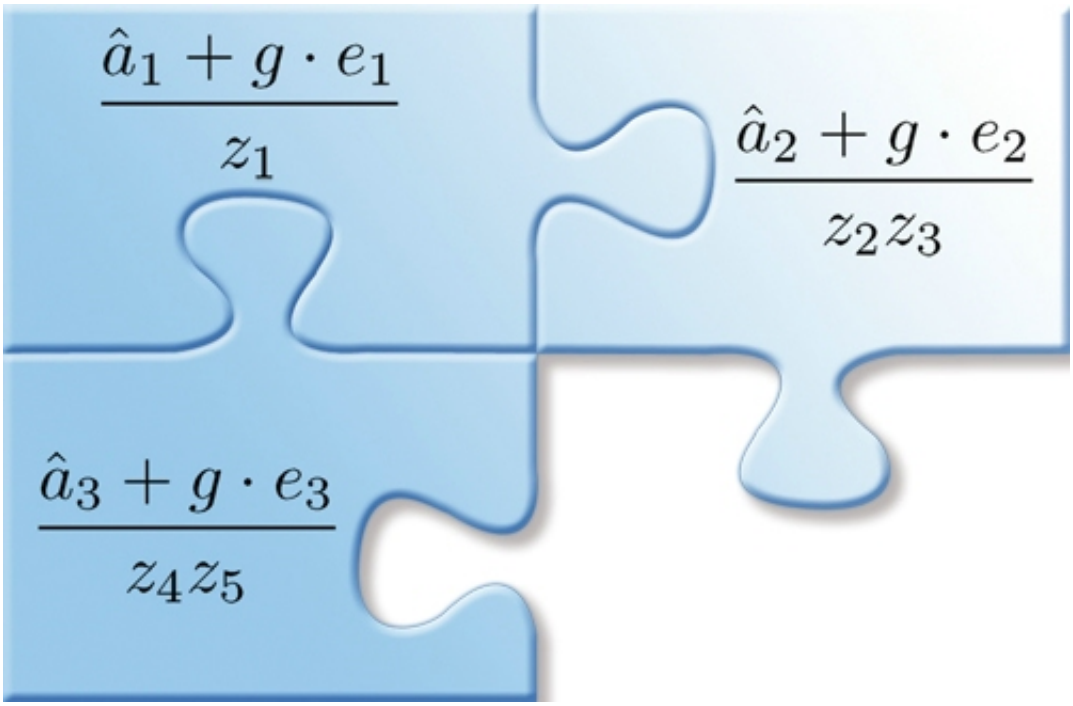
Computer scientists develop 'mathematical jigsaw puzzles' to encrypt software

July 29 2013, by Matthew Chin



Concept illustration of mathematical jigsaw puzzle Credit: UCLA Engineering

(Phys.org) —UCLA computer science professor Amit Sahai and a team of researchers have designed a system to encrypt software so that it only allows someone to use a program as intended while preventing any deciphering of the code behind it. This is known in computer science as "software obfuscation," and it is the first time it has been accomplished.

Sahai, who specializes in cryptography at UCLA's Henry Samueli School

of Engineering and Applied Science, collaborated with Sanjam Garg, who recently earned his doctorate at UCLA and is now at IBM Research; Craig Gentry, Shai Halevi and Mariana Raykova of IBM Research; and Brent Waters, an assistant professor of computer science at the University of Texas at Austin. Garg worked with Sahai as a student when the research was done.

Their peer-reviewed paper will be formally presented in October at the 54th annual IEEE Symposium on Foundations of Computer Science, one of the two most prominent conferences in the field of theoretical computer science. Sahai has also presented this research in recent invited talks at Stanford University and the Massachusetts Institute of Technology.

"The real challenge and the great mystery in the field was: Can you actually take a piece of software and encrypt it but still have it be runnable, executable and fully functional," Sahai said. "It's a question that a lot of companies have been interested in for a long time."

According to Sahai, previously developed techniques for obfuscation presented only a "speed bump," forcing an attacker to spend some effort, perhaps a few days, trying to reverse-engineer the software. The new system, he said, puts up an "iron wall," making it impossible for an adversary to reverse-engineer the software without solving mathematical problems that take hundreds of years to work out on today's computers—a game-change in the field of cryptography.

The researchers said their mathematical obfuscation mechanism can be used to protect intellectual property by preventing the theft of new algorithms and by hiding the vulnerability a software patch is designed to repair when the patch is distributed.

"You write your software in a nice, reasonable, human-understandable

way and then feed that software to our system," Sahai said. "It will output this mathematically transformed piece of software that would be equivalent in functionality, but when you look at it, you would have no idea what it's doing."

The key to this successful obfuscation mechanism is a new type of "multilinear jigsaw puzzle." Through this mechanism, attempts to find out why and how the software works will be thwarted with only a nonsensical jumble of numbers.

"The real innovation that we have here is a way of transforming software into a kind of mathematical jigsaw puzzle," Sahai said. "What we're giving you is just math, just numbers, or a sequence of numbers. But it lives in this mathematical structure so that these individual pieces, these sequences of numbers, can only be combined with other numbers in very specified ways.

"You can inspect everything, you can turn it upside-down, you can look at it from different angles and you still won't have any idea what it's doing," he added. "The only thing you can do with it is put it together the way that it was meant to interlock. If you tried to do anything else—like if you tried to bash this piece and put it in some other way—you'd just end up with garbage."

Functional encryption

The new technique for software obfuscation paved the way for another breakthrough called functional encryption. With functional encryption, instead of sending an encrypted message, an encrypted function is sent in its place. This offers a much more secure way to protect information, Sahai said. Previous work on functional encryption was limited to supporting very few functions; the new work can handle any computable function.

For example, a single message could be sent to a group of people in such a way that each receiver would obtain different information, depending on characteristics of that particular receiver. In another example, a hospital could share the outcomes of treatment with researchers without revealing details such as identifying patient information.

"Through functional encryption, you only get the specific answer, you don't learn anything else," Sahai said.

More information: Paper: eprint.iacr.org/2013/451

Provided by University of California, Los Angeles