Quantum Computing and the Generalized Knapsack Code

A mathematical paper clarifying the use of number in computing and cryptography has been recently released which has implications for quantum computing.

PULLMAN, WA, UNITED STATES, April 5, 2017 /EINPresswire.com/ -- Many intelligent people believe that quantum computing as a reality on this planet is within our near future. The field of cryptography is somewhat unique in that probably relatively few people know what is going on in the field as a whole at any given time. That said, the primary codes that are thought to be widely in use for commerce today (wire transfers, internet transactions) are known to be insecure in the presence of a quantum computer.

Quantum computers are not only much faster than conventional computers, but they are capable of running special algorithms or processes that a conventional computer cannot. One of these, Shor's algorithm, is known to be effective against RSA, codes based on the discrete logarithm, and elliptic curve cryptography.

A knapsack code is code based on an old math problem called the knapsack problem. The knapsack problem involved a set of physical objects of different weights, some of which were placed in a knapsack. To solve the problem, someone had to figure out which weights had been used merely from knowing the possible original weights and the weight of the knapsack.

The original knapsack code, which was one of the first public-key or asymmetrical cryptosystems proposed, was vulnerable to some important attacks which presupposed the numbers used were being represented in binary, which is in many ways a very reasonable assumption in the context of conventional computing at least.

The publicly available research that has been done so far would seem to indicate that a secure, lattice-based code would not be vulnerable to attacks by a quantum computer, perhaps because it is not possible to design algorithms that could run on quantum or conventional computers to defeat them.

Lattice-based codes, put as simply as possible, are codes where the solution is a vector in a lattice. A simple example of a lattice is the Cartesian plane we learn to graph lines on in algebra in school.

The generalized knapsack code is one such lattice-based code, and it differs from the original knapsack code in that it relies on many other representations than binary. It is the multiple forms of representation that I believe make it a secure code, though I also believe these matters need to be primarily settled on the basis of physical testing.

As to the other types of non-binary representations, I can give an analogy. You can think of binary as black and white, and base 10 as a palette of 10 colors that can only be used in a very specific way. The other representations may contain any number of colors, with each representation having its own more or less unique way of using the colors. One of the more well-known alternate representations is the Zeckendorf representation, which makes use of the Fibonacci sequence.

The difference between the GKC and the original one is in the approach to the knapsack code used. In the original code, you either used a weight or you didn't, and each choice was relatively independent from the others, except with regard to the size of the knapsack. In the generalized knapsack code, you might use more than one weight of the same size, and you might choose a group of weights linked together rather than just a single one.

My newest paper, which can be found online in the Open Journal of Discrete Mathematics (http://www.scirp.org/Journal/PaperInformation.aspx?PaperID=73743), Number in Mathematical Cryptography, is primarily aimed at helping the scholarly community, and perhaps the business community, understand why the generalized knapsack is likely to be a good option for post-quantum public-key cryptography. A secondary aim is to provide a context in which people from different disciplines can meaningfully discuss some of the deeper issues surrounding number that perhaps have been somewhat compartmentalized in our recent history. I use the word "number" here because one of the reasons my paper was needed is that "number" is very difficult to define, and different people are using slightly different definitions. Here I am concerned primarily with the natural numbers: 1,2,3,4,5,6, etc.

Physicists, engineers, pure and applied mathematicians, and perhaps even philosophers make use of number in their work, and in some respects quantum computing intersects with each of their fields.

This latter concern is for me not a merely academic issue, as in my experience many specialists have certain relatively minor assumptions about numbers which perhaps work rather well within their fields, but do not transfer very well to other related fields. Public-key cryptography was a child of pure mathematics, but engineers, computers scientists and perhaps others as well are very heavily involved in its implementation and maintenance. In my understanding, both quantum computing and the generalized knapsack code require an unusual amount of clarity on these matters, and I am hoping to at least get the discussion going, but I also hope that I have effectively pointed the way as to how these issues might be resolved.

A code like the generalized knapsack code, which could be implemented on both conventional and quantum computers, would in many ways be ideal for navigating the transition to quantum computing as it would require a relative minor change in hardware, if not in software as well.

I am optimistic that these issues will be resolved when they need to be.

Nathan Thomas Hamlin
Washington State University
509 335 0844
email us here
