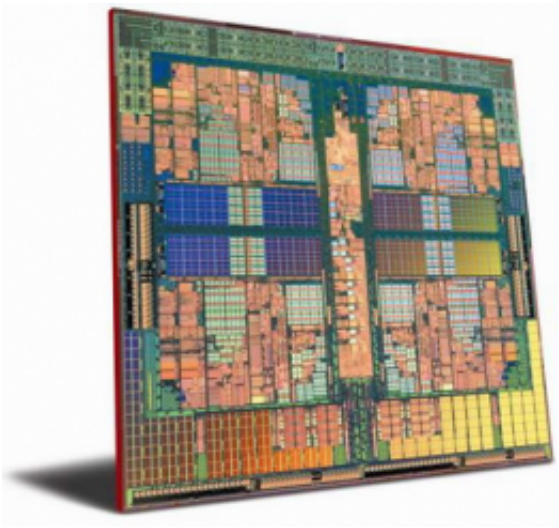
Cloud security reaches silicon: System for defending against memory-access attacks implemented in chips

April 22 2015



In the last 10 years, computer security researchers have shown that malicious hackers don't need to see your data in order to steal your data. From the pattern in which your computer accesses its memory banks, adversaries can infer a shocking amount about what's stored there.

The risk of such attacks is particularly acute in the cloud, where you have no control over whose applications are sharing server space with yours. An antagonist could load up multiple cloud servers with small

programs that do nothing but spy on other people's data.

Two years ago, researchers in the group of MIT's Srini Devadas, the Edwin Sibley Webster Professor in MIT's Department of Electrical Engineering and Computer Science, proposed a method for thwarting these types of attacks by disguising memory-access patterns. Now, they've begun to implement it in hardware.

In March, at the Architectural Support for Programming Languages and Operating Systems conference, they presented the layout of a custom-built chip that would use their scheme, which is now moving into fabrication. And at the IEEE International Symposium on Field-Programmable Custom Computing Machines in May, they will describe some additional improvements to the scheme, which they've tested on reconfigurable chips.

The principle behind the scheme is that, whenever a chip needs to fetch data from a particular memory address, it should query a bunch of other addresses, too, so that an adversary can't determine which one it's really interested in. Naturally, this requires shipping much more data between the chip and memory than would otherwise be necessary.

To minimize the amount of extra data needed, the researchers store memory addresses in a data structure known as a "tree." A family tree is a familiar example of a tree, in which each "node" (a person's name) is attached to only one node above it (the node representing the person's parents) but may connect to several nodes below it (the person's children).

Every address is randomly assigned to a path through the tree—a sequence of nodes stretching from the top of the tree to the bottom, with no backtracking. When the chip requires the data stored at a particular address, it also requests data from all the other nodes on the same path.

In earlier work, researchers in Devadas' group were able to prove that pulling data from a single path was as confounding to an adversary as if the chip had pulled data from every single memory address in use—every node of the tree.

Breaking the logjam

After reading data from a path, however, the chip also has to write data to the whole path; otherwise, an adversary could determine which node was the one of interest. But the chip rarely stores data in the same node that it read it from.

Most nodes lie on multiple paths: To take the most basic example, the single node at the top, or root, of the tree lies on every path. When the chip writes a block of data to memory, it pushes it as far down the tree as it can, which means finding the last vacancy before the block's assigned path branches off from path that was just read.

"The root of the tree is a lot smaller than the bottom of tree," says Albert Kwon, an MIT graduate student in electrical engineering and computer science and one of the papers' co-authors. "So intuitively, you want to push down as far as you can toward the bottom, so that there's no congestion at the top."

In writing data, the chip still has to follow the sequence of nodes in the path; otherwise, again, an adversary might be able to infer something about the data being stored. In previous attempts at similar systems, that meant sorting the memory addresses according to their ultimate locations in the tree.

"Sort is not easy to do in hardware," says Chris Fletcher, another graduate student in Devadas' group and first author on the new paper. "So by the time you've sorted everything, you've taken a real

performance hit."

In the chip described in their latest paper, Fletcher, Devadas, Kwon, and their co-authors—Ling Ren, also an MIT graduate student in electrical engineering and computer science, and colleagues at the University of Connecticut, the University of California at Berkeley, and the Qatar Computing Research Institute—took a different approach. They gave their chip an extra memory circuit, with storage slots that can be mapped onto the sequence of nodes in any path through the tree. Once a data block's final location is determined, it's simply stored at the corresponding slot in the circuit. All of the blocks are then read out in order.

Stockpiled secrets

The new chip features another trick to improve efficiency: Rather than writing data out every time it reads data in, it writes only on every fifth read. On the other reads, it simply discards all of the decoy data. When it finally does write data back out, it will have, on average, five extra blocks of data to store on the last path it read. But there are generally enough vacancies in the tree to accommodate the extra blocks. And when there aren't, the system's ordinary protocols for pushing data as far down the tree as possible can handle the occasional logjam at the top.

Today's chips have small, local memory banks called caches in which they store frequently used data; for applications that use caching efficiently, all that extra reading and writing generally increases computation time by only about 20 percent. For applications that don't use caching efficiently, computation time can increase fivefold, or even more.

But according to the researchers, one of the advantages of their scheme is that the circuits that implement it can simply be added to existing chip

designs, without much retooling. The extra layer of security can then be switched on and off as needed. Some cloud applications may use it all the time; others may opt against it entirely; still others may activate it only when handling sensitive information, such as credit card numbers.

Provided by Massachusetts Institute of Technology