined Radio based Cybersecurity course. Our course balanced theory and implementation by offering students short descriptions of their upcoming labs and then following a student-centric learning approach during the labs. The five lab modules we designed not only introduced students to wireless security concepts but also leveraged the latest findings in related research fields. Students used SDRs throughout the term to implement their algorithms via hardware implementations, embedded programming, and wireless signal processing techniques. By going outside the boundries of textbooks, and integrating current research into the classroom, we were able to offer a more up-to-date learning experience to our students. Although, our job as course developers was made more challenging by not being bound to a textbook and by following and integrating the latest research achievements into our course, testimonials we received show that the end product was appreciated by students.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. DUE-1241631. The authors would like to also thank Doug Pfeil for his valuable comments throughout the design stage of the course.

References

[1] "Statistics - YouTube," https://www.youtube.com/yt/press/statistics.html, accessed: 2015-04-24.

[2] "More colleges offer programs in cybersecurity," http://www.wsj.com/articles/SB10001424127887323527004579081031537896834, accessed: 2015-03-20.

[3] C. Paulsen, E. McDuffie, W. Newhouse, and P. Toth, "NICE: Creating a cybersecurity workforce and aware public," *IEEE Security & Privacy*, vol. 10, no. 3, pp. 0076–79, 2012.

[4] "National Initiative for Cybersecurity Education (NICE)," http://csrc.nist.gov/nice/, accessed: 2015-03-20.

[5] "IBM - a collaborative approach for cybersecurity education," http://www.ibm.com/smarterplanet/us/en/ centerforappliedinsights/article/cybersecurity.html, accessed: 2015-03-20.

[6] "PBS NOVA CyberLab," http://www.pbs.org/wgbh/nova/labs/lab/cyber/, accessed: 2015-03-20.

[7] S. Katz and J. Flynn, "Using software defined radio (sdr) to demonstrate concepts in communications and signal processing courses," in *Frontiers in Education Conference, 2009. FIE'09. 39th IEEE*. IEEE, 2009, pp. 1–6.

[8] "GNURadioCompanion - GNU Radio - gnuradio.org," https://gnuradio.org/redmine/projects/gnuradio/wiki/ GNURadioCompanion, accessed: 2015-04-27.

[9] J. Snyder, B. McNair, S. Edwards, and C. Dietrich, "Ossie: An open source software defined radio platform for education and research," in *International conference on frontiers in education: computer science and computer engineering (FECS11). World congress in computer science, Computer engineering and applied computing. Las Vegas, NV*, 2011.

[10] L. S. Nagurney, "Software defined radio in the electrical and computer engineering curriculum," in *Frontiers in Education Conference, 2009. FIE'09. 39th IEEE*. IEEE, 2009, pp. 1–6.

[11] T. Rappaport, "Wireless communications: Principles and practice," 2001.

[12] N. Ferguson, B. Schneier, and T. Kohno, *Cryptography engineering: design principles and practical applications*. John Wiley & Sons, 2011.

[13] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM, 2008, pp. 128–139.

[14] M. E. Huba and J. E. Freed, "Learner centered assessment on college campuses: Shifting the focus from teaching to learning," *Community College Journal of Research and Practice*, vol. 24, no. 9, pp. 759–766, 2000.

[15] C. Sahin and P. Abichandani, "Should the first course in computational problem solving and programming be student-centered or teacher-centered?" in *Frontiers in Education Conference, 2013 IEEE*. IEEE, 2013, pp. 748–754.

[16] J. H. Sandholtz *et al.*, *Teaching with technology: Creating student-centered classrooms.* ERIC, 1997.

[17] M. Papastergiou, "Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation," *Computers & Education*, vol. 52, no. 1, pp. 1–12, 2009.

[18] K. Kiili, "Digital game-based learning: Towards an experiential gaming model," *The Internet and higher education*, vol. 8, no. 1, pp. 13–24, 2005.

[19] M. Prensky, "Computer games and learning: Digital game-based learning," *Handbook of computer game studies*, vol. 18, pp. 97–122, 2005.

[20] "Ettus research - product detail," http://www.ettus.com/product/details/UN210-KIT, accessed: 2015-04-27.

[21] J. Chacko, C. Sahin, D. Nguyen, D. Pfeil, N. Kandasamy, and K. Dandekar, "Fpga-based latency-insensitive ofdm pipeline for wireless research," in *High Performance Extreme Computing Conference (HPEC), 2014 IEEE*, Sept 2014, pp. 1–6.

[22] J. Chacko, C. Sahin, D. Pfiel, N. Kandasamy, and K. Dandekar, "Rapid prototyping of wireless physical layer modules using flexible software/hardware design flow," in *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '15. New York, NY, USA: ACM, 2015, pp. 32–35. [Online]. Available: http://doi.acm.org/10.1145/2684746.2689084

[23] D. Nguyen, C. Sahin, B. Shishkin, N. Kandasamy, and K. R. Dandekar, "A real-time and protocol-aware reactive jamming framework built on software-defined radios," in *Proceedings of the 2014 ACM Workshop on Software Radio Implementation Forum*, ser. SRIF '14. New York, NY, USA: ACM, 2014, pp. 15–22. [Online]. Available: http://doi.acm.org/10.1145/2627788.2627798

[24] M. Strasser, S. Capkun, and M. Cagalj, "Jamming-resistant key establishment using uncoordinated frequency hopping," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 64–78.