Blase Eric Ur: Teaching Statement

I have been fortunate to have numerous teaching and mentoring experiences very early in my career that have collectively shaped my teaching philosophy. These experiences include co-teaching graduate-level classes, teaching 500 freshmen programming, and mentoring K-12 students typically underrepresented in CS.

Prior Teaching Experience
At CMU, Lorrie Cranor and I twice co-taught an interdisciplinary course in Usable Privacy and Security (08-734), enrolling 20–30 students each time. Under Lorrie’s guidance, I gave about 40% of the lectures, wrote the assignments, mentored half of the course projects, restructured the course from prior offerings, and selected the readings. In addition to preparing me to teach my own course combining security/privacy with HCI, my co-teaching experience taught me about designing an interdisciplinary course that engages students with very different technical backgrounds and varied goals.

Although I fulfilled my CMU teaching requirements through the 08-734 course, I chose to gain additional preparation for more traditional computer security classes by serving as a TA for CMU’s graduate-level class on Secure Software Systems (18-732), taught by Lujo Bauer. This course, which enrolled 60 students, focused on building security into software systems. We covered defensive techniques ranging from software analysis (e.g., static analysis, taint tracking) to more formal solutions (e.g., trusted computing). I wrote and graded assignments, held office hours, and wrote and graded exams. I gave guest lectures on bounded model checking for 18-732 and on authentication for multiple other CMU security classes.

Before coming to CMU, I was the lead instructor for 2.5 years of an introductory computer programming course (14:440:127) at Rutgers University. In this class, required for all engineering undergraduates, I taught the basics of computer programming and data analysis using Matlab. Each semester, the course enrolled up to 500 students, providing a “trial by fire” as my first lead teaching experience.

Experience Mentoring Undergraduates and Master’s Students
I have been privileged to mentor nine CMU students during graduate school. Three of them were master’s students, while six were undergraduates. I worked with these students as co-advisor on a master’s thesis, primary mentor for independent-study research, or as primary supervisor for summer research internships. In each case, I provided the initial idea for the project and met at least weekly with each mentee to discuss their ideas, progress, and technical challenges. One student’s master’s thesis led to us co-authoring a SOUPS paper, while an undergrad and I co-authored a WEIS paper. I have also supervised course projects, leading to two papers and five posters. Furthermore, my collaboration with Michael Littman at Brown has enabled me to work with his students; all of our papers have included multiple undergraduate co-authors.

While mentoring a student, I believe it is crucial for the student to take ownership of the project and to know that the ideas they contribute will be respected. This semester, three undergrads and one master’s student are working with me on designing a better password-strength meter. In addition to individual meetings, we meet weekly as a group. Each student knows that some aspect of the meter is their domain.

Mentoring to Broaden Participation in CS
I greatly enjoy working to broaden participation in computer science, focusing on students from historically underrepresented social groups in the field. I believe such efforts are crucial to building a more representative CS community. I have been deeply involved in outreach efforts through CMU’s Women@SCS and SCS4ALL.

One of my main efforts has been volunteering twice weekly at a local middle school. I chose topics like forming a “middle-school hacking team” and creating interactive art using Arduinos, recruiting students who did not initially plan to study CS. In addition, I have led a workshop for 30 middle-school girls on password cracking and presented over a dozen “roadshows” that describe to K-12 students the possibilities of a CS
career. This semester, I began to serve monthly as a facilitator for CMU’s Bias Busters program to combat unconscious bias in CS. Based on my involvement and leadership, I was one of four CMU students selected to present our campus-wide outreach efforts to Arne Duncan, the U.S. Secretary of Education.

I have also mentored six high-school students. Three of them worked on passwords projects, two focused on privacy, and the final one investigated digital music. Three of my advisees have earned first-place awards in Pittsburgh’s science fair, and one student won numerous prizes at our statewide science fair.

For three years, broadening participation in CS and engineering was my full-time job. At Rutgers University, I coordinated the NJ Governor’s School of Engineering & Technology, a summer program for high-school juniors. I also advised students from the Educational Opportunity Fund program that supports low-income undergraduate engineering students. In addition, I coordinated a regional science fair for high-school students, presented at schools, and ran an honors seminar for engineering undergraduates.

Philosophy
From working with K-12 students, I have concluded that a good teacher must be able to explain the intuition behind even advanced security concepts in terms that a 12-year-old can understand. I believe in never underestimating what students can accomplish with the right guidance. My peers were skeptical when I decided to teach my middle-school students the network monitoring tool Wireshark, in addition to explaining the underlying science and ethical considerations. My students justified my trust by using Wireshark both deftly and responsibly. Similarly, one student expressed a desire, at the age of 13, to build a privacy-protective Chrome extension. I am happy to report that the beta version of his tool is available on Github.

I find it crucial for students, with my active guidance, to identify and articulate exactly what research problem they hope to solve. When a student takes ownership of a problem, I have an opening as a teacher to introduce new concepts to help solve that problem. This approach generalizes to lectures, as well. I once started a lecture on usable encryption by showing a video Edward Snowden had made to teach the journalist Glenn Greenwald to use encryption. Seeing Snowden’s struggles to convey this information motivated the students to pay close attention as we dove into the technical details of cryptography and adversarial models.

My experiences with traditional security classes taught me to use applications to motivate students, particularly the many who were in CMU’s professional master’s program in security. Drawing on my hobbies, I created a running thread through assignments in which a music group was using a networked software system to craft songs during performances. Based on efficiency and assumptions about rival musicians trying to disrupt live performances or steal demos, the students decided which software-security techniques to deploy.

Finally, I believe it essential to build a strong community around my future research group. One person cannot scale infinitely, but he or she can reach similar ends by imbuing a culture into a research group. I have seen this approach work well in the CUPS lab run by my advisor, Lorrie Cranor. Lorrie first transferred her research philosophy to her senior students, and we disseminate it further to new group members.

Teaching Plan
My diverse teaching experiences have prepared me to teach a range of courses. Having already served as co-instructor, I am excited to teach courses in Usable Security and Privacy. Such a course teaches students core technical concepts in cybersecurity and privacy, as well as how to design rigorous user studies to evaluate whether average people can use security and privacy tools. I could target this course to PhD students with a research focus, or to BS/MS students by focusing on applications. I would be similarly excited to teach interdisciplinary courses merging security and privacy technologies with ethics and society.

I would also be thrilled to teach core classes in cybersecurity (e.g., Introduction to Computer Security) and in Human-Computer Interaction. With some preparation, I can build on my experience as TA for CMU’s graduate Secure Software Systems course and my extensive coursework to teach classes like Network Security, Software Security, or Introductory Cryptography. Similarly, I could draw on my experiences designing quantitative and qualitative user studies to teach more advanced HCI classes.

Broadening and increasing participation in CS more generally is a major goal of my teaching. Drawing on my unique experience teaching introductory programming courses to hundreds of first-year undergraduates at Rutgers, I would be happy to teach introductory computer science courses as a first liaison between prospective students and the department. I would similarly enjoy teaching courses for non-majors that draw on my interests applying computational thinking to art, music, and society.

Outreach will remain an important part of my identity. Beyond striving to be an excellent mentor to my graduate and undergraduate students, I will continue to integrate K-12 students into my group’s research and engage with traditionally underserved and underrepresented students to improve our CS community.