Zero Knowledge and the Department of Defense

Susan Landau

The game is simple and apparently paradoxical: Prove you know something—an ID number, an access code—without revealing even a single bit of the information itself. The importance is obvious: from credit card numbers to computer passwords, we are increasingly reliant on the secure electronic transmission of what are known as signatures. Yet how can one transmit a signature without potentially revealing to an eavesdropper—or an unethical vendor—all the information he needs in order to masquerade as the sender?

Three Israeli computer scientists—Uriel Feige, Amos Fiat and Adi Shamir, of the Weizmann Institute—figured out how to play the game, called "zero knowledge proofs of identity". They publicized their result at conferences, and they applied for U.S. patent protection. Ironically the United States said disclosure was "detrimental to the national security," and imposed a secrecy order. The three Israelis sought relief, and, with intervention from powerful sources, they got it. Though no one will say for certain, it appears that the National Security Agency (NSA), the government decrypter of secrets, stepped in to help. What the research is, and why the NSA had reason to involve itself, is the story we present here.

The technical part has its genesis in the work of Stephen Cook and Richard Karp of the early seventies. Our model of a computer is a RAM, a Random Access Machine. At issue is complexity: on a problem of input size "m", how many steps does it take as a function of m to solve the problem? Certain problems are "easy"; by the obvious method, two $m \times m$ matrices can be multiplied in $O(m^3)$ steps (although there are considerably more sophisticated algorithms which require only $O(m^{2.376})$ steps). Other problems are less obvious. The crucial distinction comes between those problems with polynomial time solutions (the class P), and those which require more than polynomial time. The latter are considered infeasible.

What Cook did was to show that the question "Is a boolean expression satisfiable?" occupies a special place in the hierarchy of problems. It is solvable in polynomial time by a nondeterministic RAM[1] (it is in NP, Nondeterministic Polynomial time), and it is as hard as any other problem in NP (it is complete). If satisfiability has a polynomial time solution, so will any other problem in NP. Karp then showed that a number of combinatorial problems shared that characteristic, called NP-completeness, including k-colorability (can the vertices of a given undirected graph be colored with k colors so that no two adjacent vertices have the same color?), knapsack (given a finite set of integers n_i, is there a subset which sums to an integer K?),

This article is the seventeenth in the series of Special Articles published in the *Notices*. The author, Susan Landau, is an assistant professor of computer science at Weslyan University. She received her Ph. D from M.I.T.; her thesis gave a polynomial time method for determining solvability by radicals. Her research interests include computational complexity and algebra.

[1] A nondeterministic RAM can choose which of two instructions to perform at certain times. For example, by choosing 0 or 1 n times, it can "guess" a boolean assignment and then check if that assignment makes a given boolean expression true.

and Hamiltonian circuit (does a given undirected graph have a Hamiltonian circuit?). Most theoretical computer scientists believe that the NP-complete problems require exponential time, but despite fifteen years of concentrated attack, the question "Does P = NP?" remains open.

Where number theoretic questions fit into this hierarchy is unknown. In 1974, using the Extended Riemann Hypothesis (ERH), Gary Miller showed how to test the primality of an m digit integer in $O(m^5)$ steps. Robert Solovay and Volker Strassen gave a probabilistic algorithm which avoids the use of ERH and takes only $O(m^4)$ steps. Meanwhile, the present best algorithm for factoring an m digit integer requires $O(e^{c\sqrt{m \log m}})$ steps. The apparent discrepancy between the complexities of these two problems led to the first public key cryptosystem.

This is a system in which the method of encryption is publicly known, but decryption is infeasible because of contrasts in complexity. It was developed as a way to ensure secure communication between two parties while allowing changes of code (by publicly publishing it). The first public key cryptosystem was RSA, named after its inventors, Ron Rivest, Adi Shamir and Len Adleman. Although no one has proved it secure, Michael Rabin has shown that a variation of it is, provided that factoring remains infeasible.

Because RSA forms the basis for much of the following work, we briefly describe it. Each participant in the system finds two large ($\sim 10^{50}$) primes, p and q, and publishes two integers: n $(= pq)$ and a, where a is less than n and relatively prime to $\phi(n)$ $(= (p-1)(q-1))$, the Euler phi-function of n.

Suppose the Federal Reserve wants to communicate with the Bank of England. The Fed would proceed as follows:

(1) Translate the message into integers, say A=01, B=02, etc.

(2) Break the message into blocks of a convenient length.

(3) Consult the public key book for the Bank of England's n and a.

(4) Send each block as $(block)^a$ (mod n).

Decryption is easy for the Bank of England because it knows the factorization of n. Since a and $\phi(n)$ are relatively prime, there are x and y—polynomially time computable from a and $\phi(n)$—such that $ax + \phi(n)y = 1$. By Euler's theorem $((block)^a)^x \equiv block$ (mod n), so that the decrypted message can be determined in polynomial time. No one else knows n's factorization, and determining *block* (mod n) any other way appears to be computationally difficult.

A particularly useful aspect of the RSA scheme is that one can "sign" a message. For example, by encoding first using the Federal Reserve's secret decryption algorithm, and then the Bank of England's publicly available encryption method, the Federal Reserve can sign its message. Since the Federal Reserve Bank's encryption scheme is public information, the Bank of England decodes the Fed's message by employing first its own (private) decryption method, and then the publicly available Federal Reserve encryption scheme.

As long as factoring remains difficult, the method appears secure. Yet, sending the encrypted message, the Federal Reserve has, of course, informed the Bank of England what its encoded form of the message is. If the message was simply a credit card number, then an unscrupulous recipient now has all the information he needs in order to later misrepresent himself as the Fed. But how can you prove that you are who you say you are without revealing any information?

In 1985, Shafi Goldwasser, Silvio Micali and Charles Rackoff [GMR] suggested the concept of zero knowledge interactive proofs. In such a proof, the "prover" has infinite computational power, while the "verifier" is a probabilistic polynomial time machine. The prover "proves" a fact to the verifier; if the "fact" is true, the probability that the verifier accepts it is high ($1 - 1/2^n$ for an n-bit statement), if it is false, the probability of rejection is correspondingly high. The entire interactive proof is limited to polynomial time.

The zero knowledge part of the proof is that no other information is given. More precisely, an interactive proof is zero knowledge if the verifier cannot gain any information from the prover that he could not himself derive in expected polynomial time. **[GMR]** showed that certain sets in NP $\bigcap$ co-NP (sets whose complement are in NP) have zero knowledge interactive proofs.

We begin with some notation:

$\overline{\mathrm{P}}$ (straight P) represents the real prover who follows the designated protocol.

$\widetilde{\mathrm{P}}$ (crooked P) represents a cheater who can deviate from the protocol in an arbitrary way.

P represents either $\overline{\mathrm{P}}$ or $\widetilde{\mathrm{P}}$.

$\overline{\mathrm{V}}$ (straight V) represent the real verifier who follows the designated protocol.

$\widetilde{\mathrm{V}}$ (crooked V) represents an arbitrary program which tries to extract additional information from $\overline{\mathrm{P}}$.

V represents either $\overline{\mathrm{V}}$ or $\widetilde{\mathrm{V}}$.

In an example given by **[GMR]**, the language that P can recognize is $Q_n = \{y \in (Z/nZ)^* \mid$ the Jacobi symbol of y on n is 1, but y is not a square mod $n\}$, where n is an odd composite integer. V generates a sequence of integers $x_1, x_2, \ldots, x_m$ (for $m = \log n$), where each x_i is of the form: r_i^2 (mod n) or $x \equiv yr_i^2$ (mod n), with each r_i chosen randomly. V quizzes his adversary about the sequence.

If y is in Q_n, P distinguishes between the two cases. The tricky part of the protocol is that P communicates that information without revealing anything else to V. That is, P responds only to those queries about which V has convinced P that V has the answer. With high probability, $\widetilde{\mathrm{V}}$ is not able to falsely convince P that $\widetilde{\mathrm{V}}$ has that information. However the name "zero knowledge" is somewhat misleading since P does reveal to V—and any eavesdropper—when y does belong to Q_n. The protocol is rather complicated, and so we do not go into further detail here.

The importance of zero knowledge interactive proofs is in making cryptographic schemes secure. Normally it is a very hard problem. Zero knowledge gives a way to break up the components of a protocol, and prove that no information is transferred between the pieces.

Shortly afterwards, Oded Goldreich, Micali and Avi Wigderson showed if there is any secure probabilistic encryption scheme, then there are zero knowledge proofs for the colorability problem, and thus for any other NP problem. One can show that a graph is 3-colorable without revealing any other information about the graph, including such facts as whether vertex 1 and vertex 3 are colored the same. The proof is simple, and we provide a sketch.

Suppose P wants to show that a certain graph can be 3-colored. P transmits the adjacency matrix of the graph. He also transmits an encrypted version of the coloring, for example by sending for each vertex v the RSA encryption of the string $r_v c_v$, which consists of a 75 bit random integer (r_v), followed by a 2 bit integer for the color (c_v). The reason for including r_v in the communication is to make it computationally infeasible for V to determine the coloring.

V picks an edge and asks what the colors of its two vertices are. P's encryption function is known to V, so what P reveals to V is the "decrypted" coloring of the two vertices. V encrypts and checks that they match the information P had originally sent.

Now P changes the coloring (say red vertices are changed to yellow, yellow vertices to blue, and blue to red), and invites a new query. The proof repeats. The point is that after polynomially many tries, if V hasn't found an edge connecting two vertices of the same color, he believes that P has a 3-coloring of the graph. However the constant changing of the coloring scheme sufficiently conceals the information so that V has not learned anything about the coloring that he himself could not have computed in probabilistic polynomial time.

Meanwhile Zvi Galil, Stuart Haber and Moti Yung proposed an alternate model: result indistinguishable protocols. In their model, either P proves that an element belongs to a language, or P proves that the element does not belong to the language. V knows which claim is being shown, but the eavesdropper, who cannot query either V or P, is unable to determine from the computation which result is being proved. They used a number theoretic language for the protocol.

Feige, Fiat and Shamir [FFS] modified the rules. They wanted zero knowledge interactive proofs to be truly zero knowledge. The proof should not give away to anyone even the single bit of information about membership. They achieved this goal by changing the notion of zero knowledge proofs of membership to zero knowledge proofs of knowledge. They showed that any set in NP $\bigcap$ co-NP has a zero knowledge interactive proof under the new definition.

Current identification schemes, whether ID cards, computer passwords, or PIN numbers, have the prover P identify himself by presenting a predetermined fixed secret code. This gives no protection against an eavesdropper or dishonest verifier. Ideally, an identification scheme should be a protocol in which P proves his identity to V without enabling a corrupt V to later misrepresent himself as P. [FFS] invented such a secure scheme: they gave a protocol which was zero knowledge under the new definition, secure as long as factoring was difficult, and which is fast—two orders of magnitude faster than RSA-based schemes. It is practical—eminently so. We present it.

As is frequently the case, the scheme relies on the intractability of a number theory problem. If a and b are two integers less than n, with $a^2 \equiv b^2 \pmod{n}$, but $a \not\equiv \pm b \pmod{n}$, then both $a + b$ and $a - b$ share a nontrivial gcd with n. This is the basis for a number of factoring algorithms. It also shows that computing "square roots" mod n is computationally as difficult as factoring.

In this situation both the prover and the verifier have probabilistic polynomial time machines at their disposal. Let p, q be two distinct, large ($\sim 10^{75}$) primes, both of which are congruent to 3 (mod 4), and let $n = pq$. Integers of this form, known as Blum integers, are useful because -1 is a quadratic nonresidue mod n whose Jacobi symbol is $+1$. The integer n is public information. Everyone uses the same n (whose factorization is unknown).

The prover, P, chooses k integers s_j, relatively prime to n, which he keeps secret. Let the notation $c \equiv \pm d \pmod{n}$ mean that c is randomly chosen to be congruent to either d or $-d \pmod{n}$. P publishes $i_j \equiv \pm s_j^{-2}$, for $j = 1, \dots, k$. (The reason for the inverses is to speed the transaction; the protocol would work as well, though less quickly, with $i_j \equiv \pm s_j^2$.) Then, if no cheating is detected, steps (1) – (4) are repeated t times:

(1) P picks a random r in Z/nZ and sends $x \equiv \pm r^2 \pmod{n}$.

(2) V sends a random boolean vector $\bar{e} = (e_1, \dots, e_k)$ of 0's and 1's to P.

(3) P sends $y \equiv r\Pi_{e_j=1} s_j \pmod{n}$ to V.

(4) V checks that $x \equiv \pm y^2\Pi_{e_j=1} i_j$ (mod n).

Feige, Fiat and Shamir showed that the protocol proves that P knows the s_j's without giving away *any* information when $k = O(\log\log n)$ and $t = O(\log n)$.

Note that if P follows the protocol, then

$$y^2\Pi_{e_j=1} i_j \equiv (r\Pi_{e_j=1} s_j)^2\Pi_{e_j=1} i_j \equiv \pm r^2\Pi_{e_j=1}(s_j^2 i_j) \equiv \pm r^2 \equiv x \text{ (mod } n).$$

Thus V will verify the computation in step (4). It is not hard to show—although we will not—that there is a RAM M such that if $\tilde{\mathrm{P}}$ can cheat V with a non-negligible probability (say $\geq 1/n^c$, for some constant c), then M can produce $s_1, \ldots, s_k$, in probabilistic polynomial time. That is, M, with $\tilde{\mathrm{P}}$'s aid, could factor in probabilistic polynomial time.

Eavesdropping on a $\overline{\mathrm{V}}$-$\overline{\mathrm{V}}$ interaction gives no more information than first generating a set of vectors $\{\overline{e}\}$ and then simulating an identical interaction, since those two probability distributions are identical. Hence the proof is zero knowledge.

Furthermore, even a crooked verifer gains no information from the interaction. $\tilde{\mathrm{V}}$ cannot hope to use the same interaction in a proof of identity, since he cannot anticipate the vectors $\{\overline{e}\}$ about which he would be queried during a proof. Even if he chooses a special set of vectors $\{\overline{e}\}$ (say $\overline{e}_1 = (1, 0, \ldots, 0), \overline{e}_2 = (0, 1, 0, \ldots, 0), \ldots, \overline{e}_k = (0, \ldots, 0, 1)$), the choice of r in each round is sufficient to mask any information that the cheater is hoping to glean.

The result is striking. "Smart" cards which contain a computer chip can be programmed to conduct the protocol. Essentially they prove: "This is a valid card. My name is A** S*****." A public national registry stores the value of "n". At the time you sign the card, you enter the k integers s_j, and publish the k integers i_j. From then on, anyone who witnesses you using the card gains no information. Assuming factoring is hard, one can make unforgeable ID cards. Theft, of course, remains a problem.

The work clearly has commercial applications. On July 9, 1986 the three authors submitted a U.S. patent application. The Patent Office had six months to respond with a secrecy order. At the request of the U.S. Army, on January 6, 1987—three days before the end of the six-month period—the Patents and Trademarks Office imposed the order, informing Shamir that "... disclosure or publication of the subject matter ... would be detrimental to the national security... " Shamir et al were ordered to notify all Americans to whom the research had been disclosed that unauthorized disclosure could lead to two years imprisonment, or a ten thousand dollar fine, or both. Furthermore, Shamir, Feige and Fiat were to inform the Commissioner of Patents and Trademarks of all foreign nationals to whom the information had been disclosed.

The horse had long since fled the barn. Throughout the summer of 1986, Shamir and his coauthors had given talks about the research at universities in Israel, Europe and the United States. They had presented it at the International Congress of Mathematicians in Berkeley, and at the CRYPTO 86 conference at Santa Barbara a week later. They had also submitted a paper to the Association of Computing Machinery (ACM) conference on the Theory of Computing to be held in New York in May 1987.

Shamir wrote the program committee of the ACM conference, informing them of the secrecy order, and their consequent legal obligations (" [Destroy] all copies of the paper made during the refereeing process, and ... [warn] all people involved about the secrecy order ... ") and asking their advice. He mentioned that the Weizmann Institute would likely appeal the order: the work had been performed in Israel, by Israeli scientists, with Israeli funding, but he wanted to know what to do if it were not removed in time for the publication or the presentation dates.

An informal network swung into action. The program committee told colleagues. Several Bell Labs scientists made some well-placed calls. A *New York Times* reporter was informed, and he prepared a front page story. But the response with the most impact seems to have come from an agency which refuses to acknowledge that it had any role in the affair: the National Security Agency.

The NSA was chartered in 1952. Its mission includes devising codes for the military as well as cryptoanalysis of foreign intelligence. It is possibly the world's largest employer of cryptologists. No one knows for certain; like most information about the agency, the number of employees is classified.

For most of its existence, the agency faced little competition from research published by industry or academia. The introduction of computers into all facets of modern life changed that. Banks, business, industry must be able to transfer securely large amounts of information; hence cryptography flourished in the private sector and in openly published research.

Early on, there were several public squabbles between the NSA and academic researchers. In 1977, Shamir and his two coresearchers at MIT sought to present the RSA scheme at a conference at Cornell. An employee of the NSA—the agency claims that he acted on his own—warned that doing so was in possible violation of a 1954 Munitions Control Act. In 1980, Adleman was denied NSF funding because the NSA feared certain "national security implications".

Both cases were later settled. Bobby Inman, at the time director of the NSA, told the academic community that open publication was harmful to the NSA's mission, and that he sought a dialogue regarding the publication of cryptography research. He warned that if an agreement could not be reached, the agency might seek legislative relief.

On his urgings, the American Council on Education formed the Public Crytography Study Group (PCSG). In 1981, after a year of meetings, the group proposed guidelines under which members of the academic community would voluntarily submit cryptography papers to the NSA prior to publication. The agency might suggest changes, including deletion. The NSA might even ask that the paper not be published.

George Davida, a member of the panel, and professor of computer science at the University of Wisconsin, disagreed with the guidelines. He felt that the greater risk to national security was the vulnerability of the private sector to electronic espionage. NSA needs were outweighed by the need for secure cryptographic schemes within the private sector. "NSA's efforts to control cryptography [are] unnecessary, divisive, wasteful," said Davida, "It is only by allowing progress in the field ... that the NSA will remain effective."

Davida had his own reasons for distrusting governmental secrecy. In 1977, the Wisconsin Alumni Research Foundation, acting on his behalf, had submitted a patent application for an encryption device which would protect computers from unauthorized penetration. Several months later, the Patent and Trademarks Office informed Davida that the device was protected by an Invention Secrecy Order. This came at the request of the NSA.

Davida protested. The chancellor of the University of Wisconsin appealed to the NSF (which had funded the research). The press—local and national—played up the story. Two months later the order was rescinded with the explanation that it was all a "bureaucratic snafu".

Since 1917 the United States has had laws which permit the government to classify private ideas. The reason was to protect the country at war, and the grounds were whenever "... publication might be detrimental to the public safety or defense ..." In a recent (1979) review of the patent secrecy system, Congress expressed concern,"The invention secrecy enterprise ... conflicts with the principles of the patent system ... Invention secrecy ... is heavily weighted against private inventors who work outside the classified and defense community ... It gives those nonmember inventors the choice of presenting their discoveries to the public without ownership protection, or of trying to obtain a patent and thereby risking Government confiscation of their ideas."

The current Secrecy Act dates from 1951. Under it, secrecy orders are issued by the Commissioner of Patents and Trademarks at the request of a defense agency. Normally the orders must be renewed each year, except in times of national emergency. (The Korean War emergency, declared in 1950, lasted until 1978.) Recently the Patent Office has issued about 350 new secrecy orders annually. From these, the Patent Office typically receives about fifty petitions for some relief—not necessarily total—of the order.

Secrecy orders are subject to little review. The head of a defense agency makes a request, and the Patent Office is compelled to impose it. Despite the potential importance of these actions, the defense agencies have often relegated the decision-making to the ranks.

In the [FFS] case, it really does appear that there was a bureaucratic snafu. The patent application included the phrase "... potential military applications ...". That meant the Patent Office had to send it to all the defense agencies for examination. When the Army requested a secrecy order, the Patent Office was required to comply. Yet if the Army evaluators had known the work was by non-U.S. citizens—the official reason for the removal of the order—or that it had already been publicized, they would not have asked for a secrecy order, according to Department of Defense spokesmen.

The NSA cryptologists did know the relevant facts, but the agency was not informed of the secrecy order before it was imposed. When American computer scientists became aware of the order, several of them made calls to the NSA. So did the *New York Times* reporter. Within two days the secrecy order was rescinded (although Shamir was not officially informed until a month and a half later, three days before the conference publication date). Shamir and others involved are convinced that the agency pulled strings to have the order removed. The NSA predictably had "no comment".

In one sense, the story is very simple. The Army mistakenly requested a secrecy order. It was removed. Even without NSA involvement, it is likely the order would have been rescinded. The interesting issue is why the agency may have intervened.

The NSA has had a relatively comfortable relationship with the theoretical computer science community since the PCSG report. By 1982, thirty-five papers had been submitted to the NSA, and changes and deletions were requested in two. Now over five hundred papers have been submitted, and although the agency will no longer give precise figures, it does say that changes were asked for in "a small number". On at least one occasion, a researcher did not publish work because of an NSA request.

Some computer scientists find even the voluntary submission of papers to have a chilling effect on research, and others are disturbed by the increasing proportion of funding from military sources. Yet there does not appear to be the tension between the two communities that there was in the late seventies and early eighties. The theoretical computer science community has not found its early fears of suppression of research and loss of civilian funding to be realized. The NSA would prefer to keep things comfortable; after all, it is much easier to find out what the competition is doing if they send you their papers.

Would the cryptography community be better off if only the NSA did patent reviews? Probably not. In this case, it seems that the NSA argued against an Army secrecy order; some other time the situation might be reversed. Many years ago, Dwight Eisenhower argued against concentrating power in a single military command because its uncontested voice would carry an inordinate weight in civilian affairs. It is likely that the research community benefits from the diversity of agencies performing reviews.

Shamir did give his talk on May 26 at the ACM conference. About three hundred computer scientists attended. He thanked anonymous readers (viz the Army) of an early version of the paper for being so counterproductive, and said that " ... the NSA guys ... were extremely helpful behind the scenes in removing the order ... . "

References

[PP] J. Bamford, *The Puzzle Palace*, Houghton Mifflin Co., Boston, MA (1982).

[BC] G. Brassard, and C. Crepeau, *Non-transitive transfer of confidence: a perfect zero knowledge interactive protocol for SAT and beyond*, Proc. Twenty-Seventh Annual Sympos. on Foundations of Computer Science (1986), pp.187-195.

[Cook] S. Cook, *The complexity of theorem proving procedures*, Proc. 3rd Annual ACM Sympos. on Theory of Computing (1971), pp. 151-158.

[CW] D. Coppersmith, and S. Winograd, *Matrix multiplication via arithmetic progression*, Proc. 19th Annual ACM Sympos. on Theory of Computing (1987), pp.1-6.

[FFS] U. Feige, A. Fiat and A. Shamir, *Zero knowledge proofs of identity*, Proc. 19th Annual ACM Sympos. on Theory of Computing (1987), pp.210-217.

[FS] A. Fiat and A. Shamir, *How to prove yourself: Practical solutions to identification and signature problems*, Proc. CRYPTO 86 Conf. (Santa Barbara, August 1986).

[GHY] Z. Galil, S. Haber and M. Yung, *A private interactive test of a boolean predicate and minimum-knowledge public key cryptosystems*, Proc. 26th Annual Sympos. on Foundations of Computer Science (1985), pp. 360-371.

[Gleick] J. Gleick, *A new approach to protecting secrets is discovered*, The New York Times, February 17, 1987, p. C1.

[GMW] O. Goldreich, S. Micali, and A.Wigderson, *Proofs that yield nothing but their validity and a methodology of cryptographic protocol design*, Proc. 27th Annual Sympos. on Foundations of Computer Science (1986), pp.174-187.

[GMR] S. Goldwasser, S. Micali and C. Rackoff, *Knowledge complexity of interactive proof systems*, Proc. 17th ACM Annual Sympos. on Theory of Computing (1985), pp. 291-304.

[Karp] R. Karp, *Reducibility among combinatorical problems* in *Complexity of Computer Computations* (Miller and Thatcher, eds.), Plenum Press, New York, 1972.

[Land] S. Landau, *Primes, codes and the National Security Agency*, Notices Amer. Math. Soc., **30** (1983), 7-10.

[Len] H.W. Lenstra, Jr., *Factoring integers with elliptic curves*, Ann. of Math., to appear.

[Mil] G. L. Miller, *Riemann's hypothesis and tests for primality*, J. Comput. Systems Sci. **13** (1976), 300-317.

[PCSG] Public Cryptography Study Group, *Report of the public cryptography study group*, American Council on Education, February 1981, reprinted in Notices Amer. Math. Soc., **28** (October 1981), 518-526.

[Re] H. Relyea, *Striking a balance: National security and scientific freedoms first discussions*, American Association for the Advancement of Science, Washington, D.C., May 1985.

[RSA] R. Rivest, A. Shamir and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. of the ACM (February 1978), 120-126.

[TW] M. Tompa and H. Woll, *Random self-reducibility and zero knowledge proofs of possession of information*, Proc. 28th Annual Sympos. on Foundations of Computer Science (1987), pp. 472-482.

[US] U.S. Congress, House of Representatives, 34th Report by the Committee on Government Operations, Together with Additional Views, *The Government's Classification of Private Ideas*, House Report 96-1540, December 1980.

Jeffrey C. Lagarias is joining Ronald L. Graham as an Associate Editor of Special Articles. The series of Special Articles was created to provide a place for articles on mathematical subjects of interest to the general membership of the Society. The Editorial Committee of the *Notices* is especially interested in the quality of exposition and intends to maintain the highest standards in order to assure that the Special Articles will be accessible to mathematicians in all fields. The articles must be interesting and mathematically sound. They are first refereed for accuracy and (if approved) accepted or rejected on the basis of the breadth of their appeal to the general mathematical public.

Items for this series are solicited and, if accepted, will be paid for at the rate of $250 per page up to a maximum of $750. Manuscripts to be considered for this series should be sent to Ronald L. Graham or Jeffrey C. Lagarias, Associate Editor for Special Articles, *Notices of the American Mathematical Society*, Post Office Box 6248, Providence, Rhode Island 02940.