

TREE OF TRUST



Pengyan Yu (left) and Eric Simpson demonstrate SAM (secure authenticated mode). Unauthorized devices see only the noise on the right, while the authorized device sees the magic word.

While hackers and viruses grab headlines in computing security, Patrick Schaumont’s team is working to protect data from a newer, growing threat—a threat from the loss and theft of embedded computers that store personal and private information.

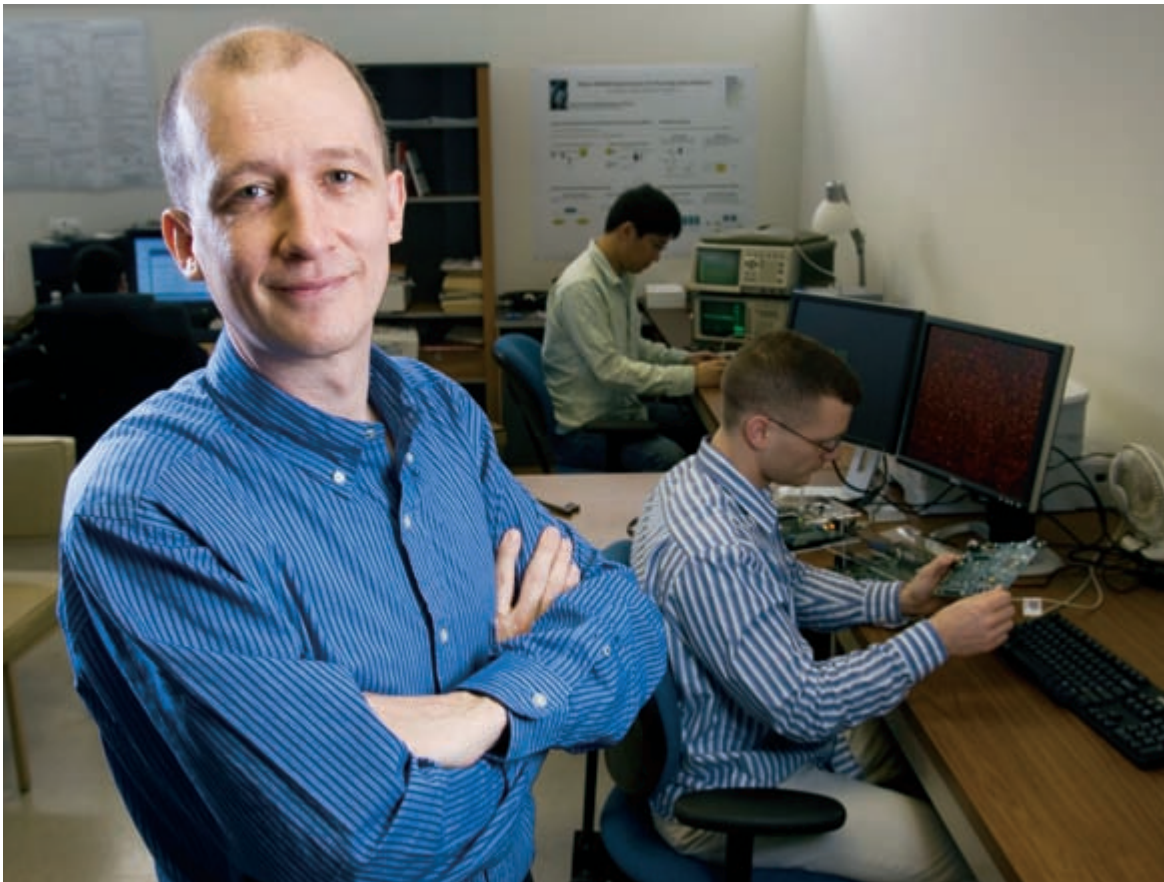
“Today’s computers are so small that we can carry them around and lose them,” he says. “Who among us can claim never to have lost a credit card, a cell phone, a PDA, a laptop?”

Medical records, high-resolution pictures of pen-drawn signatures, and codes to unlock electronic car

locks are just some of the examples of information now stored on smart cards, key fobs, and other embedded computers. The problem in keeping the information safe, Schaumont says, is that conventional computer security and cryptography focuses on protecting data during transmission, but once thieves have possession of the computer, transmission is not an issue and the data is vulnerable.

The solution is to carefully design security into the hardware and software of a computer, but engineers commonly add protection only as an afterthought. Schaumont is working to develop a methodology for embedded systems engineers to follow in designing both

“We could break the code in just three minutes by measuring the power.”



Patrick Schaumont received an NSF CAREER Award for his work on protecting embedded systems from side-channel attacks.

hardware and software for security.

His team is seeking answers to questions, including: How can we systematically deal with private information processing in a portable computer? How can we implement cost-effective encryption mechanisms that are energy-efficient and secure? How can we store secrets in a reliable way in a portable system?

“Securing secrets is not part of the typical engineer’s toolbox,” he says. “We want to change that.”

Schaumont has received funding for his effort from a National Science Foundation (NSF) Faculty Career Development Program (CAREER) award. The CAREER grant is worth \$400,000 over five years and is NSF’s most prestigious award for junior faculty members. Schaumont joined ECE in 2005 as an assistant professor.

Greatest hazards are side channel attacks

“As in any kind of security, a secure embedded system is only as safe as its weakest link,” Schaumont says. Once thieves have

possession of an embedded computer, secrets can be easily probed out, and the thief doesn’t need to know the password to do that. For example, a cryptographic chip can often be analyzed with a side channel attack by measurement of only its power consumption pattern.

“You don’t need to open the chip to do this,” Schaumont explains. “It’s a systematic problem, much like a communications problem. You can filter out the noise and detect the code you seek.” He was on a team in the Secure Embedded Systems Group at the University of California at Los Angeles (UCLA) that demonstrated such a side-channel attack on a standard encryption chip. “We could break the code in just three minutes by measuring the power,” he says.

Side-channel attacks have been recognized as a security issue only in the past decade, according to Schaumont. “Most people are familiar with software attacks and physical attacks, but not side-channel attacks,” he says. (*Continued on p. 10.*)

Software attacks often arrive via virus or spyware and involve a complicated process that requires expert knowledge, Schaumont explains. “Logical attacks require brains, but they are cheap. All an attacker needs is patience.” Classical security software is designed to thwart these attacks.

Physical attacks, often used in reverse engineering, require disassembly and are costly. “Physical attacks require expensive, specialized equipment, such as microscopes and chemicals to get down to the chip layer,” he says. “It’s expensive, but it’s not hard. Using a systematic approach, it can be done.”

Side-channel attacks are ideal for embedded systems from an attacker’s perspective. “You don’t need to open a chip and don’t need expensive hardware. Plus, it is systematic.” Protecting against these attacks is very challenging, Schaumont says. “There are many abstraction layers—all of them exposed. Current countermeasures are all point solutions, implemented only in software or hardware. They offer no guarantees for the overall system.”

Hardware/Software Codesign

The solution to designing secure embedded systems is to pay attention to all abstraction layers, and this requires hardware/software codesign, he says. “The flexibility of software supports complex crypto-algorithms. On the other hand, hardware can provide side-channel countermeasures that are hard to implement with software, such as constant-power execution. A combination of the two will enable flexible and secure system design.”

“Securing secrets is not part of the typical engineer’s toolbox. We want to change that.”

Tree of Trust

Schaumont wants to develop a systematic approach to side-channel-resistant embedded system design. “Security is hard to optimize, hard to quantify,” he explains. He is proposing a systematic approach to side-channel-resistant embedded system design he calls “The Tree of Trust.”

With the Tree of Trust, a designer can systematically partition a system into a secure-critical and a non-critical part. A designer needs to be able to systematically shield those secrets. Secrets go into the critical part. A completely isolated secret is meaningless, however. All these protected secrets are only useful if they can interact. The Tree of Trust makes sure that secrets are integrated into the system with a secure interface.

“You are trying to defend your root of trust,” he says, explaining that the root of trust is the element that is implicitly trusted, such as the key of an encryption algorithm. “Implementing the root of trust is always expensive, so the smaller you can make this, the cheaper your system is. Indeed, the non-trusted elements can use standard components or software.”

The embedded system contains several abstraction levels, including the protocol, the algorithm, the architecture, hardware, and circuit levels. “If you partition the system correctly at every level, you will obtain a very small root-of-trust which can be protected with cost-effective countermeasures across the hardware/software boundaries,” he explains.

Building With Secrets

His goal, he says, is “to come up with a way in which people can systematically deal with the design of secrets when designing a system . . . We are trying to find a canned sequence of operations that a designer of secure hardware and software could use.”

His team is applying the Tree of Trust concept to different embedded-system implementations. One ongoing application involves protecting content using “Secure Authenticated Mode,” or SAM for short. Using SAM, a video message can be created that can be displayed only on a unique and single device. Another project is the development of protection mechanisms for DSP software in sensor nodes that remain active in the field for several years. “The sensor nodes contain highly advanced signal processing to gather information on activities in their environment. The software represents an important amount of intellectual property,” he says.

Targeting the Engineers

A critical part of Schaumont’s plan is educating the engineers who design embedded systems. He has involved undergraduate students in the research project and introduced a new undergraduate course last fall called Hardware/Software Codesign. He is also working on a team that is developing a freely available CD-ROM for undergraduates that includes a codesign environment and tools.

“Hardware/software codesign has typically been a graduate-level topic, however we want our undergraduates to be able to compete on a global scale and be capable of designing complex embedded systems,” he says. Schaumont plans to develop an additional course at the graduate level focusing specifically on secure embedded systems.

“We want engineers to develop a sense about how to build secure embedded systems and to understand what is breakable and what is not.”