Thwarting the cleverest attackers

May 1 2012, by: Larry Hardesty

In the last 10 years, cryptography researchers have demonstrated that even the most secure-seeming computer is shockingly vulnerable to attack. The time it takes a computer to store data in memory, fluctuations in its power consumption and even the noises it emits can betray information to a savvy assailant.

Attacks that use such indirect sources of information are called side-channel attacks, and the increasing popularity of cloud computing makes them an even greater threat. An attacker would have to be pretty motivated to install a device in your wall to measure your computer's power consumption. But it's comparatively easy to load a bit of code on a server in the cloud and eavesdrop on other applications it's running.

Fortunately, even as they've been researching side-channel attacks, cryptographers have also been investigating ways of stopping them. Shafi Goldwasser, the RSA Professor of Electrical Engineering and Computer Science at MIT, and her former student Guy Rothblum, who's now a researcher at Microsoft Research, recently posted a long report on the website of the Electronic Colloquium on Computational Complexity, describing a general approach to mitigating side-channel attacks. At the Association for Computing Machinery's Symposium on Theory of Computing (STOC) in May, Goldwasser and colleagues will present a paper demonstrating how the technique she developed with Rothblum can be adapted to protect information processed on web servers.

In addition to preventing attacks on private information, Goldwasser says, the technique could also protect devices that use proprietary

algorithms so that they can't be reverse-engineered by pirates or market competitors — an application that she, Rothblum and others described at last year's AsiaCrypt conference.

Today, when a personal computer is in use, it's usually running multiple programs — say, a word processor, a browser, a PDF viewer, maybe an email program or a spreadsheet program. All the programs are storing data in memory, but the laptop's operating system won't let any program look at the data stored by any other. The operating systems running on servers in the cloud are no different, but a malicious program could launch a side-channel attack simply by sending its own data to memory over and over again. From the time the data storage and retrieval takes, it can infer what the other programs are doing with remarkable accuracy.

Goldwasser and Rothblum's technique obscures the computational details of a program, whether it's running on a laptop or a server. Their system converts a given computation into a sequence of smaller computational modules. Data fed into the first module is encrypted, and at no point during the module's execution is it decrypted. The still-encrypted output of the first module is fed into the second module, which encrypts it in yet a different way, and so on.

The encryption schemes and the modules are devised so that the output of the final module is exactly the output of the original computation. But the operations performed by the individual modules are entirely different. A side-channel attacker could extract information about how the data in any given module is encrypted, but that won't let him deduce what the sequence of modules do as a whole. "The adversary can take measurements of each module," Goldwasser says, "but they can't learn anything more than they could from a black box."

The report by Goldwasser and Rothblum describes a type of compiler, a program that takes code written in a form intelligible to humans and

converts it into the low-level instruction intelligible to a computer. There, the computational modules are an abstraction: The instruction that inaugurates a new module looks no different from the instruction that concluded the last one. But in the STOC paper, the modules are executed on different servers on a network.

Provided by Massachusetts Institute of Technology