The First Ten Years of Public-Key Cryptography

WHITFIELD DIFFIE

Invited Paper

Public-key cryptosystems separate the capacities for encryption and decryption so that 1) many people can encrypt messages in such a way that only one person can read them, or 2) one person can encrypt messages in such a way that many people can read them. This separation allows important improvements in the management of cryptographic keys and makes it possible to 'sign' a purely digital message.

Public key cryptography was discovered in the Spring of 1975 and has followed a surprising course. Although diverse systems were proposed early on, the ones that appear both practical and secure today are all very closely related and the search for new and different ones has met with little success. Despite this reliance on a limited mathematical foundation public-key cryptography is revolutionizing communication security by making possible secure communication networks with hundreds of thousands of subscribers.

Equally important is the impact of public key cryptography on the theoretical side of communication security. It has given cryptographers a systematic means of addressing a broad range of security objectives and pointed the way toward a more theoretical approach that allows the development of cryptographic protocols with proven security characteristics.

I. Initial Discoveries

Public key cryptography was born in May 1975, the child of two problems and a misunderstanding.

- First came the problem of key distribution. If two people who have never met before are to communicate privately using conventional cryptographic means, they must somehow agree in advance on a key that will be known to themselves and to no one else.
- The second problem, apparently unrelated to the first, was the problem of signatures. Could a method be devised that would provide the recipient of a purely digital electronic message with a way of demonstrating to other people that it had come from a particular person, just as a written signature on a letter allows the recipient to hold the author to its contents?

On the face of it, both problems seem to demand the impossible. In the first case, if two people could somehow communicate a secret key from one to the other without ever having met, why could they not communicate their message in secret? The second is no better. To be effective, a signature must be hard to copy. How then can a digital message, which can be copied perfectly, bear a signature?

The misunderstanding was mine and prevented me from rediscovering the conventional key distribution center. The virtue of cryptography, I reasoned, was that, unlike any other known security technology, it did not require trust in any party not directly involved in the communication, only trust in the cryptographic systems. What good would it do to develop impenetrable cryptosystems, I reasoned, if their users were forced to share their keys with a key distribution center that could be compromised by either burglary or subpoena.

The discovery consisted not of a solution, but of the recognition that the two problems, each of which seemed unsolvable by definition, could be solved at all and that the solutions to both problems came in one package.

First to succumb was the signature problem. The conventional use of cryptography to authenticate messages had been joined in the 1950s by two new applications, whose functions when combined constitute a signature.

Beginning in 1952, a group under the direction of Horst Feistel at the Air Force Cambridge Research Center began to apply cryptography to the military problem of distinguishing friendly from hostile aircraft. In traditional *Identification Friend or Foe* systems, a fire control radar determines the identity of an aircraft by challenging it, much as a sentry challenges a soldier on foot. If the airplane returns the correct identifying information, it is judged to be friendly, otherwise it is thought to be hostile or at best neutral. To allow the correct response to remain constant for any significant period of time, however, is to invite opponents to record a legitimate friendly response and play it back whenever they themselves are challenged. The approach taken by Feistel's group, and now used in the MK XII IFF system, is to vary the exchange cryptographically from encounter to encounter. The radar sends a randomly selected challenge and judges the aircraft by whether it receives a correctly encrypted response. Because the challenges are never repeated, previously recorded responses will not be judged correct by a challenging radar.

Later in the decade, this novel authentication technique was joined by another, which seems first to have been

Manuscript received January 19, 1988; revised March 25, 1988.
The author is with Bell-Northern Research, Mountain View, CA 94039, USA.
IEEE Log Number 8821645.

applied by Roger Needham of Cambridge University [112]. This time the problem was protecting computer passwords. Access control systems often suffer from the extreme sensitivity of their password tables. The tables gather all of the passwards together in one place and anyone who gets access to this information can impersonate any of the system's users. To guard against this possibility, the password table is filled not with the passwords themselves, but with the images of the passwords under a *one-way function.* A one-way function is easy to compute, but difficult to invert. For any password, the correct table entry can be calculated easily. Given an output from the one-way function, however, it is exceedingly difficult to find any input that will produce it. This reduces the value of the password table to an intruder tremendously, since its entries are not passwords and are not acceptable to the password verification routine.

Challenge and response identification and one-way functions provide protection against two quite different sorts of threats. Challenge and response identification resists the efforts of an eavesdropper who can spy on the communication channel. Since the challenge varies randomly from event to event, the spy is unable to replay it and fool the challenging radar. There is, however, no protection against an opponent who captures the radar and learns its cryptographic keys. This opponent can use what he has learned to fool any other radar that is keyed the same. In contrast, the one-way function defeats the efforts of an intruder who captures the system password table (analogous to capturing the radar) but scuccumbs to anyone who intercepts the login message because the password does not change with time.

I realized that the two goals might be achieved simultaneously if the challenger could pose questions that it was unable to answer, but whose answers it could judge for correctness. I saw the solution as a generalization of the one-way function: a *trap-door one-way function* that allowed someone in possession of secret information to go backwards and compute the function's inverse. The challenger would issue a value in the range of the one-way function and demand to know its inverse. Only the person who knew the trapdoor would be able to find the corresponding element in the domain, but the challenger, in possession of an algorithm for computing the one-way function, could readily check the answer. In the applications that later came to seem most important, the role of the challenge was played by a message and the process took on the character of a signature, a *digital signature.*

It did not take long to realize that the trap-door one-way function could also be applied to the baffling problem of key distribution. For someone in possession of the forward form of the one-way function to send a secret message to the person who knew the trapdoor, he had only to transform the message with the one-way function. Only the holder of the trap-door information would be able to invert the operation and recover the message. Because knowing the forward form of the function did not make it possible to compute the inverse, the function could be made freely available. It is this possibility that gave the field its name: *public-key cryptography.*

The concept that emerges is that of a *public-key cryptosystem:* a cryptosystem in which keys come in inverse pairs [36] and each pair of keys has two properties.

- Anything encrypted with one key can be decrypted with the other.
- Given one member of the pair, the *public key,* it is infeasible to discover the other, the *secret key.*

This separation of encryption and decryption makes it possible for the subscribers to a communication system to list their public keys in a "telephone directory" along with their names and addresses. This done, the solutions to the original problems can be achieved by simple protocols.

- One subscriber can send a private message to another simply by looking up the addressee's public key and using it to encrypt the message. Only the holder of the corresponding secret key can read such a message; even the sender, should he lose the plaintext, is incapable of extracting it from the ciphertext.
- A subscriber can sign a message by encrypting it with his own secret key. Anyone with access to the public key can verify that it must have been encrypted with the corresponding secret key, but this is of no help to him in creating (forging) a message with this property.

The first aspect of public-key cryptography greatly simplifies the management of keys, especially in large communication networks. In order for a pair of subscribers to communicate privately using conventional end-to-end cryptography, they must both have copies of the same cryptographic key and this key must be kept secret from anyone they do not wish to take into their confidence. If a network has only a few subscribers, each person simply stores one key for every other subscriber against the day he will need it, but for a large network, this is impractical.

In a network with n subscribers there are $n(n-1)/2$ pairs, each of which may require a key. This amounts to five thousand keys in a network with only a hundred subscribers, half a million in a network with one thousand, and twenty million billion in a network the size of the North American telephone system. It is unthinkable to distribute this many keys in advance and undesirable to postpone secure communication while they are carried from one party to the other by courier.

The second aspect makes its possible to conduct a much broader range of normal business practices over a telecommunication network. The availability of a signature that the receiver of a message cannot forge and the sender cannot readily disavow makes it possible to trust the network with negotiations and transactions of much higher value than would otherwise be possible.

It must be noted that both problems can be solved without public-key cryptography, but that conventional solutions come at a great price. Centralized *key distribution centers* can on request provide a subscriber with a key for communicating with any other subscriber and protocols for this purpose will be discussed later on. The function of the signature can also be approximated by a central registry that records all transactions and bears witness in cases of dispute. Both mechanisms, however, encumber the network with the intrusion of a third party into many conversations, diminishing security and degrading performance.

At the time public-key cryptography was discovered, I was working with Martin Hellman in the Electrical Engineering Department at Stanford University. It was our immediate reaction, and by no means ours alone, that the problem of producing public-key cryptosystems would be quite difficult. Instead of attacking this problem in earnest, Marty and I forged ahead in examining the consequences.

The first result of this examination to reach a broad audience was a paper entitled "Multi-User Cryptographic Techniques" [35], which we gave at the National Computer Conference in 1976. We wrote the paper in December 1975 and sent preprints around immediately. One of the preprints went to Peter Blatman, a Berkeley graduate student and friend since childhood of cryptography's historian David Kahn. The result was to bring from the woodwork Ralph Merkle, possibly the single most inventive character in the public-key saga.

Merkle's Puzzles

Ralph Merkle had registered in the Fall of 1974 for Lance Hoffman's course in computer security at U.C. Berkeley. Hoffman wanted term papers and required each student to submit a proposal early in the term. Merkle addressed the problem of public-key distribution or as he called it "Secure Communication over Insecure Channels" [70]. Hoffman could not understand Merkle's proposal. He demanded that it be rewritten, but alas found the revised version no more comprehensible than the original. After one more iteration of this process, Merkle dropped the course, but he did not cease working on the problem despite continuing failure to make his results understood.

Although Merkle's original proposal may have been hard to follow, the idea is quite simple. Merkle's approach is to communicate a cryptographic key from one person to another by hiding it in a large collection of puzzles. Following the tradition in public-key cryptography the parties to this communication will be called Alice and Bob rather than the faceless *A* and *B*, *X* and *Y*, or *I* and *J*, common in technical literature.

Alice manufactures a million or more puzzles and sends them over the exposed communication channel to Bob. Each puzzle contains a cryptographic key in a recognizable standard format. The puzzle itself is a cryptogram produced by a block cipher with a fairly small key space. As with the number of puzzles, a million is a plausible number. When Bob receives the puzzles, he picks one and solves it, by the simple expedient of trying each of the block cipher's million keys in turn until he finds one that results in plaintext of the correct form. This requires a large but hardly impossible amount of work.

In order to inform Alice which puzzle he has solved, Bob uses the key it contains to encrypt a fixed test message, which he transmits to Alice. Alice now tries her million keys on the test message until she finds the one that works. This is the key from the puzzle Bob has chosen.

The task facing an intruder is more arduous. Rather than selecting one of the puzzles to solve, he must solve on average half of them. The amount of effort he must expend is therefore approximately the square of that expended by the legitimate communicators.

The n to n^2 advantage the legitimate communicators have over the intruder is small by cryptographic standards, but sufficient to make the system plausible in some circumstances. Suppose, for example, that the plaintext of each puzzle is 96 bits, consisting of 64 bits of key together with a thirty-two bit block of zeros that enables Bob to recognize the right solution. The puzzle is constructed by encrypting this plaintext using a block cipher with 20 bits of key. Alice produces a million of these puzzles and Bob requires about half a million tests to solve one. The bandwidth and computing power required to make this feasible are large but not inaccessible. On a DS1 (1.544 Mbit) channel it would require about a minute to communicate the puzzles. If keys can be tried on the selected puzzle at about ten-thousand per second, it will take Bob another minute to solve it. Finally, it will take a similar amount of time for Alice to figure out, from the test message, which key has been chosen.

The intruder can expect to have to solve half a million puzzles at half a million tries apiece. With equivalent computational facilities, this requires twenty-five million seconds or about a year. For applications such as authentication, in which the keys are no longer of use after communication is complete, the security of this system might be sufficient.

When Merkle saw the preprint of "Multi-User Cryptographic Techniques" he immediately realized he had found people who would appreciate his work and sent us copies of the paper he had been endeavoring unsuccessfully to publish. We in turn realized that Merkle's formulation of the problem was quite different from mine and, because Merkle had isolated one of the two intertwined problems I had seen, potentially simpler.

Even before the notion of putting trap-doors into one-way functions had appeared, a central objective of my work with Marty had been to identify and study functions that were easy to compute in one direction, but difficult to invert. Three principal examples of this simplest and most basic of cryptographic phenomena occupied our thoughts.

- John Gill, a colleague in the Electrical Engineering Department at Stanford, had suggested discrete exponentiation because the inverse problem, discrete logarithm, was considered very difficult.
- I had sought suitable problems in the chapter on NP-complete functions in Aho, Hopcroft, and Ullman's book on computational complexity [3] and selected the knapsack problem as most appropriate.
- Donald Knuth of the Stanford Computer Science Department had suggested that multiplying a pair of primes was easy, but that factoring the result, even when it was known to have precisely two factors, was exceedingly hard.

All three of these one-way functions were shortly to assume great importance.

II. Exponential Key Exchange

The exponential example was tantalizing because of its combinatorial peculiarities. When I had first thought of digital signatures, I had attempted to achieve them with a scheme using tables of exponentials. This system failed, but Marty and I continued twisting exponentials around in our minds and discussions trying to make them fit. Marty eventually made the breakthrough early one morning in May 1976. I was working at the Stanford Artificial Intelligence Laboratory on the paper that we were shortly to publish under the title "New Directions in Cryptography" [36] when Marty called and explained exponential key exchange in its unnerving simplicity. Listening to him, I realized that the notion had been at the edge of my mind for some time, but had never really broken through.

Exponential key exchange takes advantage of the ease with which exponentials can be computed in a Galois (finite) field $GF(q)$ with a prime number q of elements (the numbers $\{0, 1, \cdots, q-1\}$ under arithmetic modulo q) as compared with the difficulty of computing logarithms in the same field. If

$$Y = \alpha^X \bmod q, \qquad \text{for } 1 < X < q - 1$$

where α is a fixed primitive element of $GF(q)$ (that is the powers of α produce all the nonzero elements $1, 2, \cdots, q - 1$ of $GF(q)$), then X is referred to as the logarithm of Y to the base α, over $GF(q)$:

$$X = \log_\alpha Y \text{ over } GF(q), \qquad \text{for } 1 < Y < q - 1.$$

Calculation of Y from X is easy: Using repeated squaring, it takes at most $2 \times \log_2 q$ multiplications. For example

$$\begin{aligned}\alpha^{37} &= \alpha^{32+4+1}\\ &= \left(\left(\left((\alpha^2)^2\right)^2\right)^2\right)^2 \times (\alpha^2)^2 \times \alpha.\end{aligned}$$

Computing X from Y, on the other hand, is typically far more difficult [104], [83], [29]. If q has been chosen correctly, extracting logarithms modulo q requires a precomputation proportional to

$$L(q) = e^{\sqrt{\ln q \times \ln \ln q}},$$

though after that individual logarithms can be calculated fairly quickly. The function $L(q)$ also estimates the time needed to factor a composite number of comparable size and will appear again in that context.

To initiate communication Alice chooses a random number X_A uniformly from the integers $1, 2, \cdots, q - 1$. She keeps X_A secret, but sends

$$Y_A = \alpha^{X_A} \bmod q$$

to Bob. Similarly, Bob chooses a random number X_B and sends the corresponding Y_B to Alice. Both Alice and Bob can now compute

$$K_{AB} = \alpha^{X_A X_B} \bmod q$$

and use this as their key. Alice computes K_{AB} by raising the Y_B she obtained from Bob to the power X_A

$$\begin{aligned}K_{AB} &= Y_B^{X_A} \bmod q\\ &= (\alpha^{X_B})^{X_A} \bmod q\\ &= \alpha^{X_B X_A} = \alpha^{X_A X_B} \bmod q\end{aligned}$$

and Bob obtains K_{AB} in a similar fashion

$$K_{AB} = Y_A^{X_B} \bmod q.$$

No one except Alice and Bob knows either X_A or X_B so anyone else must compute K_{AB} from Y_A and Y_B alone. The equivalence of this problem to the discrete logarithm problem is a major open question in public-key cryptography. To date no easier solution than taking the logarithm of either Y_A or Y_B has been discovered.

If q is a prime about 1000 bits in length, only about 2000 multiplications of 1000-bit numbers are required to compute Y_A from X_A, or K_{AB} from Y_A and X_B. Taking logarithms over $GF(q)$, on the other hand, currently demands more than 2^{100} (or approximately 10^{30}) operations.

The arithmetic of exponential key exchange is not restricted to prime fields; it can also be done in Galois Fields with 2^n elements, or in prime product rings [103], [68]. The '2^n' approach has been taken by several people [64], [117], [56] because arithmetic in these fields can be performed with linear shift registers and is much faster than arithmetic over large primes. It has turned out, however, that discrete logarithms can also be calculated much more quickly in '2^n' fields and so the sizes of the registers must be about 50 percent greater.

Marty and I immediately recognized that we had a far more compact solution to the key distribution problem than Merkle's puzzles and hastened to add it to both the upcoming National Computer Conference presentation and to "New Directions." The latter now contained a solution to each aspect of the public-key problem, though not the combined solution I had envisioned. It was sent off to the IEEE TRANSACTIONS ON INFORMATION THEORY prior to my departure for NCC and like all of our other papers was immediately circulated in preprint.

III. TRAP-DOOR KNAPSACKS

Later in the same year, Ralph Merkle began work on his best known contribution to public-key cryptography: building trapdoors into the knapsack one-way function to produce the trap-door knapsack public-key cryptosystem.

The knapsack problem is fancifully derived from the notion of packing gear into a knapsack. A shipping clerk faced with an odd assortment of packages and a freight container will naturally try to find a subset of the packages that fills the container exactly with no wasted space. The simplest case of this problem, and the one that has found application in cryptography is the one dimensional case: packing varying lengths of fishing rod into a tall thin tube.

Given a *cargo vector* of integers $\boldsymbol{a} = (a_1, a_2 \cdots, a_n)$ it is easy to add up the elements of any specified *subvector*. Presented with an integer S, however, it is not easy to find a subvector of $\boldsymbol{a}$ whose elements sum to S, even if such a subvector is known to exist. This *knapsack problem* is well known in combinatorics and is believed to be extremely difficult in general. It belongs to the class of NP-complete problems, problems thought not to be solvable in polynomial time on any deterministic computer.

I had previously identified the knapsack problem as a theoretically attractive basis for a one-way function. The cargo vector $\boldsymbol{a}$ can be used to encipher an n-bit message $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ by taking the dot product $S = \boldsymbol{a} \cdot \boldsymbol{x}$ as the ciphertext. Because one element of the dot product is binary, this process is easy and simply requires n additions. Inverting the function by finding a binary vector $\boldsymbol{x}$ such that $\boldsymbol{a} \cdot \boldsymbol{x} = S$ solves the knapsack problem and is thus believed to be computationally infeasible if $\boldsymbol{a}$ is randomly chosen. Despite this difficulty in general, many cases of the knapsack problem are quite easy and Merkle contrived to build a trapdoor into the knapsack one-way function by starting with a simple cargo vector and converting it into a more complex form [71].

If the cargo vector $\boldsymbol{a}$ is chosen so that each element is larger than the sum of the preceding elements, it is called *superincreasing* and its knapsack problem is particularly simple. (In the special case where the components are 1, 2, 4, 8, etc., this is the elementary operation of binary decomposition.) For example, if $\boldsymbol{a}' = (171, 197, 459, 1191, 2410)$ and $S' = 3798$ then x_5 must equal 1. If it were 0 then even if x_1, x_2, x_3, and x_4 were all equal to 1, the dot product $\boldsymbol{a} \cdot \boldsymbol{x}$ would be too small. Since $x_5 = 1$, $S' - a_5' = 3797 - 2410$

$= 1387$ must be a sum of a subset of the first four elements of a'. The fact that $1387 > a'_4 = 1191$ means that x_4 too must equal 1. Finally $S' - a'_5 - a'_4 = 196 = a'_2$ so $x_3 = 0$, $x_2 = 1$, and $x_1 = 0$.

The simple cargo vector $\mathbf{a}'$ cannot be used as a public enciphering key because anyone can easily recover a vector $\mathbf{x}$ for which $\mathbf{x} \cdot \mathbf{a}' = S'$ from $\mathbf{a}'$ and S' by the process described above. The algorithm for generating keys therefore chooses a random superincreasing cargo vector $\mathbf{a}'$ (with a hundred or more components) and keeps this vector secret. It also generates a random integer m, larger than $\Sigma \mathbf{a}'_i$, and a random integer w, relatively prime to m, whose inverse w^{-1} mod m will be used in decryption. The public cargo vector or enciphering key $\mathbf{a}$ is produced by multiplying each component of $\mathbf{a}'$ by w mod m

$$\mathbf{a} = w\mathbf{a}' \bmod m.$$

Alice publishes a transposed version of $\mathbf{a}$ as her public key, but keeps the transposition, the simple cargo vector $\mathbf{a}'$, the multiplier w and its inverse, and the modulus m secret as her private key.

When Bob wants to send the message $\mathbf{x}$ to Alice he computes and sends

$$S = \mathbf{a} \cdot \mathbf{x}.$$

Because

$$\begin{aligned} S' &= w^{-1}S \bmod m \\ &= w^{-1} \sum a_i x_i \bmod m \\ &= w^{-1} \sum (wa'_i \bmod m) x_i \bmod m \\ &= \sum (w^{-1}wa'_i \bmod m) x_i \bmod m \\ &= \sum a'_i x_i \bmod m \\ &= \mathbf{a}' \cdot \mathbf{x} \end{aligned}$$

when $m > \Sigma a'_i$, Alice can use her secret information, w^{-1} and m, to transform any message S that has been enciphered with her public key into $S' = w^{-1} \times S$ and solve the easy knapsack problem $S' = \mathbf{a}' \cdot \mathbf{x}$ to obtain $\mathbf{x}$.

For example, for the secret vector $\mathbf{a}'$, above, the values $w = 2550$ and $m = 8443$, result in the public vector $\mathbf{a} = (5457, 4213, 5316, 6013, 7439)$, which hides the structure present in $\mathbf{a}'$.

This process can be iterated to produce a sequence of cargo vectors with more and more difficult knapsack problems by using transformations (w_1, m_1), (w_2, m_2), etc. The overall transformation that results is not, in general, equivalent to any single (w, m) transformation.

The trap-door knapsack system does not lend itself readily to the production of signatures because most elements S of the ciphertext space $\{0 \le S \le \Sigma a_i\}$, do not have inverse images. This does not interfere with the use of the system for sending private messages, but requires special adaptation for signature applications [71], [98]. Merkle had great confidence in even the single iteration knapsack system and posted a note on his office offering a $100 reward to anyone who could break it.

IV. The RSA System

Unknown to us at the time we wrote "New Directions" were the three people who were to make the single most spectacular contribution to public-key cryptography: Ronald Rivest, Adi Shamir, and Leonard Adleman. Ron Rivest had been a graduate student in computer science at Stanford while I was working on proving the correctness of programs at the Stanford Artificial Intelligence Laboratory. One of my colleagues in that work was Zohar Manna, who shortly returned to Israel and supervised the doctoral research of Adi Shamir, at the Weitzman Institute. Len Adleman was a native San Françiscan with both undergraduate and graduate degrees from U.C. Berkeley. Despite this web of near connections, not one of the three had previously crossed our paths and their names were unfamiliar.

When the New Directions paper reached MIT in the fall of 1976, the three took up the challenge of producing a full-fledged public-key cryptosystem. The process lasted several months during which Rivest proposed approaches, Adleman attacked them, and Shamir recalls doing some of each.

In May 1977 they were rewarded with success. After investigating a number of possibilities, some of which were later put forward by other researchers [67], [1], they had discovered how a simple piece of classical number theory could be made to solve the problem. The resulting paper [91] also introduced Alice and Bob, the first couple of cryptography [53].

The RSA cryptosystem is a block cipher in which the plaintexts and ciphertexts are integers between 0 and $N - 1$ for some N. It resembles the exponential key exchange system described above in using exponentiation in modular arithmetic for its enciphering and deciphering operations but, unlike that system, RSA must do its arithmetic not over prime numbers, but over composite ones.

Knowledge of a plaintext M, a modulus N, and an exponent e are sufficient to allow calculation of M^e mod N. Exponentiation, however, is a one-way function with respect to the extraction of roots as well as logarithms. Depending on the characteristics of N, M, and e, it may be very difficult to invert.

The RSA system makes use of the fact that finding large (e.g., 200 digit) prime numbers is computationally easy, but that factoring the product of two such numbers appears computationally infeasible. Alice creates her secret and public keys by selecting two very large prime numbers, P and Q, at random, and multiplying them together to obtain a *bicomposite* modulus N. She makes this product public together with a suitably chosen enciphering exponent e, but keeps the factors, P and Q secret.

The enciphering process of exponentiation modulo N can be carried out by anyone who knows N, but only Alice, who knows the factors of N, can reverse the process and decipher.

Using P and Q, Alice can compute the Euler totient function $\phi(N)$, which counts the number of integers between 1 and N that are relatively prime to N and consequently invertible in arithmetic modulo N. For a bicomposite number this is

$$\phi(N) = (P - 1)(Q - 1).$$

The quantity $\phi(N)$ plays a critical role in Euler's theorem, which says that for any number x that is invertible modulo N (and for large N that is almost all of them)

$$x^{\phi(N)} \equiv 1 \pmod{N}$$

or slightly more generally

$$x^{k\phi(N)+1} \equiv x \pmod N.$$

Using $\phi(N)$ Alice can calculate [60] a number d such that

$$e \times d \equiv 1 \pmod{\phi(N)}$$

which is equivalent to saying that

$$e \times d = k \times \phi(N) + 1.$$

When the cryptogram $M^e \bmod N$ is raised to the power d the result is

$$(M^e)^d = M^{ed} = M^{k\phi(N)+1} \equiv M \pmod N$$

the original plaintext M.

As a very small example, suppose $P = 17$ and $Q = 31$ are chosen so that $N = PQ = 527$ and $\phi(N) = (P-1)(Q-1) = 480$. If $e = 7$ is chosen then $d = 343$. ($7 \times 343 = 2401 = 5 \times 480 + 1$). And if $M = 2$ then

$$C = M^e \bmod N = 2^7 \bmod 527 = 128.$$

Note again that only the public information (e, N) is required for enciphering M. To decipher, the private key d is needed to compute

$$\begin{aligned} M &= C^d \bmod N \\ &= 128^{343} \bmod 527 \\ &= 128^{256} \times 128^{64} \times 128^{16} \times 128^4 \times 128^2 \times 128^1 \bmod 527 \\ &= 35 \times 256 \times 35 \times 101 \times 47 \times 128 \bmod 527 \\ &= 2 \bmod 527. \end{aligned}$$

Just as the strength of the exponential key exchange system is not known to be equivalent to the difficulty of extracting discrete logarithms, the strength of RSA has not been proven equivalent to factoring. There might be some method of taking the eth root of M^e without calculating d and thus without providing information sufficient to factor. While at MIT in 1978, M. O. Rabin [86] produced a variant of RSA, subsequently improved by Hugh Williams of the University of Manitoba [113], that is equivalent to factoring. Rivest and I have independently observed [38], [92], however, that the precise equivalence Rabin has shown is a two-edged sword.

V. The McEliece Coding Scheme

Within a short time yet another public-key system was to appear, this due to Robert J. McEliece of the Jet Propulsion Laboratory at Cal Tech [69]. McEliece's system makes use of the existence of a class of error correcting codes, the Goppa codes, for which a fast decoding algorithm is known. His idea was to construct a Goppa code and disguise it as a general linear code, whose decoding problem is NP-complete. There is a strong parallel here with the trapdoor knapsack system in which a superincreasing cargo vector, whose knapsack problem is simple to solve, is disguised as a general cargo vector whose knapsack problem is NP-complete.

In a knapsack system, the secret key consists of a superincreasing cargo vector $\boldsymbol{v}$, together with the multiplier w and the modulus m that disguise it; in McEliece's system, the secret key consists of the generator matrix G for a Goppa code together with a nonsingular matrix S and a permutation matrix P that disguise it. The public key appears as the encoding matrix $G' = SGP$ of a general linear code.

- To encode a data block u into a message x, Alice multiplies it by Bob's public encoding matrix G' and adds a locally generated noise block z.
- To decode, Bob multiplies the received message x by P^{-1}, decodes xp^{-1} to get a word in the Goppa code and multiplies this by S^{-1} to recover Alice's data block.

McEliece's system has never achieved wide acceptance and has probably never even been considered for implementation in any real application. This may be because the public keys are quite large, requiring on the order of a million bits; it may be because the system entails substantial expansion of the data; or it may be because McEliece's system bears a frightening structural similarity to the knapsack systems whose fate we shall discover shortly.

VI. The Fall of the Knapsacks

Nineteen eighty-two was the most exciting time for public-key cryptography since its spectacular first three years. In March, Adi Shamir sent out a research announcement: He had broken the single iteration Merkle-Hellman knapsack system [101], [102]. By applying new results of Lenstra at the Mathematische Centrum in Amsterdam, Shamir had learned how to take a public cargo vector and discover a w' and m' that would convert it back into a superincreasing "secret" cargo vector—not necessarily the same one the originator had used, but one that would suffice for decrypting messages encrypted with the public cargo vector.

Shamir's original attack was narrow. It seemed that perhaps its only consequence would be to strengthen the knapsack system by adding conditions to the construction rules for avoiding the new attack. The first response of Gustavus J. Simmons, whose work will dominate a later section, was that he could avoid Shamir's attack without even changing the cargo vector merely by a more careful choice of w and m [16]. He quickly learned, however, that Shamir's approach could be extended to break a far larger class of knapsack systems [16].

Crypto '82 revealed that several other people had continued down the trail Shamir had blazed. Shamir himself had reached the same conclusions. Andy Odlyzko and Jeff Lagarias at Bell Labs were on the same track and Len Adleman had not only devised an attack but programmed it on an Apple II. The substance of the attacks will not be treated here since it is central to another paper in this special section (E. F. Brickell and A. M. Odlyzko "Cryptanalysis: A Survey of Recent Results"). The events they engendered, however, will.

I had the pleasure of chairing the cryptanalysis session at Crypto '82 in which the various results were presented. Ironically, at the time I accepted the invitation to organize such a session, Shamir's announcement stood alone and knapsack systems were only one of the topics to be discussed. My original program ran into very bad luck, however. Of the papers initially scheduled only Donald Davies's talk on: "The Bombe at Bletchley Park," was actually presented. Nonetheless, the lost papers were more than replaced by presentations on various approaches to the knapsack problem.

Last on the program were Len Adleman and his computer, which had accepted a challenge on the first night of the conference. The hour passed; various techniques for attacking knapsack systems with different characteristics were heard; and the Apple II sat on the table waiting to reveal the results of its labors. At last Adleman rose to speak mumbling something self-deprecatingly about "the theory first, the public humiliation later" and beginning to explain his work. All the while the figure of Carl Nicolai moved silently in the background setting up the computer and copying a sequence of numbers from its screen onto a transparency. At last another transparancy was drawn from a sealed envelope and the results placed side by side on the projector. They were identical. The public humiliation was not Adleman's, it was knapsack's.

Ralph Merkle was not present, but Marty Hellman, who was, gamely arose to make a concession speech on their behalf. Merkle, always one to put his money where his mouth was, had long since paid Shamir the $100 in prize money that he had placed on the table nearly six years before.

The press wrote that knapsacks were dead. I was skeptical but ventured that the results were sufficiently threatening that I felt "nobody should entrust anything of great value to a knapsack system unless he had a much deeper theory of their functioning than was currently available." Nor was Merkle's enthusiasm dampened. He promptly raised his bet and offered $1000 to anyone who could break a multiple iteration knapsack [72].

It took two years, but in the end, Merkle had to pay [42]. The money was finally claimed by Ernie Brickell in the summer of 1984 when he announced the destruction of a knapsack system of forty iterations and a hundred weights in the cargo vector in about an hour of Cray-1 time [17]. That Fall I was forced to admit: "knapsacks are flat on their back."

Closely related techniques have also been applied to make a dramatic reduction in the time needed to extract discrete logarithms in fields of type $GF(2^n)$. This approach was pioneered by Blake, Fuji-Hara, Vanstone, and Mullin in Canada [10] and refined by Coppersmith in the U.S. [28]. A comprehensive survey of this field was given by Andy Odlyzko at Eurocrypt '84 [79].

VII. Early Responses to Public Key

A copy of the MIT report [90] on the RSA cryptosystem was sent to Martin Gardner, Mathematical Games editor of *Scientific American*, shortly after it was printed. Gardner promptly published a column [48] based on his reading of both the MIT report and "New Directions." Bearing the title: "A New Kind of Cryptosystem That Would Take Millions of Years to Break," it began a confusion that persists to this day between the two directions explored by the "New Directions" paper: public-key cryptography and the problem of proving the security of cryptographic systems. More significant, however, was the prestige that public-key cryptography got from being announced in the scientific world's most prominant lay journal more than six months before its appearance in the *Communications of the ACM*.

The excitement public-key cryptosystems provoked in the popular and scientific press was not matched by corresponding acceptance in the cryptographic establishment, however. In the same year that public-key cryptography was discovered, the National Bureau of Standards, with the support of the National Security Agency, proposed a conventional cryptographic system, designed by IBM, as a federal *Data Encryption Standard* [44]. Hellman and I criticized the proposal on the grounds that its key was too small [37], but manufacturers were gearing up to support the proposed standard and our criticism was seen by many as an attempt to disrupt the standards-making process to the advantage of our own work. Public key in its turn was attacked, in sales literature [74] and technical papers [76], [59] alike, more as though it were a competing product than a recent research discovery. This, however, did not deter NSA from claiming its share of the credit. Its director, in the words of the *Encyclopaedia Britannica* [110], "pointed out that two-key cryptography had been discovered at the agency a decade earlier," though no evidence for this claim was ever offered publicly.

Far from hurting public key, the attacks and counterclaims added to a ground swell of publicity that spread its reputation far faster than publication in scientific journals alone ever could. The criticism nonetheless bears careful examination, because the field has been affected as much by discoveries about how public key cryptosystems should be used as by discoveries about how they can be built.

In viewing public-key cryptography as a new form of cryptosystem rather than a new form of key management, I set the stage for criticism on grounds of both security and performance. Opponents were quick to point out that the RSA system ran about one thousandth as fast as DES and required keys about ten times as large. Although it had been obvious from the beginning that the use of public-key systems could be limited to exchanging keys for conventional cryptography, it was not immediately clear that this was necessary. In this context, the proposal to build *hybrid* systems [62] was hailed as a discovery in its own right.

At present, the convenient features of public-key cryptosystems are bought at the expense of speed. The fastest RSA implementations run at only a few thousand bits per second, while the fastest DES implementations run at many million. It is generally desirable, therefore, to make use of a hybrid in which the public-key systems are used only during key management processes to establish shared keys for employment with conventional systems.

No known theorem, however, says that a public-key cryptosystem must be larger and slower than a conventional one. The demonstrable restrictions mandate a larger minimum block size (though perhaps no larger than that of DES) and preclude use in stream modes whose chunks are smaller than this minimum. For a long time I felt that "high-efficiency" public-key systems would be discovered and would supplant both current public key and conventional systems in most applications. Using public-key systems throughout, I argued, would yield a more uniform architecture with fewer components and would give the best possible damage limitation in the event of a key distribution center compromise [38]. Most important, I thought, if only one system were in use, only one certification study would be required. As certification is the most fundamental and most difficult problem in cryptography, this seemed to be where the real savings lay.

In time I saw the folly of this view. Theorems or not, it seemed silly to expect that adding a major new criterion to the requirements for a cryptographic system could fail to slow it down. The designer would always have more latitude with systems that did not have to satisfy the public key property and some of these would doubtless be faster. Even more compelling was the realization that modes of operation incompatible with the public-key property are essential in many communication channels.

To date, the "high-efficiency public-key systems" that I had hoped for have not appeared and the restriction of public-key cryptography to key management and signature applications is almost universally accepted. More fundamental criticism focuses on whether public-key actually makes any contribution to security, but, before examining this criticism, we must undertake a more careful study of key distribution mechanisms.

Key Management

The solution to the problem of key management using conventional cryptography is for the network to provide a *key distribution center (KDC):* a trusted network resource that shares a key with each subscriber and uses these in a bootstrap process to provide additional keys to the subscribers as needed. When one subscriber wants to communicate securely with another, he first contacts the KDC to obtain a *session key* for use in that particular conversation.

Key distribution protocols vary widely depending on the cost of messages, the availability of multiple simultaneous connections, whether the subscribers have synchronized clocks, and whether the KDC has authority not only to facilitate, but to allow or prohibit, communications. The following example is typical and makes use of an important property of cryptographic authentication. Because a message altered by anyone who does not have the correct key will fail when tested for authenticity, there is no loss of security in receiving a message from the hands of a potential opponent. In so doing, it introduces, in a conventional context, the concept of a *certificate*—a cryptographically authenticated message containing a cryptographic key—a concept that plays a vital role in modern key management.

1) When Alice wants to call Bob, she first calls the KDC and requests a key for communicating with Bob.
2) The KDC responds by sending Alice a pair of certificates. Each contains a copy of the required session key, one encrypted so that only Alice can read it and one so that only Bob can read it.
3) When Alice calls Bob, she presents the proper certificate as her introduction. Each of them decrypts the appropriate certificate under the key that he shares with the KDC and thereby gets access to the session key.
4) Alice and Bob can now communicate securely using the session key.

Alice and Bob need not go through all of this procedure on every call; they can instead save the certificates for later use. Such *cacheing* of keys allows subscribers to avoid calling the KDC every time they pick up the phone, but the number of KDC calls is still proportional to the number of distinct pairs of subscribers who want to communicate securely. A far more serious disadvantage of the arrangement described above is that the subscribers must share the secrecy of their keying information with the KDC and if it is penetrated, they too will be compromised.

A big improvement in both economy and security can be made by the use of public-key cryptography. A certificate functions as a letter of introduction. In the protocol above, Alice has obtained a letter that introduces her to Bob and Bob alone. In a network using public-key encryption, she can instead obtain a single certificate that introduces her to any network subscriber [62].

What accounts for the difference? In a conventional network, every subscriber shares a secret key with the KDC and can only authenticate messages explicitly meant for him. If one subscriber has the key needed to authenticate a message meant for another subscriber, he will also be able to create such a message and authentication fails. In a public-key network, each subscriber has the public key of the KDC and thus the capacity to authenticate any message from the KDC, but no power to forge one.

Alice and Bob, each having obtained a certificate from the KDC in advance of making any secure calls, communicate with each other as follows:

1) Alice sends her certificate to Bob.
2) Bob sends his certificate to Alice.
3) Alice and Bob each check the KDC's signature on the certificates they have received.
4) Alice and Bob can now communicate using the keys contained in the certificates.

When making a call, there is no need to call the KDC and little to be gained by cacheing the certificates. The added security arises from the fact that the KDC is not privy to any information that would enable it to spy on the subscribers. The keys that the KDC dispenses are public keys and messages encrypted with these can only be decrypted by using the corresponding secret keys, to which the KDC has no access.

The most carefully articulated attack came from Roger Needham and Michael Schroeder [76], who compared conventional key distribution protocols with similar public-key ones. They counted the numbers of messages required and concluded that conventional cryptography was more efficient than public-key cryptography. Unfortunately, in this analysis, they had ignored the fact that security was better under the public-key protocol they presented than the conventional one.

In order to compromise a network that employs conventional cryptography, it sufficies to corrupt the KDC. This gives the intruders access to information sufficient for recovering the session keys used to encrypt past, present, and perhaps future messages. These keys, together with information obtained from passive wiretaps, allow the penetrators of the KDC access to the contents of any message sent on the system.

A public-key network presents the intruder with a much more difficult problem. Even if the KDC has been corrupted and its secret key is known to opponents, this information is insufficient to read the traffic recorded by a passive wiretap. The KDC's secret key is useful only for signing certificates containing subscribers' public keys; it does not enable the intruders to decrypt any subscriber traffic. To be able to gain access to this traffic, the intruders must use their ability to forge certificates as a way of tricking subscribers into encrypting messages with phony public keys.

In order to spy on a call from Alice to Bob, opponents who have discovered the secret key of the KDC must intercept the message in which Alice sends Bob the certificate for her public key and substitute one for a public key they have manufactured themselves and whose corresponding secret key is therefore known to them. This will enable them to decrypt any message that Alice sends to Bob. If such a misencrypted message actually reaches Bob, however, he will be unable to decrypt it and may alert Alice to the error. The opponents must therefore intercept Alice's messages, decrypt them, and reencrypt them in Bob's public key in order to maintain the deception. If the opponents want to understand Bob's replies to Alice, they must go through the same procedure with Bob, supplying him with a phony public key for Alice and translating all the messages he sends her.

The procedure above is cumbersome at best. Active wiretaps are in principle detectable, and the number the intruders must place in the net in order to maintain their control, grows rapidly with the number of subscribers being spied on. Over large portions of many networks—radio broadcast networks, for example—the message deletions essential to this scheme are extremely difficult. This forces the opponents to place their taps very close to the targets and recreates the circumstances of conventional wiretapping, thereby denying the opponents precisely those advantages of communications intelligence that make it so attractive.

It is worth observing that the use of a hybrid scheme diminishes the gain in security a little because the intruder does not need to control the channel after the session key has been selected. This threat, however, can be countered, without losing the advantages of a session key, by periodically (and unpredictably) using the public keys to exchange new session keys [40].

Public-key techniques also make it possible to conquer another troubling problem of conventional cryptographic security, the fact that compromised keys can be used to read traffic taken at an earlier date. At the trial of Jerry Whitworth, a spy who passed U.S. Navy keying information to the Russians, the judge asked the prosecution's expert witness [27]: "Why is it necessary to destroy yesterday's . . . [key] . . . list if it's never going to be used again?" The witness responded in shock: "A used key, Your Honor, is the most critical key there is. If anyone can gain access to that, they can read your communications."

The solution to this problem is to be found in a judicious combination of exponential key exchange and digital signatures, inherent in the operation of a secure telephone currently under development at Bell-Northern Research [41], [81] and intended for use on the Integrated Services Digital Network.

Each ISDN secure phone has an operating secret-key/public-key pair that has been negotiated with the network's key management facility. The public-key portion is embodied in a certificate signed by the key management facility along with such identifying information as its phone number and location. In the call setup process that follows, the phone uses this certificate to convey its public key to other phones.

1) The telephones perform an exponential key exchange to generate session keys unique to the current phone call. These keys are then used to encrypt all subsequent transmissions in a conventional cryptosystem.
2) Having established an encrypted (though not yet authenticated) channel, the phones begin exchanging credentials. Each sends the other its public-key certificate.
3) Each phone checks the signature on the certificate it has received and extracts from it the other phone's public key.
4) The phones now challenge each other to sign test messages and check the signatures on the responses using the public keys from the certificates.

Once the call setup is complete, each phone displays for its user the identity of the phone with which it is in communication.

The use of the exponential key exchange creates unique session keys that exist only inside the phones and only for the duration of the call. This provides a security guarantee whose absence in conventional cryptography is at the heart of many spy cases: once a call between uncompromised ISDN secure phones is completed and the session keys are destroyed, no compromise of the long term keys that still reside in the phones will enable anyone to decrypt the recording of the call. Using conventional key management techniques, session keys are always derivable from a combination of long-term keying material and intercepted traffic. If long-term conventional keys are ever compromised, all communications, even those of earlier date, encrypted in derived keys, are compromised as well.

In the late 1970s, a code clerk named Christopher Boyce, who worked for a CIA-sponsored division of TRW, copied keying material that was supposed to have been destroyed and sold it to the Russians [66]. More recently, Jerry Whitworth did much the same thing in the communication center of the Alameda Naval Air Station [8]. The use of exponential key exchange would have rendered such previously used keys virtually worthless.

Another valuable ingredient of modern public-key technology is the *message digest*. Implementing a digital signature by encrypting the entire document to be signed with a secret key has two disadvantages. Because public key systems are slow, both the signature process (encrypting the message with a secret key), and the verification process (decrypting the message with a public key) are slow. There is also another difficulty. If the signature process encrypts the entire message, the recipient must retain the ciphertext for however long the signed message is needed. In order to make any use of it during this period, he must either save a plaintext copy as well or repeatedly decrypt the ciphertext.

The solution to this problem seems first to have been proposed by Donald Davies and Wyn Price of the National Physical Laboratory in Teddington, England. They proposed constructing a cryptographically compressed form or digest of the message [33] and signing by encrypting this with the secret key. In addition to its economies, this has the advantage of allowing the signature to be passed around independently of the message. This is often valuable in protocols in which a portion of the message that is required in the authentication process is not actually transmitted because it is already known to both parties.

Most criticism of public-key cryptography came about because public-key management has not always been seen from the clear, certificate oriented, view described above. When we first wrote about public key, we spoke either of users looking in a public directory to find each other's keys or simply of exchanging them in the course of communication. The essential fact that each user had to authenticate any public key he received was glossed over. Those with an investment in traditional cryptography were not slow to point out this oversight. Public-key cryptography was stigmatized as being weak on authentication and, although the problems the critics saw have long been solved, the criticism is heard to this day.

VIII. Application and Implementation

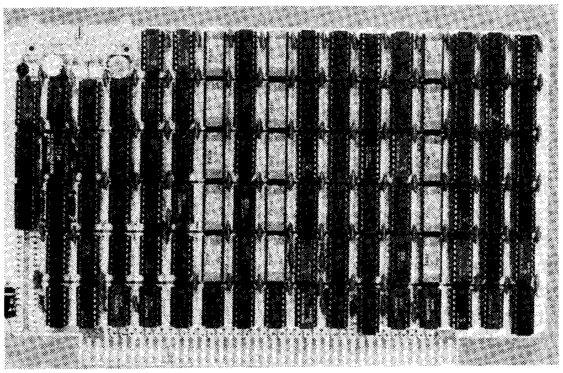
While arguments about the true worth of public-key cryptography raged in the late 1970s, it came to the attention of one person who had no doubt: Gustavus J. Simmons, head of the mathematics department of Sandia National Laboratories. Simmons was responsible for the mathematical aspects of nuclear command and control and digital signatures were just what he needed. The applications were limitless: A nuclear weapon could demand a digitally signed order before it would arm itself; a badge admitting someone to a sensitive area could bear a digitally signed description of the person; a sensor monitoring compliance with a nuclear test ban treaty could place a digital signature on the information it reported. Sandia began immediately both to develop the technology of public-key devices [108], [107], [89] and to study the strength of the proposed systems [105], [16], [34].

The application about which Simmons spoke most frequently, test-ban monitoring by remote seismic observatories [106], is the subject of another paper in this special section (G. J. Simmons, "How to Insure that Data Acquired to Verify Treaty Compliance are Trustworthy"). If the United States and the Soviet Union could put seismometers on each other's territories and use these seismometers to monitor each other's nuclear tests, the rather generous hundred and fifty kiloton upper limit imposed on underground nuclear testing by the Limited Nuclear Test Ban Treaty of 1963 could be tightened considerably—perhaps to ten kilotons or even one kiloton. The problem is this: A *monitoring* nation must assure itself that the *host* nation is not concealing tests by tampering with the data from the monitor's observatories. Conventional cryptographic authentication techniques can solve this problem, but in the process create another. A host nation wants to assure itself that the monitoring nation can monitor only total yield and does not employ an instrument package capable of detecting staging or other asects of the weapon not covered by the treaty. If the data from the remote seismic observatory are encrypted, the host country cannot tell what they contain.

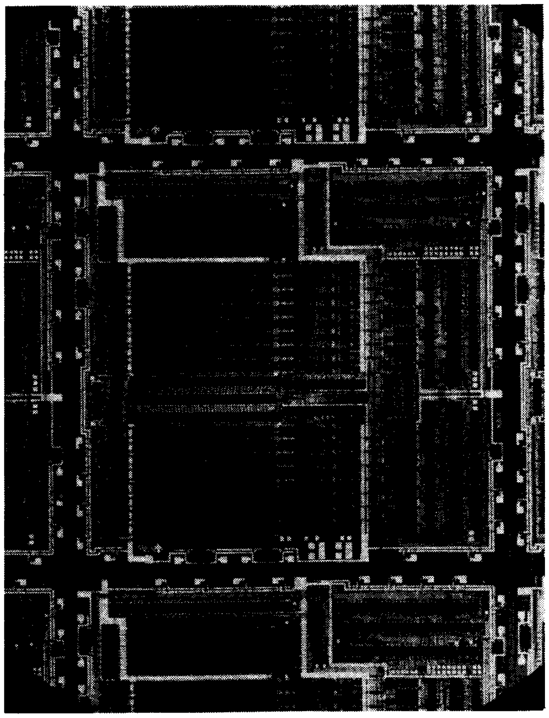
Digital signatures provided a perfect solution. A digitally signed message from a remote seismic observatory cannot be altered by the host, but can be read. The host country can assure itself that the observatory is not exceeding its authority by comparing the data transmitted with data from a nearby observatory conforming to its own interpretation of the treaty language.

The RSA system was the one best suited to signature applications, so Sandia began building hardware to carry out the RSA calculations. In 1979 it announced a board implementation intended for the seismic monitoring application [106]. This was later followed by work on both low- and high-speed chips [89], [94].

Sandia was not the only hardware builder. Ron Rivest and colleagues at MIT, ostensibly theoretical computer scientists, learned to design hardware and produced a board at approximately the same time as Sandia. The MIT board would carry out an RSA encryption with a one hundred digit modulus in about a twentieth of a second. It was adequate "proof of concept" but too expensive for the commercial applications Rivest had in mind.



Sandia 256-bit RSA board.

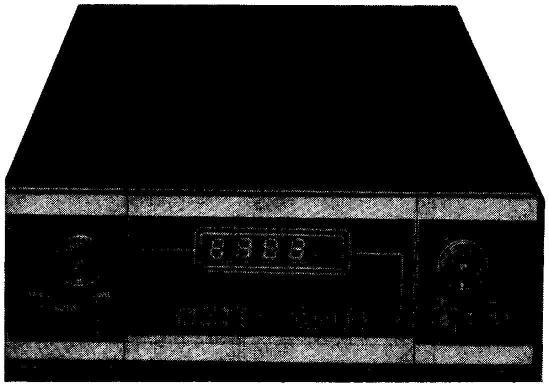


Wafer photo: Sandia low speed chip.

No sooner was the board done than Rivest started studying the recently popularized methods for designing large-scale integrated circuits. The result was an experimental nMOS chip that operated on approximately 500 bit numbers and should have been capable of about three encryptions per second [93]. This chip was originally intended as a prototype for commercial applications. As it happend, the chip was never gotten to work correctly, and the appearance of a commercially available RSA chip was to await the brilliant work of Cylink corporation in the mid-1980s [31].

As the present decade dawned, public-key technology began the transition from esoteric research to product development. Part of AT&T's response to a Carter Administration initiative to improve the overall security of American telecommunications, was to develop a specialized cryptographic device for protecting the Common Channel Interoffice Signaling (CCIS) on telephone trunks. The devices were link encryptors that used exponential key exchange to distribute DES keys [75], [16].

Although AT&T's system was widely used within its own huge network, it was never made available as a commercial product. At about the same time, however, Racal-Milgo began producing the Datacryptor II, a link encryption device that offered an RSA key exchange mode [87]. One device used exponential key exchange, the other RSA, but overall function was quite similar. When the public-key option of the Datacryptor is initialized, it manufactures a new RSA key pair and communicates the public portion to the Datacryptor at the other end of the line. The device that receives this public key manufactures a DES key and sends it to the first Datacryptor encrypted with RSA. Unfortunately, the opportunity for sophisticated digital signature based authentication that RSA makes possible was missed.



Racal-Milgo Datacryptor II.

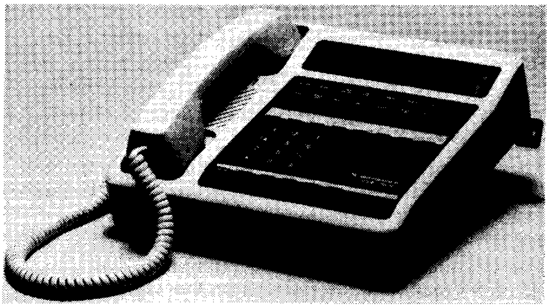
Future Secure Voice System

As the early 1980s became the mid-1980s, public-key cryptography finally achieved official, if nominally secret, acceptance. In 1983, NSA began feasibility studies for a new secure phone system. There was fewer than ten-thousand of their then latest system the Secure Telephone Unit II or STU-II and already the key distribution center for the principal network was overloaded, with users often complaining of busy signals. At $12 000 or more apiece, ten-thousand STU-IIs may have been all the government could afford, but it was hardly all the secure phones that were needed. In its desire to protect far more than just explicitly classified communications, NSA was dreaming of a million phones, each able to talk to any of the others. They could not have them all calling the key distribution center every day.

The system to be replaced employed electronic key distribution that allowed the STU-II to bootstrap itself into direct end-to-end encryption with a different key on every call. When a STU-II made a secure call to a terminal with which it did not share a key, it acquired one by calling a key distribution center using a protocol similar to one described earlier.

Although the STU-II seemed wonderful when first fielded in the late seventies, it had some major shortcomings. Some cacheing of keys was permitted, but calls to the KDC entailed significant overhead. Worse, each network had to be at a single clearance level, because there was no way for a STU-II to inform the user of the clearance level of the phone with which it was talking. These factors, as much as the high price and large size, conspired against the feasibility of building a really large STU-II network.

The STU-III is the size of a large conventional telephone and, at about $3000 apiece, substantially cheaper than its predecessor. It is equipped with a two-line display that, like the display of the ISDN secure phone, provides information to each party about the location, affiliation, and clearance of the other. This allows one phone to be used for the protection of information at various security levels. The phones are also sufficiently tamper resistant that unlike earlier equipment, the unkeyed instrument is unclassified. These elements will permit the new system to be made much more widely available with projections of the number in use by the early 1990s running from half a million to three million [18], [43].



Motorola STU-III secure telephone.

To make a secure call with a STU-III, the caller first places an ordinary call to another STU-III, then inserts a key-shaped device containing a cryptographic variable and pushes a "go secure" button. After an approximately fifteen second wait for cryptographic setup, each phone shows information about the identity and clearance of the other party on its display and the call can proceed.

In an unprecedented move, Walter Deeley, NSA's deputy director for communications security, announced the STU-III or Future Secure Voice System in an exclusive interview given to *The New York Times* [18]. The objective of the new system was primarily to provide secure voice and low-speed data communications for the U.S. Defense Department and its contractors. The interview did not say much about how it was going to work, but gradually the word began to leak out. The new system was using public key.

The new approach to key management was reported early on [88] and one article [6] spoke of phones being "reprogrammed once a year by secure telephone link," a turn of phrase strongly suggestive of a certificate passing protocol, similar to that described earlier, that minimizes the need for phones to talk to the key management center. Recent reports have been more forthcoming, speaking of a key management system called *FIREFLY* that, [95] "evolved from public key technology and is used to establish pair-wise traffic encryption keys." Both this description and testimony submitted to Congress by Lee Neuwirth of Cylink [78] suggest a combination of key exchange and certificates similar to that used in the ISDN secure phone and it is plausible that FIREFLY too is based on exponentiation.

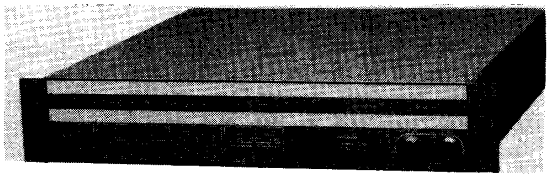
Three companies: AT&T, Motorola, and RCA are manufacturing the instruments in interoperable versions and GTE is building the key management system. So far, contracts have been issued for an initial 75 000 phones and deliveries began in November 1987.

Current Commercial Products

Several companies dedicated to developing public-key technology have been formed in the 1980s. All have been established by academic cryptographers endeavoring to exploit their discoveries commercially.

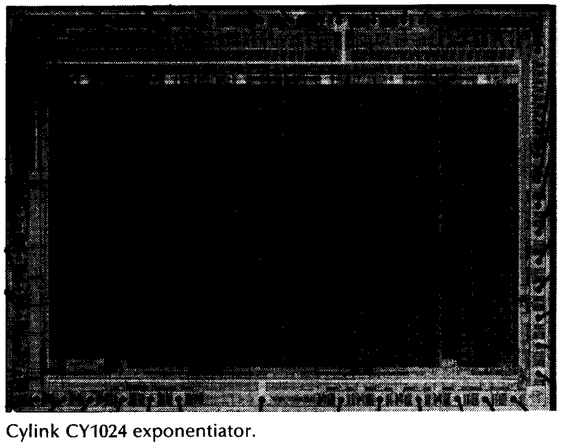
The first was RSA Data Security, founded by Rivest, Shamir, and Adleman, the inventors of the RSA cryptosystem, to exploit their patent on RSA and develop products based on the new technology. RSA produces a stand-alone software package called *Mailsafe* for encrypting and signing electronic mail. It also makes the primitives of this system available as a set of embeddable routines called *Bsafe* that has been licensed to major software manufacturers [9].

Cylink Corporation of Sunnyvale, California, has chalked up the most impressive engineering record in the public-key field. Its first product was the CIDEC HS [32], [63], a high-speed (1.544-Mbit) data encryptor for protecting DS1 telephone trunks. Like AT&T's CCIS encryptor, it uses exponential key exchange to establish DES session keys [77].

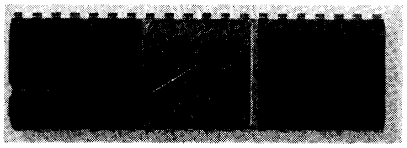


Cylink CIDEC-HS.

Cylink is also first to produce a commercially available RSA chip [7], [31]. The CY1024 is, despite its name, a 1028 bit exponential engine that can be cascaded to perform the calculations for RSA encryptions on moduli more than sixteen thousand bits long. A single CY1024 does a thousand bit encryption in under half a second—both modulus size and speed currently being sufficient for most applications.



Cylink CY1024 exponentiator.



Chip photo: Cylink CY1024.

The cryptography group at Waterloo University in Ontario have brought the fruits of their labors to market through a company called Cryptech. Their initial inroads into the problem of extracting logarithms over finite fields with 2^n elements [10] made it necessary to employ larger fields. This in turn inspired them to develop high-speed exponentiation algorithms. The result is a system providing both exponential key exchange and half megabit data encryption with the same system [56].

IX. Multiplying, Factoring, and Finding Primes

The successes of the RSA system and of exponential key exchange over prime fields have led to significant development in three areas: multiplying, factoring, and finding prime numbers.

Factoring the modulus has remained the front runner among attacks on the RSA system. As factoring has improved, the modulus size required for security has more than doubled, requiring the system's users to hunt for larger and larger prime numbers in order to operate the system securely. As the numbers grow larger, faster and faster methods for doing modular arithmetic are required. The result has been not only the development of a technical base for public-key cryptography, but an inspiration and source of support for number theory [61], [65].

Factoring

In addressing the question of how large the primes in the RSA system should be, Rivest, Shamir, and Adleman's original memo spoke of a number d such that: "determining the prime factorization of a number n which is the product of just two prime numbers of length d (in digits) is 'computationally impossible'." When MIT/LCS/TM-82 first appeared, it contained the statement "Choosing $d = 40$ seems to be satisfactory at present." In a second printing the recommended value of d was changed to 50 and in a third took a sharp leap to 100. This escalation is symbolic of the direction of factoring in the late 1970s and early 1980s.

In 1975, the factoring of a 39 digit number [73] constituted a landmark. The advent of the RSA system, however, was to usher in a decade of rapid progress in this field. By the end of that decade, numbers twice as long could be factored, if not with ease, at least with hours of Cray-1 time [34]. These factorizations confirmed, by actual computer implementation, the number theorists' predictions about factoring speed.

Several factoring techniques of comparable performance have become available in recent years [85]. All factor, in time, proportional to

$$L(n) = e^{\sqrt{\ln n \times \ln \ln n}}$$

a figure that has already been seen in connection with discrete logarithms. The one that has been most widely applied is called quadratic sieve factoring [34] and lends itself well to machine implementation. One of factoring's gurus, Marvin Wunderlich, gave a paper in 1983 [116] that examined the way in which quadratic sieve factoring could exploit parallel processing to factor a hundred digit number in two months. In the same lecture, Wunderlich also explained the importance of uniformity in factoring methods applied in cryptanalysis. To be used in attacking RSA, a factoring method must be uniform, at least over the class of bicomposite numbers. If it is only applicable to numbers of some particular form, as many methods used by number theorists have been, the cryptographers will simply alter their key production to avoid numbers of that form.

More recently, Carl Pomerance [85] has undertaken the design of a modular machine employing custom chips and specialized to factoring. The size of the numbers you can factor is dependent on how much of such a machine you can afford. He has begun building a $25 000 implementation that he expects to factor 100 digit numbers in two weeks [96]. Ten million dollars worth of similar hardware would be able to factor hundred and fifty digit numbers in a year, but Pomerance's analysis does not stop there. Fixing one year as a nominal upper limit on our patience with factoring any one number, he is prepared to give a dollar estimate for factoring a number of any size. For a two hundred digit number, often considered unapproachable and a benchmark in judging RSA systems, the figure is one hundred billion dollars. This is a high price to be sure, but not beyond human grasp.

Prime Finding

Prime finding has followed a somewhat different course from factoring. This is in part because there are probabilistic techniques that identify primes with sufficient certainty to satisfy all but perhaps the pickiest of RSA users and in part because primality is not in itself a sufficient condition for numbers to be acceptable as RSA factors.

Fermat's Little Theorem guarantees that if n is prime then for all $0 < b < n$

$$b^{n-1} \equiv 1 \pmod{n}$$

and any number that exhibits this property for some b is said to pass the pseudoprime test to base b. Composite numbers that pass pseudoprime tests to all bases exist, but they are rare and a number that passes several pseudoprime tests is probably a prime.

The test can be refined by making use of the fact that if n is an odd prime only the numbers 1 and -1 are square roots of 1, whereas if n is the product of distinct odd primes, the number of square roots of unity grows exponentially in the number of factors. If the number n passes the pseudoprime test to base b, it can be further examined to see if

$$b^{\frac{n-1}{2}} \equiv \pm 1 \pmod{n}.$$

Tests of this kind are called strong pseudoprime tests to base b and very few composite numbers that pass strong pseudoprime tests to more than a few bases are known.

Although there has been extensive work in the past decade on giving genuine proofs of primality [84], [2], [51], the strong pseudoprime tests take care of the primality aspect of choosing the factors of RSA moduli. Another aspect arises from the fact that not all prime numbers are felt to be equally good. In many RSA implementations, the factors of the modulus are not random large primes p, but large primes chosen for particular properties of the factors of $p - 1$ [91], [52].

High-Speed Arithmetic

Because of the progress in factoring during the decade of public-key's existence, the size of the numbers used in RSA has grown steadily. In the early years, talk of hundred digit moduli was common. One hundred digit numbers, 332 bits, did not seem likely to be factored in the immediate future and, with the available computing techniques, systems with bigger moduli ran very slowly. Today, hundred digit numbers seem only just out of reach and there is little discussion of moduli smaller than 512 bits. Two hundred digits, 664 bits, is frequently mentioned, and Cylink has not only chosen to make its chip a comfortable 1028 bits, but also to allow up to sixteen chips to be used in cascade. If this expansion has been pushed by advances in factoring, it has been made possible by advances in arithmetic.

Most of the computation done both in encryption and decryption and in the ancillary activity of manufacturing keys is exponentiation and each exponentiation, in turn, is made up of multiplications. Because, as discussed in the section of exponential key exchange, numbers can be raised to powers in a small number of operations by repeated squaring, it is the speed of the underlying multiplication operation that is crucial.

According to Rivest [94] multiplication on a fixed word length processor takes time proportional to the square length of the operands or $O(k^2)$. If dedicated serial/parallel hardware is constructed for the purpose, this time can be reduced to $O(k)$. In this case, the number of gates required is also proportional to the lengths of the operands, $O(k)$. The fastest implementations [15] run in time $O(\log k)$, but here the hardware requirements grow sharply to $O(k^2)$ gates.

X. Directions in Public-Key Research

Public-key cryptography has followed a curious course. In its first three years, three systems were invented. One was broken; one has generally been considered impractical; and the third reigns alone as the irreplaceable basis for a new technology. Progress in producing new public-key cryptosystems is stymied as is the complementary problem of proving the one system we have secure, or even of proving it equivalent to factoring in a useful way.

Stymied though it may be in its central problems, however, the theoretical side of public-key cryptography is flourishing. This is perhaps because the public-key problem changed the flavor of cryptography. It may be difficult to produce good conventional cryptosystems, but the difficulty is all below the surface. It is typically easier to construct a transformation that appears to satisfy the requirements of security than it is to show that a proposed system is no good. The result is a long development cycle ill-suited to the give and take of academic research. Systems that even appear to exhibit the public-key property, however, are difficult to find and this sort of difficulty is something the theoretical computer scientists can get their teeth into. The early taste of success that came with the development of RSA has inspired the search for solutions to other seemingly paradoxical problems and led to active exploration of a variety of new cryptographic disciplines.

This is not to say that contemporary research is not motivated by application. A constant caution in conventional cryptography is that the strength of a cryptosystem in one mode of operation does not guarantee its strength in another. It is widely felt, for example, that a conventional block cryptosystem such as DES is a suitable component with which to implement other modes of operation, but no proofs have been offered. This burdens anyone who chooses the system as a building block with a separate certificational examination of every configuration in which it is to be used. One objective of research in public-key cryptography has been to demonstrate the equivalence of many such secondary cryptographic problems to those that define the strength of the system. Substantial progress has been made in proving that the strength of cryptographic protocols is equivalent to the strength of the RSA system and that the protection provided by RSA is uniform [4].

There is another sort of applied flavor to even the purest of cryptographic research—a search for ways of transplanting our current social and business mechanisms to a world in which communication is primarily telecommunication. The digital signature was the first great success in this direction, which can be characterized as asking: What can we do with paper, pencil, coins, and handshakes that would be hard to do without them. And, how can we do it without them?

In 1977, I gave a talk on the problem of developing a purely electronic analog of the registered mail receipt, in the current topics session of the International Symposium on Information Theory at Cornell. My message was pessimistic, arguing for both the importance and the intractability of the problem, but fortunately my pessimism was premature. A year and a half later, the MIT group penned a note entitled "Mental Poker" [99]. It did not solve the problem of receipts for registered mail, but did show how to do something just as surprising: gamble over the telephone in a way that prevented either party from cheating without being discovered. This as it turned out was just the beginning.

To my delight, the problem of registered mail was rediscovered in Berkeley in 1982 as part of a larger category of problems that could be solved by *ping-pong protocols* and the emergence of this subject was one of the highlights of Crypto '82 [20]. Despite problems with protocols that were either broken or impossibly expensive [55], progress has been sufficient to provide hope that registered mail, contract signing, and related problems will one day have practical solutions.

In separate 1979 papers, G. R. Blakley at the University of Texas and Adi Shamir at MIT [11], [100] opened yet another direction of investigation: how secret information can be divided among several people in such a way that any k of them, but no fewer, can recover it. Although this field of *secret sharing*, unlike that of ping-pong protocols emerged full grown with provably correct and easily implementable protocols, it has been the subject of continuing examination [5], [26], [45], [58].

David Chaum, currently at the Center for Mathematics and Computer Science in Amsterdam, has applied public-key technology to a particularly challenging set of problems [21], [22]. In a society dominated by telecommunication and computers, organizations ranging from credit bureaus to government agencies can build up dossiers on private citizens by comparing notes on the credentials issued to the citizens. This dossier building occurs without the citizens' knowledge or consent and, at present, the only protection against abuses of this power lies in legal regulation. Chaum has developed technical ways of permitting an individual to control the transfer of information about him from one organization to another. Without action on the part of an individual to whom credentials have been issued, no organization is able to link the information it holds about the individual with information in the databanks of any other organization. Nonetheless, the systems guarantee that no individual can forge organizational credentials. Chaum's techniques address problems as diverse as preventing spies from tracing messages through electronic mail networks [19], [24] and protecting the privacy of participants in transactions with systems that recapture in electronic media both the assurance and the anonymity of cash [21].

The work drawing most attention at present is probably the field best known under the name of *zero-knowledge proofs* [49], [50], though similar theories, based on different assumptions about the capabilities of the participants, have been developed independently [23], [13], [14]. One of the idea's originators, Silvio Micali at MIT, described it as "the inverse of a digital signature." A zero-knowledge proof permits Alice to demonstrate to Bob that she knows something, but gives him no way of conveying this assurance to anybody else. In the original example, Alice convinced Bob that she knew how to color a map with three colors, but gave him no information whatever about what the coloring was.

The view that a zero-knowledge proof is the inverse of a digital signature now seems ironic, because a form of challenge and response authentication, applicable to the signature problem, has become the best known outgrowth of the field. In this system, the responder demonstrates to the challenger his knowledge of a secret number,, without revealing any information about what the number is. Amos Fiat and Adi Shamir have recently brought forth an identification system of this sort, and announced a proof that breaking it is equivalent to factoring [47].

A purist might respond to all this by saying that having failed to solve the real problems in public-key cryptography, cryptographers have turned aside to find other things about which to write papers. It is a situation that has been seen before in mathematics. At the end of the last century, mathematical analysis ground to a halt against intractable problems in Fourier Theory, differential equations, and complex analysis. What many mathematicians did with their time while not solving the great problems was viewed with scorn by critics who spoke of the development of point set topology and abstract algebra as "soft mathematics." Only at mid-century did it become clear what had happened. In the abstractions a great hammer had been forged and through the 1950s and 1960s the classic problems began to fall under its blows. Perhaps cryptography will be equally lucky.

XI. Where is Public Key Going?

In just over ten years, public-key cryptography has gone from a novel concept to a mainstay of cryptographic technology. It is soon to be implemented in hundreds of thousands of secure telephones and efforts are under way to apply the same mechanisms to data communications on a similar scale [97]. The outlook in the commercial world is equally bright. As early as the fourth quarter of this year, digital signatures may enter retail electronic funds transfer technology in a British experiment with point of sale terminals [57]. The demand for public key is exemplified by a recent conference on smart cards in Vienna, Austria [111], where one question was heard over and over again: When will we have an RSA card?

Now that it has achieved acceptance, public-key cryptography seems indispensable. In some ways, however, its technological base is disturbingly narrow. With the exception of the McEliece scheme and a cumbersome knapsack system devised explicitly to resist the known attacks [25], virtually all surviving public-key cryptosystems and most of the more numerous signature systems employ exponentiation over products of primes. They are thus vulnerable to breakthroughs in factoring or discrete logarithms. Key exchange systems are slightly better off since they can use the arithmetic of primes, prime products, or Galois fields with 2^n elements and are thus sensitive to progress on the discrete logarithm problem only.

From the standpoint of conventional cryptography, with its diversity of systems, the narrowness bespeaks a worrisome fragility. This worry, however, is mitigated by two factors.

- The operations on which public-key cryptography currently depends—multiplying, exponentiating, and factoring—are all fundamental arithmetic phenomena. They have been the subject of intense mathematical scrutiny for centuries and the increased attention that has resulted from their use in public-key cryptosystems has on balance enhanced rather than diminished our confidence.
- Our ability to carry out large arithmetic computations has grown steadily and now permits us to implement our systems with numbers sufficient in size to be vulnerable only to a dramatic breakthrough in factoring, logarithms, or root extraction.

It is even possible that RSA and exponential key exchange will be with us indefinitely. The fundamental nature of exponentiation makes both good candidates for eventual proof of security and if complexity theory evolves to provide convincing evidence of the strength of either, it will establish a new paradigm for judging cryptographic mechanisms. Even if new systems were faster and had smaller keys, the current systems might never be superceded altogether.

Such proofs have yet to be found, however, and proposed schemes are continually presented at the cryptographic conferences [12], [114], [80], [30], [82]. Approaches include generalizing RSA to other rings and various attempts to replace exponentials with polynomials, but in general they have not fared well and some of their fates are discussed elsewhere in this special section (E. F. Brickell and A. M. Odlyzko "Cryptanalysis: A Survey of Recent Results"). So far, the goal of improving on the performance of RSA without decreasing its security has yet to be achieved.

An appealing idea that has been put forward by Stephen Wolfram and studied by Papua Guam [54] is the use of cellular automata. Guam's system is too new to have received careful scrutiny and superficial examination suggests that it may suffer a weakness similar to one seen in other cases [46]. Even should this effort fail, however, the cellular automaton approach is attractive. Cellular automata differ from such widely accepted cryptographic mechanisms as shift registers in that, even if they are invertible, it is not possible to calculate the predecessor of an arbitrary state by simply reversing the rule for finding the successor. This makes them a viable vehicle for trap doors. Cellular automata also lend themselves to study of the randomness properties required of strong cryptographic systems [115].

What will be the outcome of such research? In an attempt to foresee the future of cryptography in 1979, I wrote [39]:

> "Prospects for development of new and more efficient public key cryptographic systems by the latter part of the eighties are quite good. Public key cryptography is more successful today than algebraic coding theory was at the age of four. The major breakthroughs in that field did not begin till the latter part of its first decade, but then progressed rapidly. The similarity of the two fields is reason for optimism that . . . public key cryptography will follow a similar course.
>
> Increasing use of the available public key systems in the 1980s will spread awareness of both their advantages and the performance shortcomings of the early examples. The research response to this awareness will probably produce better public key systems in time for use during the first half of the nineties."

My schedule was clearly too optimistic. If there are public-key cryptosystems with better performance or greater security waiting in the wings, they are proprietary systems that have yet to make even their existence known. Other aspects of the argument are closer to the mark, however. The use of public-key cryptosystems has increased dramatically and with it awareness of their advantages. Judicious use of hybrid systems and improved arithmetic algorithms have reduced the "performance shortcomings" to the status of a nuisance in most applications and the biggest motivation for seeking new systems today is probably the desire not to have all our eggs in one basket. Unless the available systems suffer a cryptanalytic disaster, moreover, the very success of public-key cryptography will delay the introduction of new ones until the equipment now going into the field becomes outmoded for other reasons.

For a discipline just entering its teens, the position of public-key cryptography should be seen not as a fragile, but as a strong one.

References

[1] L. M. Adleman and R. L. Rivest, "How to break the Lu-Lee (COMSAT) public key cryptosystem," MIT Laboratory for Computer Science, July 24, 1979.

[2] L. M. Adleman, C. Pomerance, and R. S. Rumley, "On distinguishing prime numbers from composite numbers," *Ann. Math.*, vol. 117, no. 2, pp. 173-206, 1983.

[3] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms.* Reading, MA: Addison-Wesley, 1974.

[4] W. Alexi, B. Chor, O. Goldreich, and C. P. Schnor, "RSA/Rabin bits are $1/2 + 1/(\text{poly}(\log N))$ secure," in *25th Annual IEEE Symp. on Foundations of Comp. Sci.*, pp. 449-457, 1984.

[5] C. Asmuth and J. Blum, "A modular approach to key safeguarding," *IEEE Trans. Informat. Theory*, vol. IT-29, pp. 208-210, Mar. 1983.

[6] "Contractors ready low-cost, secure telephone for 1987 service start," *Aviat. Week Space Technol.*, pp. 114-115, Jan. 1986.

[7] C. Barney, "Cypher chip makes key distribution a snap," *Electronics*, Aug. 7, 1986.

[8] J. Barron, *Breaking the Ring.* Boston, MA: Houghton Mifflin, 1987.

[9] D. ben-Aaron, "Mailsafe signs, seals, and delivers files," *Information Week*, pp. 19-22, Sept. 15, 1986.

[10] I. F. Blake, R. Fuji-Hara, R. C. Mullin, and S. A. Vanstone, "Computing logarithms in finite fields of characteristic two," *SIAM J. Alg. Disc. Methods*, vol. 5, no. 2, pp. 276-285, June 1984.

[11] G. R. Blakley, "Safeguarding cryptographic keys," in *National Computer Conf.*, pp. 313-317, 1979.

[12] G. R. Blakley and D. Chaum, Eds., *Advances in Cryptology: Proceedings of Crypto '84.* Berlin, Germany: Springer-Verlag, 1985.

[13] G. Brassard and C. Crépeau, "Non-transitive transfer of confidence: A perfect zero-knowledge interactive protocol for SAT and beyond," in *27th Annual IEEE Symp. on the Foundations of Comp. Sci.*, pp. 188-195, 1986.

[14] G. Brassard, C. Crépeau, and D. Chaum, "Minimum disclosure proofs of disclosure proofs of knowledge," Center for Mathematics and Computer Science, Amsterdam, Rep. PM-R8710, Dec. 1987. (To appear as an invited paper in *J. Comput. Syst. Sci.*)

[15] E. F. Brickell, "A fast modular multiplication algorithm with application to two key cryptography," in *Crypto '82* [20], pp. 51-60.

[16] E. F. Brickell and G. J. Simmons, "A status report on knapsack based public key cryptosystems," *Congressus Numerantium*, vol. 7, pp. 3-72, 1983. The CCIS encryptor is mentioned on pp. 4-5.

[17] E. F. Brickell, "Breaking iterated knapsacks," in *Crypto '84* [12], pp. 342-358.

[18] D. Burnham, "NSA seeking 500,000 'secure' telephones," *The New York Times*, October 7, 1984.

[19] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *CACM*, vol. 24, no. 2, pp. 84-88, Feb. 1981.

[20] D. Chaum, R. L. Rivest, and A. T. Sherman, Eds., *Advances in Cryptology, Proceedings of Crypto '82.* New York, NY: Plenum, 1983.

[21] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *CACM*, vol. 28, no. 10, pp. 1030-1044, Oct. 1985.

[22] D. Chaum and J.-H. Evertse, "A secure and privacy-protecting protocol for transmitting personal information between organizations," in *Crypto '86* [80], pp. 118-167.

[23] D. Chaum, "Demonstrating that a public predicate can be satisfied without revealing any information about how," in *Crypto '86* [80], pp. 195-199.

[24] ——, "The dining cryptographers problem: Unconditional sender untraceability," *J. Cryptology*, vol. 1, no. 1, pp. 65-75, 1988.

[25] B. Chor and R. L. Rivest, "A knapsack type public-key cryptosystem based on arithmetic in finite fields," in *Crypto '84* [12], pp. 54-65.

[26] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults," in *26th Annual IEEE Symp. on the Foundations of Comp. Sci.*, pp. 383-395, 1985.

[27] Testimony of David Earl Clark at the trial of Jerry Alfred Whitworth before Judge J. P. Vukasin, Jr., in the U.S. District Court, Northern District of California, Mar. 25, 1986. Reported by Vivian Pella Balboni, pp. 11-1345.

[28] D. Coppersmith, "Fast evaluation of logarithms in fields of characteristic two," *IEEE Trans. Informat. Theory*, vol. IT-30, pp. 587-594, 1984.

[29] D. Coppersmith, A. M. Odlyzko, and R. Schroeppel, "Discrete logarithms in GF(p)," *Algorithmica*, vol. 1, pp. 1-16, 1986.

[30] N. Cot and I. Ingemarsson, Eds., *Advances in Cryptology, Proceedings of EUROCRYPT '84.* Berlin, Germany: Springer-Verlag, 1985.

[31] "Cidec-HS high speed DES encryption for digital networks," product description, Cylink Corporation, Sunnyvale, CA.

[32] "Key management development package," product description, Cylink Corporation, Sunnyvale, CA.

[33] D. W. Davies and W. L. Price, "The applications of digital signatures based on public key cryptosystems," National Physical Laboratory Rep. DNACS 39/80, Dec. 1980.

[34] J. A. Davis, D. B. Holdridge, and G. J. Simmons, "Status report on factoring (at the Sandia National Laboratories)," in *Eurocrypt '84* [30], pp. 183-215.

[35] W. Diffie and M. E. Hellman, "Multiuser cryptographic techniques," in *Proc. Nat. Computer Conf.*, (New York, NY), pp. 109-112, June 7-10, 1976.

[36] ——, "New directions in cryptography," *IEEE Trans. Informat. Theory*, vol. IT-22, pp. 644-654, Nov. 1976.

[37] ——, "Exhaustive cryptanalysis of the NBS data encryption standard," *Computer*, vol. 10, no. 6, pp. 74-84, June 1977.

[38] W. Diffie, "Conventional versus public key cryptosystems," in [109], pp. 41-72. Rabin's system is discussed on p. 70, the relative strength of conventional and public-key distribution on pp. 64-66.

[39] ——, "Cryptographic technology: Fifteen-year forecast," in [109], pp. 301-327.

[40] ——, "Securing the DoD transmission control protocol," in *Crypto '85* [114], pp. 108-127.

[41] W. Diffie, L. Strawczynski, B. O'Higgins, and D. Steer, "An ISDN secure telephone unit," in *Proc. National Communications Forum 1987*, pp. 473-477.

[42] E. Dolnick, "N.M. scientist cracks code, wins $1000," *The Boston Globe*, Nov. 6, 1984.

[43] Electronic Industries Association, "Comsec and Compusec market study," Jan. 14, 1987.

[44] Federal Register, "Encryption algorithm for computer data protection," vol. 40, no. 52, pp. 12134-12139, Mar. 17, 1975.

[45] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *28th Annual IEEE Symp. on the Foundations of Comp. Sci.*, pp. 427-437, 1987.

[46] H. Fell and W. Diffie, "Analysis of a public key approach based on polynomial substitution," in *Crypto '85* [114], pp. 108-127.

[47] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Crypto '86* [80], pp. 186-212.

[48] M. Gardner, "A new kind of cipher that would take millions of years to break," *Sci. Amer.*, pp. 120-124 (Mathematical Games), Aug. 1977.

[49] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity and a methodology of cryptographic protocol design," in *27th Annual IEEE Conf. on the Foundations of Comp. Sci.*, pp. 174-187, 1986.

[50] S. Goldwasser, S. Micali, and C. Rackoff, "Knowledge complexity of interactive proofs," in *17th Symp. on the Theory of Computing*, pp. 291-304, 1985.

[51] S. Goldwasser and J. Killian, "All primes can be quickly certified," in *18th Symp. on the Theory of Computing*, pp. 316-329, 1986.

[52] J. Gordon, "Strong primes are easy to find," in *Eurocrypt '84* [30], pp. 215-223.

[53] ——, speech at the Zurich Seminar, 1984. In this lecture, which has unfortunately never been published, Gordon assembled the facts of Alice and Bob's precarious lives, which had previously been available only as scattered references in the literature.

[54] P. Guam, "Cellular automaton public key cryptosystem," *Complex Systems*, vol. 1, pp. 51-56, 1987.

[55] J. Hastad and A. Shamir, "The cryptographic security of trun-

cated linearly related variables," in *17th Symp. on Theory of Computing*, pp. 356-362, 1985.

[56] D. Helwig, "Coding chip devised in waterloo," *The Globe and Mail*, Jan. 1, 1987.

[57] "National EFT POS to use public key cryptography," *Information Security Monitor*, vol. 2, no. 12, p. 1, Nov. 1987.

[58] M. Ito, A. Saito, and T. Nishizeki, "Secret sharing scheme realizing general access structure," in *Globecom '87*, pp. 361-364, 1987.

[59] C. S. Kline and G. J. Popek, "Public key vs. conventional key encryption," in *National Computer Conf.*, 1979.

[60] D. Knuth, "Semi-numerical algorithms," in *The Art of Computer Programming*, vol. 2, 2nd ed. Reading, MA: Addison-Wesley, 1981, pp. 316-336.

[61] N. Koblitz, *A Course in Number Theory and Cryptography*. New York, NY: Springer-Verlag, 1987.

[62] L. M. Kohnfelder, "Toward a practical public key cryptosystem," Bachelors Thesis, MIT Dept. of Electrical Engineering, May 1978.

[63] R. Kopeck, "T1 encryption plan protects data," *PC Week*, March 3, 1987.

[64] J. Kowalchuk, B. P. Schanning, and S. Powers, "Communication privacy: Integration of public and secret key cryptography," in *National Telecommunications Conf.*, (Houston, TX), pp. 49.1.1-5, Nov. 30-Dec. 4, 1980.

[65] E. Kranakis, *Primality and Cryptography*. New York, NY: Wiley, 1986.

[66] R. Lindsey, *The Falcon and the Snowman*. New York, NY: Simon and Schuster, 1979.

[67] S. Lu and L. Lee, "A simple and effective public key cryptosystem," *Comsat Technical Rev.*, vol. 9, no. 1, Spring 1979.

[68] K. S. McCurley, "A key distribution system equivalent to factoring," Department of Mathematics, University of Southern California, June 3, 1987.

[69] R. J. McEliece, "A public key cryptosystem based on algebraic coding theory," JPL DSN Progress Rep. 42-44, pp. 114-116, Jan.-Feb. 1978.

[70] R. Merkle, "Secure communication over insecure channels," *CACM*, pp. 294-299, Apr. 1978.

[71] R. C. Merkle and M. E. Hellman, "Hiding information and signatures in trap door knapsacks," *IEEE Trans. Informat. Theory*, vol. IT-24, pp. 525-30, Sept. 1978.

[72] R. Merkle, Letters to the Editor, *Time Magazine*, vol. 120, no. 20, p. 8, Nov. 15, 1982.

[73] M. A. Morrison and J. Brillhart, "A method of factoring and the factorization of F_7," *Math. Comp.*, vol. 29, pp. 18-205, 1975.

[74] "Advanced techniques in network security," Motorola Government Electronics Division, Scottsdale, AZ, about 1977.

[75] F. H. Myers, "A data link encryption system," in *National Telecommunications Conf.*, (Washington, DC), pp. 4.5.1-4.5.8, Nov. 27-29, 1979.

[76] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *CACM*, vol. 21, pp. 993-999, Dec. 1978.

[77] L. Neuwirth, "A comparison of four key distribution methods," *Telecommunications*, pp. 110-111, 114-115, July 1986.

[78] "Statement of Lee Neuwirth of Cylink on HR145," submitted to Congressional committees considering HR145, Feb. 1987.

[79] A. M. Odlyzko, "Discrete logarithms in finite fields and their cryptographic significance," in *Eurocrypt '84* [30], pp. 225-314.

[80] —, Ed., *Advances in Cryptology-CRYPTO '86*. Berlin, Germany: Springer-Verlag, 1987.

[81] B. O'Higgins, W. Diffie, L. Strawczynski, and R. de Hoog, "Encryption and ISDN-A natural fit," in *International Switching Symp.*, (Phoenix, AZ), pp. A11.4.1-7, Mar. 16-20, 1987.

[82] F. Pichler, Ed., *Advances in Cryptology-Proceedings of EUROCRYPT '85*. Berlin, Germany: Springer-Verlag, 1986.

[83] S. C. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms in GF(p) and its cryptographic significance," *IEEE Trans. Informat. Theory*, vol. IT-24, pp. 106-110, Jan. 1978.

[84] C. Pomerance, "Recent developments in primality testing," *The Mathematical Intelligence*, vol. 3, no. 3, pp. 97-105, 1981.

[85] C. Pomerance, J. W. Smith, and R. Tuler, "A pipe-line architecture for factoring large integers with the quadratic sieve algorithm," to appear in a special issue on cryptography of the *SIAM J. Computing*.

[86] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," MIT Laboratory for Computer Science, MIT/LCS/TR-212, Jan. 1979.

[87] "Datacryptor II, public key management option," Racal-Milgo, Sunrise Florida, 1981.

[88] "AT&T readying new spy-proof phone for big military and civilian markets," *The Report on AT&T*, pp. 6-7, June 2, 1986.

[89] R. F. Rieden, J. B. Snyder, R. J. Widman, and W. J. Barnard, "A two-chip implementation of the RSA public-key encryption algorithm," in *GOMAC* (Government Microcircuit Applications Conference), (Orlando, FL), pp. 24-27, Nov. 1982.

[90] R. L. Rivest, A. Shamir, and L. Adleman, "On digital signatures and public key cryptosystems," MIT Laboratory for Computer Science, MIT/LCS/TR-212, Jan. 1979.

[91] —, "A method for obtaining digital signatures and public key cryptosystems," *CACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[92] R. Rivest, personal communication with H. C. Williams cited on p. 729 in [113].

[93] —, "A description of a single-chip implementation of the RSA cipher," *Lambda*, vol. 1, no. 3, pp. 14-18, Fall 1980.

[94] —, "RSA chips (past/present/future)," in *Eurocrypt '84* [30], pp. 159-165.

[95] H. L. Rogers, "An overview of the caneware program," paper 31, presented at the 3rd Annual Symp. on Physical/Electronic Security, Armed Forces Communications and Electronics Association, Philadelphia Chapter, Aug. 1987.

[96] "Toward a new factoring record," *Science News*, p. 62, Jan. 23, 1987.

[97] "SDNS: A network on implementation," in *10th National Computer Security Conf.*, (Baltimore, MD), pp. 150-174, Sept. 21-24, 1987. Session containing six papers on the Secure Data Network System.

[98] A. Shamir, "A fast signature scheme," M.I.T. Laboratory for Computer Science, Technical Memorandum, MIT/LCS/TM-107, July 1978.

[99] A. Shamir, R. L. Rivest, and L. M. Adleman, "Mental poker," MIT Laboratory for Computer Science, Technical Memorandum, MIT/LCS/TM-125, Jan. 29, 1979.

[100] A. Shamir, "How to share a secret," *CACM*, vol. 22, no. 11, pp. 612-613, Nov. 1979.

[101] —, "A polynomial time algorithm for breaking Merkle-Hellman cryptosystems (extended abstract)," Research announcement, preliminary draft, Applied Mathematics, Weizmann Institute, Rehovot, Israel, April 20, 1982. This paper appeared with a slightly different title: "A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem (extended abstract)," in *Crypto '82* [20], pp. 279-288.

[102] —, "A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem," *IEEE Trans. Informat. Theory*, vol. IT-30, no. 5, pp. 699-704, Sept. 1984.

[103] Z. Shmuely, "Composite Diffie-Hellman public-key generating systems are hard to break," Computer Science Department, Technion, Haifa, Israel, Technical Rep. 356, Feb. 1985.

[104] R. Silver, "The computation of indices modulo P," Mitre Corporation, Working Paper WP-07062, p. 3, May 7, 1964.

[105] G. J. Simmons and M. J. Norris, "Preliminary comments on the M.I.T. public key cryptosystem," *Cryptologia*, vol. 1, pp. 406-414, Oct. 1977.

[106] G. J. Simmons, "Authentication without secrecy: A secure communications problem uniquely solvable by asymmetric encryption techniques," in *IEEE EASCON '79* (Washington, DC), pp. 661-662, Oct. 9-11, 1979.

[107] G. J. Simmons and M. J. Norris, "How to cipher faster using redundant number systems," Sandia National Laboratories, SAND-80-1886, Aug. 1980.

[108] G. J. Simmons, "High speed arithmetic utilizing redundant number systems," in *National Telecommunications Conf.*, (Houston, TX), pp. 49.3.1-2, Nov. 30-Dec. 4, 1980.

[109] —, Ed., *Secure Communications and Asymmetric Cryptosystems*. AAAS Selected Symposium 69. Boulder, CO: Westview Press, 1982.

[110] ——, "Cryptology," in *Encyclopaedia Britannica, 16th Edition*. Chicago, IL: Encyclopaedia Britannica, 1986, pp. 913–924B.

[111] Proceedings of Smart Card 2000, Vienna, Austria, Oct. 19–20, 1988.

[112] M. V. Wilkes, *Time-Sharing Computer Systems*. New York, NY: American Elsevier, 1972.

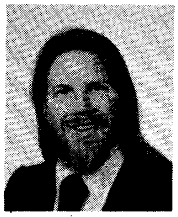
[113] H. C. Williams, "A modification of the RSA public-key cryptosystem," *IEEE Trans. Informat. Theory*, vol. IT-26, no. 6, pp. 726–729, Nov. 1980.

[114] ——, Eds., *Advances in Cryptology-CRYPTO '85*. Berlin, Germany: Springer-Verlag, 1986.

[115] S. Wolfram, "Cryptography with cellular automata," in *Crypto '85* [114], pp. 429–432.

[116] M. C. Wunderlich, "Recent advances in the design and implementation of large integer factorization algorithms," in *1983 Symp. on Security and Privacy*, (Oakland, CA, pp. 67–71, Apr. 25–27, 1983.

[117] K. Yiu and K. Peterson, "A single-chip VLSI implementation of the discrete exponential public key distribution system," in *GOMAC* (Government Microcircuit Applications Conference), (Orlando, FL), pp. 18–23, Nov. 1982.



Whitfield Diffie was born in Washington, DC, on June 5, 1944. He received the B.S. degree in mathematics from the Massachusetts Institute of Technology, Cambridge, MA, in 1965.

While at Mitre Corporation from 1965 to 1969, he worked with Carl Engelman in developing the Mathlab symbolic mathematical manipulation system, later expanded at MIT to become Macsyma. In 1969, he transferred to the Stanford University Artificial Intelligence Laboratory to work with John McCarthy on proof checking and proof of correctness of programs. While there he also developed the compiler adopted for the U.C. Irvine llisp system. In 1973, Diffie took leave from Stanford and began his work on cryptography while traveling around the U.S. He continued this work as a graduate student under Martin Hellman at Stanford University from 1975 through 1978. Since 1978, Diffie has been the Manager of Secure Systems Research for Bell-Northern Research, Mountain View, CA. His most recent work has been on key management protocols for telephones designed to operate on the developing Integrated Services Digital Network.