Keeping Secrets

How to prove a theorem so that no one else can claim it

By IVARS PETERSON

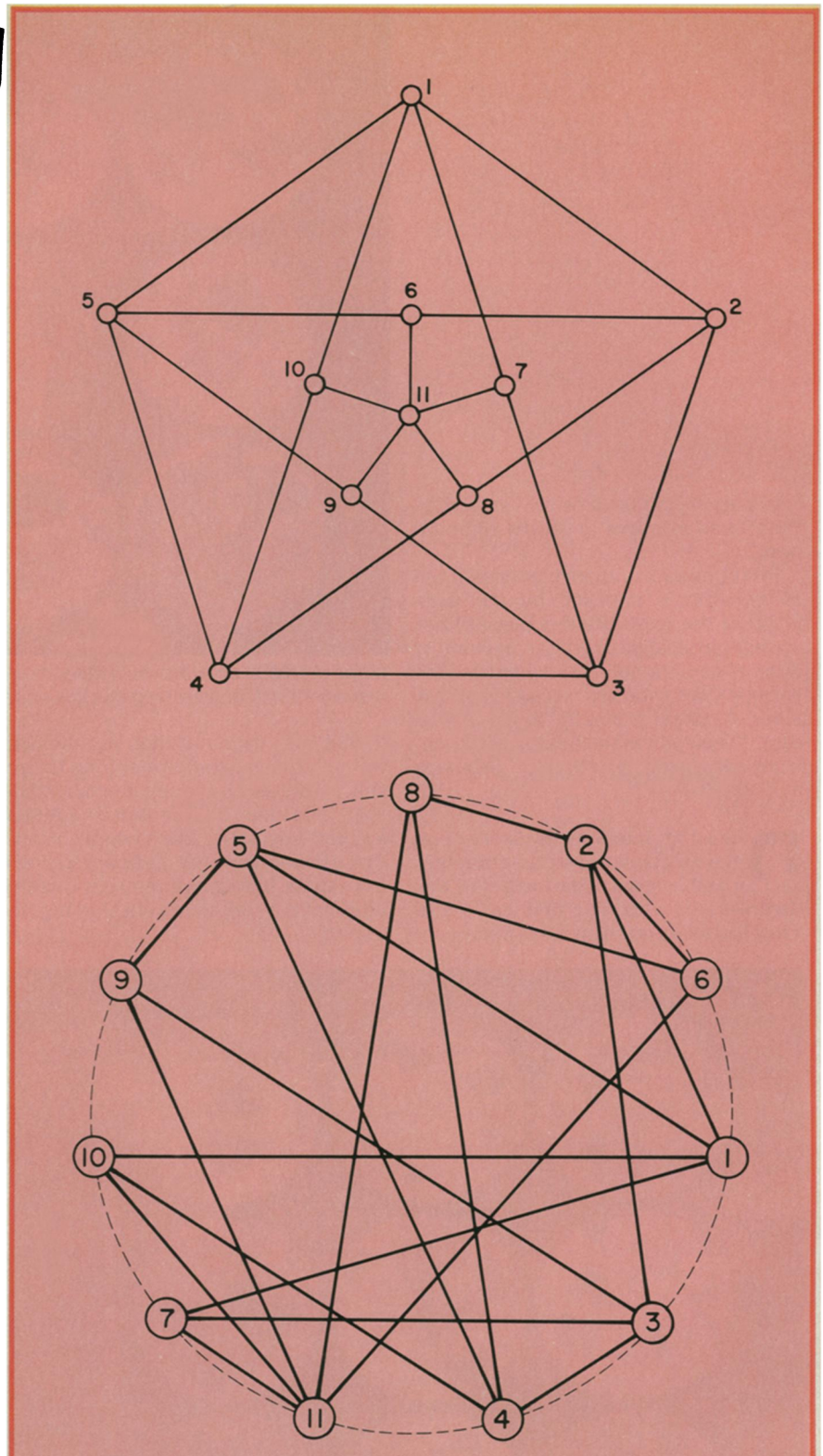
The trouble with sitting down at a computer keyboard to enter a password is that someone may be looking over your shoulder. Because your password could be stolen as you type it in, the computer system isn't completely secure.

But if you could somehow provide the computer with information that persuades the computer you know the password without actually giving away the password itself, then you would be on your way to solving the security problem. Furthermore, if no onlooker could reconstruct the password from the information you gave the computer, then no one could break into the system—at least by using a purloined password.

The mathematical basis for such a scheme has now been found. It depends on something called a "zero-knowledge" proof.

The idea is that a "prover" has found a proof for a theorem and wants to let a "verifier" know that he knows the proof without revealing the proof itself. The verifier can ask a special question that requires the equivalent of a yes or no answer. If the prover really knows the proof, then he can answer the question correctly every time it is asked. If he doesn't know the proof, then the prover has only a 50 percent chance of being right each time. After, say, a dozen tries, the chances of fooling the verifier get very small. Neither the question nor the possible answers give away even a hint of the proof itself – hence, the term zero-knowledge proof.

The concept of zero-knowledge proofs was first defined last year by Shafi Goldwasser and Silvio Micali of the Massachusetts Institute of Technology (MIT)



Deciding whether a particular graph – a network of points and connecting lines – has a Hamiltonian cycle is often difficult. This involves finding a continuous path along the links that passes only once through each of the graph's points and then returns to where it started. In this example (top and bottom), such a path passes through points in the order: 1, 5, 6, 2, 8, 4, 10, 11, 9, 3, 7 and then back to 1. For a zero-knowledge proof, the star-shaped graph (top) can be redrawn so that all of the points fall on the circumference of an imaginary circle (bottom), yet the connecting lines still join the same points as in the original graph.

Illustrations: Arnis Peterson

and Charles Rackoff of the University of Toronto. Earlier this year, Micali, MIT's Oded Goldreich and Avi Wigderson of Hebrew University in Jerusalem took a major step forward by showing that such a scheme works for a large class of mathematical theorems. They demonstrated the procedure for a mathematical coloring problem, which involves ensuring that no two points in certain networks of connected points have the same color.

Recently, Manuel Blum of the University of California at Berkeley showed how to give an efficient zero-knowledge proof for any mathematical theorem. He refined the earlier work and simplified the overall procedure. Blum described his work at this month's International Congress of Mathematicians, held in Berkeley.

One of the more difficult steps, says Blum, is finding the right question for the verifier to ask. Once this is done, a zero-knowledge scheme can handle any theorem provable within any logic system. All a verifier finds out is that a theorem is provable and that the prover actually knows the proof. And because the prover has to use at least as many words as the proof itself contains, he gives away an upper limit for the proof's length.

To show how the scheme works, Blum chose an example from graph theory. Any network of points (or nodes) connected by lines (or edges) is called a graph. In Blum's example, the graph consists of a star-shaped pattern of lines linking 11 points.

The prover has found a continuous path along the connecting links that passes only once through each of the 11 points on the graph and returns to where it started. This special type of path is called a Hamiltonian cycle. The prover's aim is to persuade a verifier that such a path is known without giving the verifier the slightest idea of how to construct the path.

To do this, the prover privately marks 11 nodes along the circumference of a circle and labels them randomly from one to 11. Then the nodes on the circle are connected in the same way as the points in the original graph. That is, lines would join nodes one and five, two and six, and so on. Now the resulting diagram is covered up by, say, an erasable opaque film like that used on some lottery or contest tickets.

The verifier can ask the prover to uncover the complete graph, which shows that all the points are properly linked, or she can ask to see the Hamiltonian cycle. In the latter case, the prover erases enough of the film to reveal just the lines that make up the cycle. He can do this only if he knows the right path. However, because the nodes are still covered up, the verifier doesn't know the actual path from point to point. All she can verify is that a Hamiltonian cycle exists.

This process can be repeated as many times as the verifier wishes. Each time, the prover sets up a new circle diagram, which is then hidden. Because he doesn't know whether the verifier will ask for the graph or the cycle, he has to be ready for either choice and therefore must know the cycle. Failure to produce either the correct graph or the cycle during any turn is equivalent to a wrong answer, and the verifier then knows that either the prover is lying or he doesn't actually have the proof.

As outlined, this scheme sounds somewhat cumbersome. But the opaque film used in the example can be replaced by encryption schemes that hide information. Thus, proof checking can be done electronically when the whole procedure is encoded as strings of binary digits. This makes it possible to use this concept for password protocols and in cryptological games like tossing a coin by telephone or exchanging secret keys (SN:9/26/81,p.205).

And in the sometimes turbulent world of mathematics research, it gives a wary mathematician a way to claim credit for being the first to find a particular proof without the necessity of giving away the proof's details. All that someone else can find out, until the proof itself is finally revealed, is that a particular theorem is provable. □