New System Allows Cloud Customers to Detect Program-Tampering

Andrew Carleen, Massachusetts Institute of Technology

A new version of 'zero-knowledge proofs' allows cloud customers to verify the proper execution of their software with a single packet of data.

Cambridge, MA -- For small and midsize organizations, the outsourcing of demanding computational tasks to the cloud — huge banks of computers accessible over the Internet — can be much more cost-effective than buying their own hardware. But it also poses a security risk: A malicious hacker could rent space on a cloud server and use it to launch programs that hijack legitimate applications, interfering with their execution.

In August, at the International Cryptology Conference, researchers from MIT [1] and Israel's Technion and Tel Aviv University presented a new system that can quickly verify that a program running on the cloud is executing properly. That amounts to a guarantee that no malicious code is interfering with the program's execution.

The same system also protects the data used by applications running in the cloud, cryptographically ensuring that the user won't learn anything other than the immediate results of the requested computation. If, for instance, hospitals were pooling medical data in a huge database hosted on the cloud, researchers could look for patterns in the data without compromising patient privacy.

Although the paper reports new theoretical results, the researchers have also built working code that implements their system. At present, it works only with programs written in the C programming language, but adapting it to other languages should be straightforward.

The new work, like much current research on secure computation, requires that computer programs be represented as circuits. So the researchers' system includes a "circuit generator" that automatically converts C code to circuit diagrams. The circuits it produces, however, are much smaller than those produced by its predecessors, so by itself, the circuit generator may find other applications in cryptography.

Zero Knowledge

Alessandro Chiesa, a graduate student in electrical engineering and computer science at MIT and one of the paper's authors, says that because the new system protects both the integrity of programs running in the cloud and the data they use, it's a good complement to the cryptographic technique known as homomorphic encryption, which protects the data transmitted by the users of cloud applications.

Joining Chiesa on the paper are Madars Virza, also a graduate student in electrical engineering and computer science; the Technion's Daniel Genkin and Eli Ben-Sasson, who was a visiting professor at MIT for the past two years; and Tel Aviv University's Eran Tromer, who was a postdoc at MIT. The researchers' system implements a so-called zero-knowledge proof, a type of mathematical game invented by MIT professors Shafi Goldwasser and Silvio Micali and their colleague Charles Rackoff of the University of Toronto. In its cryptographic application, a zero-knowledge proof enables one of the game's players to prove to the other that he or she knows a secret key without actually divulging it.

But as its name implies, a zero-knowledge proof is a more general method for proving mathematical theorems — and the correct execution of a computer program can be redescribed as a theorem. So zero-knowledge proofs are by definition able to establish whether or not a computer program is executing correctly.

The problem is that existing implementations of zero-knowledge proofs — except in cases where they've been tailored to particular algorithms — take as long to execute as the programs they're trying to verify. That's fine for password verification, but not for a computation substantial enough that it might be farmed out to the cloud.

The researchers' innovation is a practical, succinct zero-knowledge proof for arbitrary programs. Indeed, it's so succinct that it can typically fit in a single data packet.

Linear Thinking

As Chiesa explains, his and his colleagues' approach depends on a variation of what's known as a "probabilistically checkable proof," or PCP. "With a standard mathematical proof, if you want to verify it, you have to go line by line from the start to the end," Chiesa says. "If you were to skip one line, potentially, that could fool you. Traditional proofs are very fragile in this respect." "The PCP theorem says that there is a way to rewrite proofs so that instead of reading them line by line,"

Chiesa adds, "what you can do is flip a few coins and probabilistically sample three or four lines and have a probabilistic guarantee that it's correct."

The problem, Virza says, is that "the current known constructions of the PCP theorem, though great in theory, have quite bad practical realizations." That's because the theory assumes that an adversary who's trying to produce a fraudulent proof has unbounded computational capacity. What Chiesa, Virza and their colleagues do instead is assume that the adversary is capable only of performing simple linear operations.

"This assumption is, of course, false in practice," Virza says. "So we use a cryptographic encoding to force the adversary to only linear evaluations. There is a way to encode numbers into such a form that you can add those numbers, but you can't do anything else. This is how we sidestep the inefficiencies of the PCP theorem.

For more information visit http://www.mit.edu/ [1]

Links:
[1] http://www.mit.edu/