Can you get rich off a 232-year-old mathematical equation? Some entrepreneurs specializing in computer encryption are going to try—if they can stop squabbling long enough to divide the spoils.

Patented secrecy

By Simson L. Garfinkel

In 1763 a Swiss-born mathematical genius by the name Leonhard Euler came up with an equation that describes what kind of remainders you get when you divide whole numbers of a certain kind. For the next two centuries Euler's equation was the plaything of mathematicians—a starting point for academic researches into the abstractions of number theory. Then, in 1977, Euler's discovery turned into something extremely valuable in the commercial world. It became the basis of a system of encryption, that is, a technique for turning confidential messages into gibberish comprehensible only to the intended recipient.

Secret codes go back to the time of Julius Caesar, if not earlier; they have played an important role in war and diplomacy since then. But commercial encryption has gained new importance in the modern digital age. Encryption protects your password as it is transmitted from an automatic teller to a bank computer; it keeps crooks from stealing money by forging bank wire transfers; it enables television show owners to collect from people who own satellites; it may someday give rise to a thriving marketplace on the Internet. Encryption also makes possible virtually untappable phones—very useful to businessmen, frightening to law enforcers.

If computers create the demand for encryption, they are also the solution. To make a code uncrackable, or nearly so, you have to make it complicated. Cheap microprocessor chips are there to do the arithmetic. They can handle the millions of calculations typically necessary to encode and decode a secret message on the fly.

The 1977 application of Euler's mathematics to encryption was the work of three professors at the Massachusetts Institute of Technology: Ronald Rivest, Adi Shamir and Leonard Adleman. Sensing the commercial import of their work, MIT patented their coding formulas. The university then licensed the patent to a newly formed company, RSA Data Security.

The beauty of the RSA coding scheme is a feature that Caesar would scarcely have imagined possible: a public key. The key is the formula that translates the plaintext message into the encoded gibberish. If, for example, your code moves every letter three places forward in the alphabet (so A becomes D and Z becomes C), then the number of places moved is the key. The key is at once an encoding and decoding device.

In conventional encryption, keys must be shared by the sender and receiver of the message, and they must be kept secret. That is the great weakness of conventional encryption. What messenger can be trusted? If he is compromised or his codebook stolen and copied, the code is worthless. In a digital/wireless age, this danger is immense.

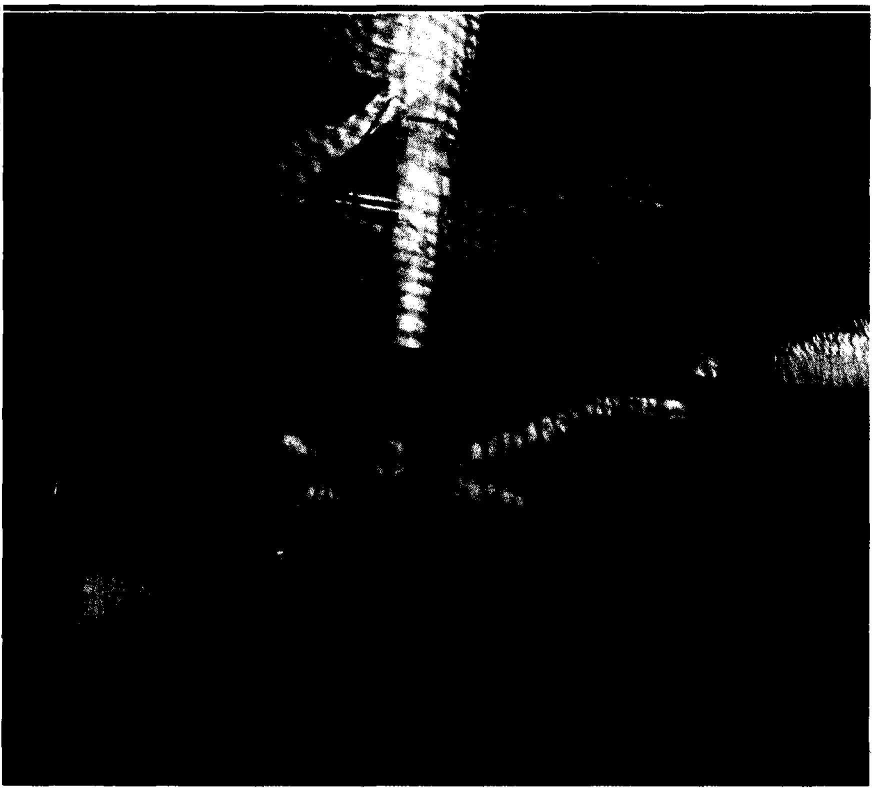
In RSA encryption, the key that encodes is published for all to see, friend and foe alike. That key is creat-

President James Bidzos of RSA Data Security

If you use Lotus Notes, he's collecting royalties on you.

Simson L. Garfinkel is the author of *Pretty Good Privacy: Encryption for Everyone* (O'Reilly & Associates, 1995).

Photographs by Eric Millette



Jimmy Omura, chairman of Cylink

If someone is eavesdropping, call him.

ed by the person who wants to receive confidential messages. A different key decodes messages, and this key is known only to the receiver. He calculates this key from certain arithmetic facts—facts he keeps to himself—about the published encoding key. The mathematics of this system is such that the public key gives no clue as to how to construct the secret decoding key *(see box, p. 124)*.

Besides keeping secrets secret, public key cryptography will have another big payoff within the next few years. This is the closely related technology of digital signatures. Simply stated, a digital signature is public key cryptography run in reverse. Instead of making secret messages, the math creates an unforgeable electronic seal that can be placed at the bottom of an electronic document. It could be anything digitized—a memo, a purchase order, a tax return, even a photograph. The seal can be used to check if the document has ever been modified since it was first sealed; it also proves the identity of the person who signed it—since only he or she would be able to make the seal.

Clearly, RSA Data Security has something extremely valuable in its encryption patent. But it does not have this field to itself. Indeed, the MIT professors did not even invent the concept of public keys; that distinction goes to two other academics. Shortly before Rivest, Shamir and Adleman invented their system, Stanford professor Martin Hellman and graduate student Whitfield Diffie had published a different system of public key cryptography. Like MIT, Stanford knew it had something valuable, and won patents. It later licensed the patents to newly formed Cylink Corp. in Sunnyvale, Calif.

Cylink has used the Stanford patents to become, by its reckoning, the world's largest supplier of commercial secure communications equipment, with 200 employees doing sales of some $30 million. "We had a customer who was bidding against foreign competitors for huge projects in foreign countries," recalls Jimmy Omura, 54, Cylink's founder and chairman. "They knew that their lines were being tapped because their bids were consistently underbid by a very small amount." Cylink's product put an end to the problem.

Rival RSA—headquartered a few miles from Cylink in Redwood City—hired James Bidzos, a former international technology broker, as president and focused on selling algorithms to software companies. Bidzos, 40, is a good salesman. RSA encryption technology can be found inside more than 300 products, including Lotus Notes, Novell NetWare and Apple's Macintosh operating system. RSA employs 35 people and has annual revenues between $5 million and $10 million.

The federal government is in the middle of the fray. It wants to be able to receive E-mail that is secret and/or contains an unforgeable digital signature—think, for example, of tax returns. So Uncle Sam wants public key encryption to be widely available. But it doesn't want the technology to be too good. Terrorists and drug smugglers could use it to make their phones untappable.

It may be too late to put the genie back in the bottle. Already, for less than $200 you can buy software that will turn a multimedia personal computer into an encrypting telephone that will thwart any eavesdropper, the FBI included.

Clearly there is a big business in encryption. What is not clear is who, if anyone, will collect the big royalties on it in coming years. To begin with, the MIT patent is an improvement on the Stanford ideas, muddying mathematical and legal waters.

Rather than fight, in 1990 Cylink and RSA pooled their patents into a partnership and went about their mostly separate lines of business, together telling potential users not to touch their patented technology without a license.

Uncle Sam doesn't want encryption to be too good. Terrorists could use it.

The federal government, meanwhile, was getting very interested in digital signatures as a means of receiving official government filings. In 1991 the National Institute of Standards & Technology, an arm of the Commerce Department, issued its initial draft for a federal digital signature standard. Instead of using RSA digital signatures, which were becoming a de facto worldwide standard, the feds chose a new algorithm designed in secret by the National Security Agency, the spy agency headquartered at Fort Meade, Md.

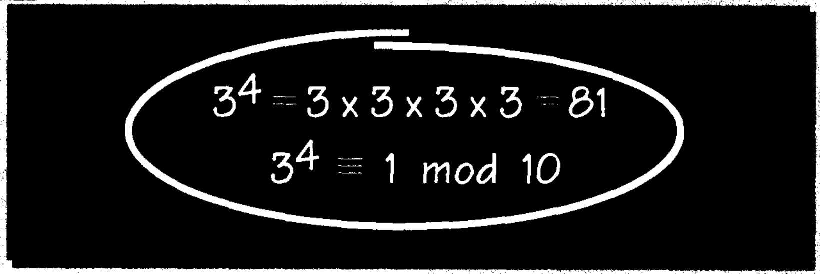
Did the NSA algorithm infringe the Cylink/RSA patents? Federal officials said it didn't. But RSA's Bidzos, who is also president of the Cylink/RSA patent partnership, argued otherwise. Anybody who used the NSA algorithm, he said, risked an expensive and lengthy patent litigation. But the partnership and Commerce came up with a deal: Give the RSA/Cylink partnership an exclusive license for the NSA algorithm, and the partners will give the government free use and license nongovernment users at no more than $1 per key per year, plus certain royalties on products using encryption. So if 10 million taxpayers had signed up for electronic filing of tax returns, the RSA/Cylink partnership could have raked in $10 million a year in royalties.

At first government negotiators agreed to the deal, but after a torrent of public objections they decided the private patent holders were asking for too much. Last May the Department of Commerce declared the NSA formula was officially available, and in October said if anyone got sued for using it to satisfy a government contract, the government would help defend the suit.

By then, long-simmering disagreements between RSA and Cylink had boiled to the surface. RSA threatened to sue Cylink for patent infringement. In a preemptive strike last June, Cylink sued RSA, alleging that the RSA patent is invalid. Now both are trying to have the partnership dissolved.

What if Cylink were right that the RSA patent is unenforceable? That would mean, retorts RSA's Bidzos, that Cylink is guilty of collecting patent license fees on a patent it believed to be invalid. "I think they have stuck their foot into something they are finding it very difficult to extricate themselves from," he says.

In the end, there is probably nothing to stop encryption of one sort or another from becoming ubiquitous in the computer industry. But it won't be as lucrative to the original purveyors as it might have been. ■



The key that locks does not unlock

PUBLIC KEY cryptography is a clever scheme for encoding secret messages with encryption keys that are known to the public.

Think of it this way. Anyone can send you a message in a locked box. Copies of the key that will *lock* the box are widely available. But once the lock is snapped shut, it takes a special key to ***unlock*** the box. The key that opens doesn't look anything like the one that locks. There is only one copy of the opening key, and you, the recipient, have it.

The first step in encoding a message is for the sender to convert it into a number. So, "Transfer $10 million to my Swiss bank account" becomes a long string of digits.

Next, the sender raises this large number to an exponent. In the first equation on the blackboard above, 4 is the exponent. It means that 3 is to be multiplied by itself 4 times. In a real-life case, the message number—maybe hundreds of digits long—would go where the 3 is.

Now the sender does some modular—that is, remainder—arithmetic. To say that 81 equals 1 modulo 10 is to say that when you divide 81 by 10 you get a remainder of 1.

Why remainder arithmetic? Because it does such a wonderful job of scrambling numbers.

The public key consists of two numbers—the exponent and the modulo. The scrambled message that is sent along is the remainder—comparable to the 1 in this example.

Okay. We've now used encryption formulas known to everyone to scramble a message and send it to a receiver. How does the recipient read it?

The secret unscrambling key is another exponent. The message receiver calculates this exponent from some other, secret numbers. Lacking these numbers, a hacker would need thousands of years on a good computer to break the code.

Where does Leonhard Euler fit in? In 1763 he devised an elegant little equation about exponents and modular arithmetic. That equation is vital to calculating the decrypting key. We don't show the equation here, but it looks a lot like the second line on the blackboard.

Euler's math became useful for cryptography only with the advent of cheap, powerful computers. Without a computer, you can't do this kind of encoding or decoding. The computations are way too large. Indeed, even PCs need to take shortcuts. If they didn't, one of the numbers would be so large that its digits wouldn't fit into a computer memory the size of the universe. ■