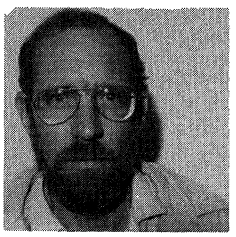
Cryptology: From Caesar Ciphers to Public-key Cryptosystems

Dennis Luciano
Gordon Prichett

Dennis Luciano is currently Associate Professor and Chairman of the Department of Mathematics and Computer Science of Western New England College in Springfield, Massachusetts, where he has been since 1977. Prior to this, he had appointments at Saint Joseph's University (Philadelphia, PA) and LeMoyne College (Syracuse, NY). He received an M.A. and Ph.D. (in commutative algebra under the direction of David Lissner) from Syracuse University. Professor Luciano is a referee for the College Mathematics Journal, a member of the Executive Committee of the Northeastern Section of the MAA, and an evaluator for the New England Association of Schools and Colleges. His current mathematical interests include number theory, graph theory, and applications of mathematics at the undergraduate level. His nonmathematical interests include running and basketball.



Gordon Prichett is Professor of Mathematics at Babson College. He received his B.A. from Williams College in 1963 and his Ph.D. from the University of Wisconsin, Madison in 1970. After teaching mathematics and mathematics education at San Diego State University, he moved to Hamilton College where he served as chairman. He has since taught at Wellesley College and Babson. He has published a variety of papers on number theory and computer applications. Dr. Prichett has spent one year as a visiting professor in the Biology Department at the University of York, England, studying population biology. He has an active interest in curriculum development and alternative approaches to instruction, and is coauthor of A Supplement to Calculus: One and Several Variables, which adapts introductory calculus to the self-paced mode of instruction.

> Few persons can be made to believe that it is not quite an easy thing to invent a method of secret writing that shall baffle investigation. Yet it may be roundly asserted that human ingenuity cannot concoct a cipher which human ingenuity cannot resolve.
>
> Edgar Allen Poe [7]

Cryptography, from the Greek 'kryptos' (hidden) and 'graphein' (to write), is the art and science of making communications unintelligible to all except the intended recipients. Its existence through the ages was usually confined to diplomatic and military circumstances, where it was used for the concealment of information communicated over secure and insecure lines. Today, there is an urgent need to

provide cost-effective, efficient, and secure systems to protect the vast quantity of digital data stored and communicated by electronic data-processing systems. With the growth in electronic fund transfers, instant electronic mail, point-of-sale terminals, home banking, and conferencing through computers, the threat of unauthorized accessibility to this data becomes a pressing concern of our society.

The science of reading secret messages and uncovering the cryptographic system utilized is called *cryptanalysis*. This science played a vital role in the conclusion of the Second World War. The early breaking of the Purple Code of Japan allowed the Allied forces to continually read secret messages pertaining to strategic movements in the Pacific; the recovery of an Enigma Machine from a sunken German submarine did the same for the Allied command in Europe [9], [11].

Edgar Allen Poe fancied himself a skilled cryptanalyst. But there is some doubt about the defensability of his quoted dictum. Cryptanalysis reveals deficiencies in existing cryptographic systems. Improved cryptographic systems then pose new problems for the cryptanalyst. The intent of this survey is to discuss the mutually reactive relationship between the two areas, with an emphasis on the underlying mathematics.

Linear Ciphers

A *cipher* is a system which transforms plaintext into ciphertext by applying a set of transformations to each character (or letter) in the plaintext. The particular transformations employed at any time are controlled by a "key" used at that time. Security of the ciphertext rests heavily on the secrecy of the key; it is the objective of the cryptanalyst to find the key and consequently break the system.

Caesar Ciphers. One of the earliest known cryptographic systems was used by Julius Ceasar and is appropriately referred to as a Caesar Cipher [26]. Around 50 B.C., Julius Ceasar wrote to Marcus Cicero, using a cipher that shifts the alphabet three places to the right and wraps the last three letters X, Y, Z back onto the first three letters:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Thus, the plaintext message

MEET YOU IN ORLANDO

is transformed into the ciphertext

PHHW BRX LQ RUODQGR.

Such a transformation, using modular arithmetic, can also be performed by a computer. Any message can be expressed digitally based on the one-to-one correspondence,

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25.

To encipher the plaintext message, we use the transformation

$$E(M) \equiv (M + 3) \text{ (modulo } 26),$$

where M is the numeric equivalent of a plaintext letter. To decipher, we utilize the transformation

$$D(C) \equiv (C + 23) \text{ (modulo } 26),$$

where C is the numeric equivalent of a ciphertext letter. Hence:

O	R	L	A	N	D	O
14	17	11	00	13	03	14
		E: ↓		D: ↑		
17	20	14	03	16	06	17
R	U	O	D	Q	G	R

It is not known why Ceasar selected 3 as the key to his cipher system; he could have chosen any integer value. Since we are operating modulo 26, there are 26 distinct keys, one of which (the identity) offers no secrecy at all. A message enciphered by a Caesar Cipher is extremely insecure since exhaustive cryptanalysis using the 25 nontrivial keys is easily performed.

Decimation and Linear Ciphers. A Decimation Cipher is similar to a Caesar Cipher but it uses multiplication, rather than addition, by a number key. In order to assure a one-to-one correspondence among the letters of the alphabet, the key number must be relatively prime to 26. [For example, the multiplier 2 yields $E(A) = E(N) = A$, since $0 \equiv 2(13) \pmod{26}$.] Thus, the enciphering transformation must be

$$E(M) \equiv kM \pmod{26},$$

where M is the numeric value of a plaintext letter, and the key number k is an odd integer which is not a multiple of 13. If $k = 3$, for example, the cipher alphabet is obtained by starting with A and selecting every third letter that follows in the plain alphabet as we cycle through it. The deciphering algorithm is then

$$D(C) \equiv 9C \pmod{26},$$

where C is the numeric equivalent of a ciphertext letter. In general, the deciphering key is the multiplicative inverse modulo 26 of the enciphering key k (the inverse must exist since k is relatively prime to 26). A decimation cipher offers no more security than a Caesar Cipher since there are only twelve distinct keys.

Decimation and Caesar Ciphers are subcases of a more general class of ciphers called linear or affine ciphers. A *linear cipher* is defined by the enciphering transformation

$$E(M) \equiv (kM + t) \pmod{26}, \tag{1}$$

where k, t are integers and k is relatively prime to 26. There are (12) (26) = 312 distinct linear ciphers.

Suppose the following ciphertext was generated using a linear cipher:

YHTQF SCUFD SBULX IOLBF ALYZT IDSCL YCSDO
FZYCU FAFMF ODITF YCDKV SBICD XBXCF TX.

We can compare the frequency distribution of the letters occurring in this ciphertext (Table 1) with the expected frequency of occurrence of letters in the English Language (Table 2).

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2	4	7	6	0	9	0	1	4	0	1	4	1	0	3	0	1	0	5	4	3	1	0	4	5	2

Table 1. Frequencies of letters in ciphertext.

Letter	Frequency	Probability	Letter	Frequency	Probability
A	321712	.0804	N	283561	.0709
B	61472	.0154	O	303844	.0760
C	122403	.0306	P	79845	.0200
D	159726	.0399	Q	4226	.0011
E	500334	.1251	R	244867	.0612
F	92100	.0230	S	261470	.0654
G	78434	.0196	T	370072	.0925
H	219481	.0549	U	108516	.0271
I	290559	.0726	V	39504	.0099
J	6424	.0016	W	76673	.0192
K	26972	.0067	X	7779	.0019
L	165559	.0414	Y	69334	.0173
M	101339	.0253	Z	3794	.0009

Table 2. Individual letter frequencies in 4 million characters of English text [6] (Based on a sample of 8000 excerpts of 500 letters taken from the Brown University Corpus of Present-Day American English).

The two letters which occur most frequently in the English Language are E and T, in that order. Therefore, it is reasonable to guess that F, which occurs most frequently in the ciphertext (see Table 1), corresponds to E, while the second most frequent letter C in this ciphertext corresponds to T. Based on the assumption that $E(4) = 5$ and $E(19) = 2$, we use (1) to generate the linear congruences

$$4k + t \equiv 5 \pmod{26}$$

$$19k + t \equiv 2 \pmod{26}.$$

The solution to this system is $k = 5$ and $t = 11$, obtained via elementary techniques described in most number theory texts. Therefore, the conjectured enciphering transformation would be

$$E(M) \equiv (5M + 11) \pmod{26}.$$

Since the multiplicative inverse modulo 26 of 5 is 21, and the additive inverse modulo 26 of 11 is 15, the corresponding deciphering transformation would be

$$D(C) \equiv 21(C + 15) \pmod{26}.$$

Based on this, we would decipher our earlier ciphertext as

ciphertext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	3	24	19	14	9	4	25	20	15	10	5	0	21	16	11	6	1	22	17	12	7	2	23	18	13	8
plaintext	D	Y	T	O	J	E	Z	U	P	K	F	A	V	Q	L	G	B	W	R	M	H	C	X	S	N	I

NUMBE RTHEO RYHAS PLAYE DANIM PORTA NTROL
EINTH EDEVE LOPME NTOFC RYPTO SYSTE MS

that is,

NUMBER THEORY HAS PLAYED AN IMPORTANT ROLE
IN THE DEVELOPMENT OF CRYPTOSYSTEMS.

Substitution Ciphers

Ceasar, decimation, and linear ciphers form a small subset of a class of ciphers known as substitution ciphers. The key to a *substitution cipher* is a permutation of the twenty-six letters of the alphabet. For example, such a key may be

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
K Y A X J B Z I N W C M G H V D T L U S P E R F O Q,

which enciphers MICKEY MOUSE as GNACJO GVPUJ.

This type of enciphering increases the number of possible keys to 26! (a number exceeding 4×10^{26}), and this effectively eliminates exhaustive cryptanalysis. To decipher a ciphertext encrypted by a general substitution cipher, one must use the statistical frequency distribution of single-letter occurrences in the English Language. If this doesn't suffice, more information can be obtained from language patterns [26]. By comparing frequency occurrences for letters, initial letters, final letters, digraphs (combinations of two letters), and trigraphs (combinations of three letters) of a ciphertext with known characteristic frequencies of the English Language, one can eventually reconstruct the key and decipher the text.

This technique is especially efficient because each plaintext letter is represented always by the same ciphertext letter. Consequently, all the properties of the plaintext language are retained in the ciphertext language. These invariant properties of substitution ciphers can be utilized to obtain the key and break the cipher. It can be shown that an average of twenty-eight letters of ciphertext is needed to uniquely solve for the key of a substitution cipher [11].

Polyalphabetic Ciphers

One way to obtain greater security is to use a cipher which guarantees that a given plaintext letter is not always represented by the same ciphertext letter. This can be achieved by using a sequence of n different substitution ciphers, in periodic fashion with period n, to encipher a message. The use of such substitution ciphers increases the effective number of possible keys from 26! to $(26!)^n$. For $n = 3$, the magnitude of this number is approximately equal to the total number of atoms in the universe!

A classic polyalphabetic enciphering procedure, devised by the French cryptographer Vigenère, utilizes both a keyword and the Caesar cipher transformation. Suppose we wish to encipher the plaintext message

MEET YOU AT SPACE MOUNTAIN,

using the keyword MATH. To encipher the message, one uses the sequence of four different Caesar ciphers, where the key values 12, 0, 19, 7 are the respective numeric equivalents of M, A, T, H. This sequence is applied consecutively to groups of four letters of plaintext, converting each group into ciphertext. For the first group of four letters, MEET, we obtain YEXA. Specifically, for $E_k(M)$ denoting $(M + k) \pmod{26}$:

Keyword	M	A	T	H
	12	0	19	7

Plaintext	M	E	E	T
	12	4	4	19
	$E_{12}\downarrow$	$E_0\downarrow$	$E_{19}\downarrow$	$E_7\downarrow$
	24	4	23	0
Ciphertext	Y	E	X	A

Proceeding in this manner for each consecutive group of four letters, we obtain the ciphertext

YEXAKONHFSIHOEFVGNMHUNSG.

Note that the double E in the plaintext MEET no longer appears as a double letter in the ciphertext. Furthermore, the two F's in the ciphertext correspond to different plaintext letters. Since plaintext letters are not always represented by the same ciphertext letters, the non-uniform frequency distributions of single letters, digraphs, trigraphs, etc. in plaintext are smoothed out in the ciphertext. This smoothing out improves the security of the system significantly.

An Unbreakable Cipher. To obtain a truly "unbreakable" system, one can select a polyalphabetic cipher whose key, consisting of randomly selected numbers, has the length of the plaintext message [13], [25]. This cipher, called a "one-time pad" or Vernian cipher, was invented by the U.S. Army Corps in 1917. The one-time pad is invincible since it is equiprobable that a plaintext character is represented by any ciphertext character, and revealing patterns no longer exist because each choice of representation is random. The number of possible keys of length n is $(26)^n$, which for large n makes exhaustive cryptanalysis infeasible. Note, for example, that $(26)^{56} \approx 10^{79}$. This cipher is also theoretically unbreakable, as the following example illustrates.

Plaintext	S	E	N	D	M	O	R	E		M	O	N	E	Y	
	18	04	13	03	12	14	17	04		12	14	13	04	24	
Key	09	26	01	07	23	15	21	14		11	11	02	08	09	
	01	04	14	10	09	03	12	18		23	25	15	12	07	
Ciphertext	B	E	O	K	J	D	M	S		X	Z	P	M	H	
Plaintext	R	U	N		T	O		E	X	E	R	C	I	S	E
	17	20	13		19	14		04	23	04	17	02	08	18	04
Key	10	10	01		17	21		25	15	14	06	23	07	20	03
	01	04	14		10	09		03	12	18	23	25	15	12	07
Ciphertext	B	E	O		K	J		D	M	S	X	Z	P	M	H

There is no logical basis for determining which of the two plaintext messages above corresponds to the ciphertext

BEOKJDMSXZPMH.

Using an appropriate sequence of values in the key, any thirteen-letter message can be enciphered to yield the same ciphertext.

The one-time pad is one method of secret radio communication used by the U.S.S.R. [10]. It is also employed on the "hot line" between Moscow and Washington. Even though this system offers absolute secrecy, it poses key-management problems of enormous proportions: both the sender and the receiver of a message must have the identical sequence of random numbers in order to communicate via such a system. Although a key could be sent in advance each time a message is to be communicated, unnoticed interception of the key would jeopardize the secrecy of the forthcoming communication. Furthermore, the key can only be used once since repeated use would generate recognizable patterns for the cryptanalyst's use.

The problem of generating, distributing, and cancelling keys is unmanageable in a system where there is a high traffic volume. In wartime, millions of key characters would be needed daily [10]. Even with today's powerful computers, this system is expensive and time consuming.

Public-key Cryptosystems

In 1975, a significant new type of cipher system, called a public-key cryptosystem, was proposed by Whitfield Diffie and Martin Hellman [4]. The security of this new system is not measured by the complexity of the enciphering algorithm, nor is it measured by statistical uncertainty. Instead, the system's security relies on discoveries in a young and important branch of computer science called computational complexity theory. Complexity theory primarily deals with the analysis and design of algorithms, and especially with the number of computational steps needed to complete an algorithm. The security of any cipher is now measured by the expected amount of time a computer would expend to break the cipher.

In a "public-key cryptosystem," each user in the communication network places an encryption algorithm E and cipher key in a *public* file (analogous to a telephone directory). The corresponding decryption algorithm D of the user is kept secret. Each user in the system must select his own encryption-decryption pair so as to satisfy the following properties:

(i) If $\mathscr{P}$ and $\mathscr{C}$, respectively, represent all feasible plaintext and ciphertext messages, then E: $\mathscr{P} \rightarrow \mathscr{C}$ and D: $\mathscr{C} \rightarrow \mathscr{P}$

(ii) E and D are inverse transformations; that is,

$$E(D(C)) = C \quad \text{and} \quad D(E(M)) = M \text{ for all } C \in \mathscr{C} \text{ and } M \in \mathscr{P}.$$

(iii) The pair (E, D) can be easily found (in the computational sense) by the owner, and E and D are easy to compute.

(iv) It is computationally infeasible for anyone except the owner to determine D, even if its inverse E is known.

Because of conditions (ii) and (iv), we call E a "trapdoor" or "one-way" function. It is "one-way" in the sense that for given M it is easy to compute $E(M)$; but given $E(M)$, it is effectively impossible to compute M unless certain private (trapdoor) information is known. According to convention in computational complexity theory, a function is "easy" to compute if there exists an algorithm which computes it using approximately kd^{α} computational steps, where k and α are constants, and d is the "size" of the input. If no such algorithm exists, it is said to be "hard" to compute.

Let us inspect more closely how the system works. Suppose person A desires to send message M to person Z. Person A looks up, in the public directory, the public enciphering algorithm E_z of Z and sends the message as $E_z(M)$. (There is absolutely no fear of interception—wiretapping and spying—since Z is the only individual that has access to the deciphering algorithm, D_z.) Then, Z deciphers the message by applying the algorithm D_z to $E_z(M)$. Specifically, $D_z(E_z(M)) = M$.

A major advantage of this system is that it avoids the necessity of distributing a new key before the message is sent. This is obviously important for the success of any electronic mail system.

The public-key cryptosystem also allows "signatures." A *signature* is a guarantee that the message has been issued by the sender. Signatures are a welcome bonus of the system. They are used in many contexts (electronic fund transfers, electronic mail, home banking, conferencing through computers, etc.).

If person A desires to send person Z a message, signed to insure its origin, then A can use his private deciphering algorithm D_A as a signature. Specifically, A sends $E_z[D_A(M)]$ to Z, and Z uses $E_A\{D_z(E_z[D_A(M)])\} = M$. Note that only Z can read the encrypted message $E_z[D_A(M)]$ since only Z knows the private deciphering algorithm D_z. When D_z is applied, Z is left with the text $D_A(M)$, which is still

unreadable. But then Z can read this message by applying the public enciphering algorithm E_A of A. If the message is now meaningful, Z has the assurance that it was sent by A, since only A knows the private deciphering algorithm D_A. It should be noted that condition (ii) of the public-key cryptosystem can be relaxed without any great damage. If $D(E(M)) = M$ for all $M \in \mathscr{P}$ and $E(D(C)) = C$ for only a fraction of the possible $C \in \mathscr{C}$, then signatures can still be obtained [12].

The enormous key management problems of the one-time pad system are virtually nonexistent in a public-key cryptosystem. Since each user of the system generates only one pair of keys, and since there is a public directory for the enciphering keys of the users, there are no distribution problems. Theoretically, this system overcomes the primary deficiency of the unbreakable one-time pad system, while it apparently maintains the same level of security.

The RSA Public-key Cryptosystem

In 1977, Ronald L. Rivest, Adi Shamir, and Leonard Adelman, all of MIT, developed an elegant way to implement the Diffie-Hellmen system, using only elementary ideas from number theory [21], [22]. A little background is necessary to completely understand the workings of their system, referred to as the RSA cryptosystem.

In 1640, Pierre Fermat communicated a result to one of his correspondents, Frènicle de Bessy, an official at the French mint. Frènicle's unique ability to work successfully with large numbers represented a challenge to Fermat, and consequently, he was sometimes the recipient of Fermat's most guarded results. This communication contained what is now known as Fermat's Little Theorem, along with the comment "I would sent you the demonstration, if I did not fear it being too long."

Fermat's Little Theorem. *If p is a prime and a is a positive integer not divisible by p, then $a^{p-1} \equiv 1 \pmod{p}$.*

(Euler published the first proof of this theorem in 1763, although Leibniz had written an identical argument in an unpublished manuscript before 1683.)

Armed with Fermat's Little Theorem, we can now design our public-key cryptosystem. To generate the keys, each user in the system must select two large distinct prime numbers p and q, each about 100 digits long. (As we shall discuss later, there exist efficient algorithms that can generate a one-hundred digit prime on a high speed computer in less than two minutes.) Now let $n = pq$ and choose an integer e between 3 and n such that e is relatively prime to $\phi(n) = (p-1)(q-1)$. (Note that any prime number e satisfying $\max\{p, q\} < e < n$ will suffice.) Since $(e, \phi(n)) = 1$, we know that e has a multiplicative inverse modulo $\phi(n)$; call it d. Thus, $ed \equiv 1 \pmod{\phi(n)}$. Moreover, the Euclidean Algorithm can be used efficiently to find both this positive integer d and a corresponding negative integer β such that $ed + \beta\phi(n) = 1$. The user then submits e and n to the public directory, and keeps d, p and q private.

Suppose we desire to encipher a plaintext message. Using the mapping $A \leftrightarrow 00$, $B \leftrightarrow 01, \ldots, Z \leftrightarrow 25$, convert the message of t letters to an integer M consisting of $2t$ digits. Partition the consecutive digits of M into blocks M_i of equal length so that each block represents a number less than $n = pq$. The enciphering transformation is given by

$$E(M_i) \equiv M_i^e \pmod{n} \text{ with } 0 \leq E(M_i) < n.$$

To decipher the numeric ciphertext C generated by enciphered cipher blocks C_i, use the transformation

$$D(C_i) \equiv C_i^d (\text{mod } n) \quad \text{with} \quad 0 \leq D(C_i) < n$$

on each cipher block C_i.

As an example of an RSA encryption-decryption pair, consider the following small, but illustrative, example. Let $p = 47$ and $q = 61$ be the two selected primes. Then $n = pq = 2867$, and $\phi(n) = (p-1)(q-1) = (46)(60) = 2760$. Selecting $e = 49$, we use the Euclidean Algorithm to verify that it is relatively prime to $\phi(n)$:

$$\begin{aligned} \phi(n) &= 56(49) + 16 \\ 49 &= 3(16) + 1 \\ 16 &= 16(1) + 0. \end{aligned}$$

The last nonzero remainder 1 is the greatest common divisor of $\phi(n)$ and e. Since $\phi(n)$ and e are relatively prime, $e = 49$ has a multiplicative inverse modulo $\phi(n)$; call it d. Using the equalities above in reverse, it is easy to find d:

$$\begin{aligned} 1 &= 49 - 3(16) \\ 1 &= 49 - 3[\phi(n) - 56(49)] = 169(49) + (-3)(2760). \end{aligned}$$

Consequently, $169(49) \equiv 1 (\text{mod } 2760)$. Therefore, $d = 169$.

The resulting encryption algorithm is

$$E(M) \equiv M^{49} (\text{mod } 2867),$$

and the decryption algorithm is

$$D(C) \equiv C^{169} (\text{mod } 2867),$$

where M and C are numeric blocks less than 2867. To encipher the plaintext message

EPCOT CENTER IS SPECTACULAR

translate first into numeric form

04 15 02 14 19 02 04 13 19 04 17 08 18 18 15 04 02 19 00 02 20 11 00 17

and partition into blocks of digit-length 4. Then encrypt and decrypt each block by the functions E and D, as defined above. The result is:

Original Block M_i	Encryption $E(M_i)$	Decryption $D[E(M_i)]$
0415	2261	0415
0214	2536	0214
1902	2329	1902
0413	2243	0413
1904	1416	1904
1708	1464	1708
1818	2688	1818
1504	1504	1504
0219	1544	0219
0002	2264	0002
2011	0786	2011
0017	2328	0017

Translating the right-hand column reveals the correct message.

In general, do E and D satisfy the four conditions of a public-key cryptosystem? If we define $\mathscr{P} = \mathscr{C}$ to be the set of all nonnegative integers less than n, then condition (i) is satisfied. To verify condition (ii), $E(D(C)) \equiv C^{de} \equiv C \pmod{n}$ and $D(E(M)) \equiv M^{ed} \equiv M \pmod{n}$ for $C \in \mathscr{C}$ and $M \in \mathscr{P}$, it suffices to prove

$$M^{ed} \equiv M \pmod{n} \quad \text{for all nonnegative integers} \quad M < n = pq. \tag{2}$$

If p doesn't divide M,

$$M^{p-1} \equiv 1 \pmod{p}$$

by Fermat's Little Theorem. Now recall that $\beta < 0$ and $ed + \beta\phi(n) = 1$. Since $(p-1)$ divides $\phi(n)$,

$$M^{-\beta\phi(n)} \equiv 1 \pmod{p}$$

and it follows that

$$M^{-\beta\phi(n)+1} \equiv M \pmod{p}.$$

Thus, $M^{ed} \equiv M \pmod{p}$ whenever p is not a factor of M. But clearly this also holds when p does divide M. In a similar manner,

$$M^{ed} \equiv M \pmod{q}.$$

Since p and q are relatively prime, (2) holds for all $M \in \mathscr{P}$.

Note that our enciphering transformation was defined only on nonnegative integers less than n in order to guarantee that it is one-to-one.

To verify condition (iii), we must first show that the pair (E, D) can be easily found from a computational point of view. The algorithm for determining this pair begins with finding two distinct one-hundred digit primes. There are several primality tests available—as, for example, Rabin's test [18] and the Solovay–Strassen test [27]. The former uses a test due to G.L. Miller:

> Let m be a positive integer with $m - 1 = 2^s t$, where s is a nonnegative integer and t is an odd positive integer. The integer m is said to *pass Miller's test for the base b* if either $b^t \equiv 1 \pmod{m}$ or $b^{2^j t} \equiv -1 \pmod{m}$ for some integer $j \in [0, s-1]$.

To test a number m for primality using Rabin's test, perform Miller's test for k bases each less than m. If the integer m passes all k tests, the probability that it is *not* prime is $(1/4)^k$. For $k = 100$, this probability is $(1/4)^{100} \approx 10^{-60}$. Although this test is not deterministic in nature, it gives a high degree of certainty.

Recently, a deterministic primality algorithm was announced that has "nearly" polynomial run time [2], [16]. The algorithm is quite complicated, both in theory and in practice, since many different cases are covered in computer program implementation. Motivated by this test, there have appeared other algorithms dependent on the work of H. Cohen and H. W. Lenstra. One algorithm with particularly impressive run times was implemented on a CDC Cyber 170-175 computer and can test a 100-digit number in about 30 seconds, a 200-digit number in 8 minutes, and a 1000-digit number in a week [5], [19].

How many 100-digit numbers must be tested before we find one which passes Rabin's test and is (most likely) a prime? In 1793, Gauss conjectured that if $\pi(x)$ represents the number of primes not exceeding x, then $\pi(x) \approx x/\ln x$ for large x. This was proved independently by Hadamard and de la Vallée Poussin (1896), and

is referred to as the Prime Number Theorem. The probability that a randomly selected odd integer in the interval from 10^y to 10^x is prime is approximately

$$\frac{\text{number of primes}}{\text{number of odd integers}} = \frac{\pi(10^x) - \pi(10^y)}{(10^x - 10^y)/2} \approx \frac{2}{10^{x-y} - 1}\left(\frac{10^{x-y}}{x \ln 10} - \frac{1}{y \ln 10}\right).$$

Consequently, the probability that a 100-digit number is prime is approximately

$$\frac{2}{9}\left(\frac{10}{100 \ln 10} - \frac{1}{99 \ln 10}\right) \approx .008676.$$

On the average,

$$\frac{1}{.008676} \approx 115$$

odd 100-digit numbers would be tested before a prime is found. The entire procedure outlined above requires only a few minutes of computer time to find a 100-digit prime. Although this may seem to be complicated, keep in mind that each user needs to complete this procedure only twice. Once p, q are obtained, one can find positive integers d and e, as described earlier. Thus, the pair (E, D) can be "easily" found by the owner. Moreover, E and D are "easy" to compute using modular exponentiation [17]. A 200-digit message can be encrypted or decrypted in a few seconds on a high speed computer.

To complete the verification that the RSA cryptosystem is a public-key cryptosystem, it suffices to show that E is a trapdoor function. Recall that the public-key information is e and n, whereas d, p, and q are kept private. If d is known, any ciphertext $C = E(M)$ can be read. But how "difficult" is it to find d? By definition, d satisfies $ed \equiv 1 \pmod{\phi(n)}$. So if $\phi(n)$ can be found, then d can be found as shown in an earlier example. Suppose $\phi(n)$ is known. Then, since n is public information, it is easy to find

$$p + q = pq - (p-1)(q-1) + 1 = n - \phi(n) + 1$$
$$p - q = \left[(p+q)^2 - 4pq\right]^{1/2} = \left[(p+q)^2 - 4n\right]^{1/2},$$

and, therefore,

$$p = \frac{1}{2}[(p+q) + (p-q)] \quad \text{and} \quad q = \frac{1}{2}[(p+q) - (p-q)].$$

Suppose, conversely, that d is known. Then a multiple of $\phi(n)$, namely $ed - 1$, can be found. Given a multiple of $\phi(n)$, it is possible to factor $n = pq$ (see [14]). Thus, finding $\phi(n)$ or d is computationally no easier than factoring n. The fastest factoring algorithm known [15] uses approximately $\exp(\sqrt{(\ln N)\ln(\ln N)})$ bit operations to factor an integer N. Assuming that each bit operation takes one microsecond (10^{-6} seconds), we obtain the following factorization times:

Integer N	Number of Bit Operations	Time
10^{50}	1.4×10^{10}	3.9 hours
10^{75}	9.0×10^{12}	104 days
10^{100}	2.3×10^{15}	74 years
10^{200}	1.2×10^{23}	3.8×10^{9} years
10^{300}	1.5×10^{29}	4.9×10^{15} years

Since the integer $n = pq$ has approximately 200 digits, it appears that it is computa-

tionally infeasible to find D, based only on knowing E. Thus, E is a one-way function and the trapdoor information is the prime factorization pq of n.

Is it possible to decipher a ciphertext message without finding the deciphering algorithm D, and hence avoiding the factorization of $n = pq$? In 1977, G. L. Simmons and M. J. Norris proposed a method of cryptanalyzing the RSA cryptosystem by using successive encryptions by E. They generated a sequence C_1, $C_2, \ldots$ by defining $C_1 = E(M)$ and $C_{j+1} = E(C_j)$, where M is the plaintext message. The sequence is terminated when an integer t is reached such that $C_t = C_1$. Then $C_{t-1} = M$, since $E(M) = E(C_{t-1})$. However, it has been shown [20] that the probability of this attack being successful is extremely low if the chosen primes p, q differ in length by only a few digits, $(p-1, q-1)$ is small, and both $p-1$ and $q-1$ have large prime factors. Other cryptanalytic approaches to the RSA cryptosystem have been suggested, many of which have been quickly followed by counterarguments.

Since the announcement of the RSA cryptosystem, no generally effective attacks have surfaced to threaten its security. Ironically, the research activity stimulated by this system has enhanced its security. In order to protect an RSA encryption cipher from attack, certain conditions must be satisfied:

(i) p and q should each be at least 100 digits
(ii) p and q should differ in length by a few digits
(iii) $(p-1, q-1)$ should be large
(iv) Each of the integers $p \pm 1$, $q \pm 1$ should have at least one large prime factor.

Property (i) ensures that n is extremely time consuming to factor by trial or by the Monte Carlo method. Properties (ii)–(iv), in conjunction with (i), guarantee that no special factorization techniques (such as the difference of squares, and the $p+1$ or $p-1$ method) or special decryption methods (such as successive encryption) can be effectively used. Since it is possible to construct a modulus satisfying all four conditions [28], we can build a secure RSA cryptosystem today.

It has not been proven that the problem of breaking the RSA cryptosystem is equivalent in difficulty to factoring the modulus $n = pq$. There could be a method that would decipher messages without identifying the trapdoor information. This appears unlikely however, since all known deciphering methods that work in the general case are equivalent to factoring n. Factorization is a computationally "hard" problem; that is, there is no general-purpose prime factorization algorithm that computes the factors of n in a number of steps bounded by a polynomial in n. As long as prime factorization continues to be a "hard" problem and no new cryptanalytic techniques develop, the RSA system appears secure; it will always be "easier" to find large primes than it will be to factor numbers of the same magnitude. It certainly is exciting that a conjecture by Fermat in 1640 plays such a pivotal role in cryptology today.

Trapdoor Knapsacks

In 1977, a public-key cipher was proposed, with its security depending on the difficulty of solving the classic knapsack problem [12]:

Given positive integers $a_1, a_2, \ldots, a_n$ *and a sum* S, *solve*

$$S = a_1x_1 + a_2x_2 + \cdots + a_nx_n$$

for the x_i*'s* $(1 \le i \le n)$, *where each* $x_i = 0$ *or* 1.

This is a notoriously "hard" problem [8]. An exhaustive search requires a check of the 2^n possibilities for $(x_1, x_2, \ldots, x_n)$. The best known method for finding a

solution of the knapsack problem requires approximately $2^{n/2}$ computational steps. So for $n = 100$, a reliable computer solution is infeasible. However, for certain sequences $a_1, a_2, \ldots, a_n$, the solution of the knapsack problem is quite "easy."

A sequence $\{a_1, a_2, a_3, \ldots, a_n\}$ is said to be *superincreasing* if it satisfies

$$\sum_{i=1}^{j-1} a_i < a_j \quad \text{for each } j = 2, 3, \ldots, n.$$

Assume we have such a sequence, and that the preselected sum S can be attained using some subset of the sequence. To solve $S = \sum_{i=1}^{n} a_i x_i$, we proceed as follows: let $x_n = 1$ if $S \geq a_n$, and $x_n = 0$ if $S < a_n$. Then find $x_{n-1}, x_{n-2}, \ldots, x_1$, in succession, using

$$x_j = \begin{cases} 1, & \text{if } S - \sum_{i=j+1}^{n} x_i a_i \geq a_j \\ 0, & \text{if } S - \sum_{i=j+1}^{n} x_i a_i < a_j \end{cases}$$

for $j = n-1, n-2, \ldots, 2, 1$. This can be solved rapidly on a computer, even for large n.

To build a public-key cryptosystem, assume that each user in the system selects a superincreasing sequence $\{a_1, a_2, \ldots, a_n\}$, an integer $m > 2a_n$, and an integer w which is relatively prime to m. Let $\bar{w}$ represent the multiplicative inverse modulo m of w (this can be easily found using the Euclidean Algorithm). Now form the not necessarily superincreasing sequence $\{b_1, b_2, \ldots, b_n\}$ by

$$b_i \equiv wa_i \pmod{m}$$

with $0 < b_i < m$ for $i = 1, 2, \ldots, n$.

The public-key information is the sequence $\{b_1, b_2, \ldots, b_n\}$, while the trapdoor information is m and w. Suppose someone wants to send this user a message M. The message is first translated into a string of 0's and 1's, using the binary equivalent to letters:

Letter	Binary Equivalent	Letter	Binary Equivalent
A	00000	N	01101
B	00001	O	01110
C	00010	P	01111
D	00011	Q	10000
E	00100	R	10001
F	00101	S	10010
G	00110	T	10011
H	00111	U	10100
I	01000	V	10101
J	01001	W	10110
K	01010	X	10111
L	01011	Y	11000
M	01100	Z	11001

This string of 0's and 1's is then split into blocks of length n (if the length $5|M|$ of a string is not divisible by n, fill in the last block with 1's). Each plaintext block $X_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ $(i = 1, 2, \ldots, k)$ is enciphered as

$$S_i = \sum_{j=1}^{n} b_j x_{ij}.$$

[Note that S_i can be interpreted as the dot product $\vec{b} \cdot \vec{X}_i$ for $\vec{b} = \{b_1, b_2, \ldots, b_n\}$.] The sums $\{S_1, S_2, \ldots, S_k\}$ corresponding to these enciphered blocks form the ciphertext message. There is no concern about interception of the ciphertext since deciphering the text requires solving a group of "hard" knapsack problems of the form $S_i = \sum_{j=1}^{n} b_j x_{ij}$. However, when m and w are known (the private trapdoor information held by the user), these "hard" knapsack problems can be transformed into "easy" knapsack problems without changing the solution sets. We illustrate this for a single block. The transformation is as follows. For $S_i = \sum_{j=1}^{n} b_j x_{ij}$:

$$\overline{w}S_i = \sum_{j=1}^{n} \overline{w} b_j x_{ij} \equiv \left(\sum_{j=1}^{n} a_j x_{ij} \right) (\text{mod } m).$$

Note that $m > 2a_n$ (by definition) and $2a_n > \sum_{j=1}^{n} a_j$ (since $\{a_1, a_2, \ldots, a_n\}$ is superincreasing) yields $\sum_{j=1}^{n} a_j x_{ij} < m$. If $\tilde{S}_i \equiv \overline{w} S_i (\text{mod } m)$ with $0 \leq \tilde{S}_i < m$, then $\tilde{S}_i = \sum_{j=1}^{n} a_j x_{ij}$. Solving these easy knapsack problems, $\tilde{S}_i = \sum_{j=1}^{n} a_j x_{ij}$, yields solutions to the hard knapsack problems $S_i = \sum_{j=1}^{n} b_j x_{ij}$, and consequently, the original plaintext message M.

Example of a knapsack cipher. Suppose your private-key is the superincreasing sequence {11, 15, 30, 60} and you've selected numbers $m = 150$ and $w = 77$. Then your public-key sequence, defined by

$$b_j \equiv 77 a_j \ (\text{mod } 150) \quad (j = 1, 2, 3, 4),$$

is {97, 105, 60, 120}. To encipher the message

DONALD DUCK

first translate each letter into its binary equivalents

D	O	N	A	L	D	D	U	C	K,
00011	01110	01101	00000	01011	00011	00011	10100	00010	01010

and then partition this sequence of 0's and 1's into blocks of length four:

0001 1011 1001 1010 0000 0101 1000 1100 0111 0100 0001 0010 1011.

For each block X_i, form the corresponding dot product sum $S_i = \vec{b} \cdot \vec{X}_i$ using $\vec{b} = \{97, 105, 60, 120\}$:

$$\begin{aligned}
S_1 &= 0(97) + 0(105) + 0(60) + 1(120) = 120 \\
S_2 &= 1(97) + 0(105) + 1(60) + 1(120) = 277 \\
S_3 &= 1(97) + 0(105) + 0(60) + 1(120) = 217 \\
S_4 &= 1(97) + 0(105) + 1(60) + 0(120) = 157 \\
S_5 &= 0(97) + 0(105) + 0(60) + 0(120) = \quad 0 \\
&\vdots \\
S_{13} &= 1(97) + 0(105) + 1(60) + 1(120) = 277.
\end{aligned}$$

The ciphertext is then 120, 277, 217, 157, 0, ..., 277. To decipher, find the multiplicative inverse (modulo 150) of $w = 77$. Solving $77\overline{w} \equiv 1 (\text{mod } 150)$ yields $\overline{w} = 113$. Then multiply each S_i by $\overline{w}$. Thus, for example, $S_1 = 120$ is deciphered to 60 since

$$113(120) \equiv 60 \ (\text{mod } 150).$$

The remaining sums become 101, 71, 41, 0, ..., 101, respectively. Each of these transformed sums $\tilde{S}_i$ corresponds to an "easy" knapsack problem relative to the

superincreasing sequence {11, 15, 30, 60}. Solving the knapsack problems

$$\tilde{S}_i = 11x_1 + 15x_2 + 30x_3 + 60x_4$$

for each sum yields

$$0001, 1011, 1001, 1010, 0000, \ldots, 1011.$$

Partitioning the string 00011011100110100000...1011 into blocks of length five identifies the message:

00011	01110	01101	00000	...	01010	.
D	O	N	A	...	K	

A cryptosystem based on the public-key knapsack cipher does not satisfy the four conditions in the definition of the public-key cryptosystem. The enciphering algorithm is clearly not onto, since only certain sums can be generated by the integers $b_1, b_2, \ldots, b_n$. It can be shown, however, that all of the conditions, except this one, are satisfied. Without $E(D(C)) = C$ holding in general, it is impossible to use the signature scheme that was described earlier. For a discussion of knapsack-based algorithms and signatures, see [23].

Knapsack cryptosystems have received their share of attention since 1978. They were initially favored over the RSA system since encryption-decryption was faster. Special integrated circuit chips to implement a knapsack cryptosystem have even been considered by a few companies. Recently, this attention has diminished significantly. In April 1982, a fundamental cryptanalytic breakthrough was made when a polynomial time attack on "almost all" (Merkle-Hellman) knapsack cryptosystems was carried out [24]. The attack, by A. Shamir, demonstrated that certain information about superincreasing sequences is not well hidden by modular multiplication. Using only the sequence $\{b_1, b_2, \ldots, b_n\}$ and ideas from Diophantine approximation, it is possible to find enough secret information to solve any knapsack problem involving the weights $b_1, b_2, \ldots, b_n$. Several knapsack cryptosystems have been proposed and broken. For example, the Graham-Shamir scheme was subsequently broken by Adleman, using a polynomial time algorithm [1].

Meanwhile, Shamir collected a prize of $100 from Merkle for breaking his basic scheme. Merkle, distressed by misleading media reports of this special case solution, offered a $1,000 prize to anyone who could break the multiply-iterated version of his basic scheme [12]. Several attacks followed. In the fall of 1984, Ernest Brickell, of Sandia Labs, collected the $1,000 prize. Brickell's technique depends on the fact that modular multiplication is the only method used to keep the private key hidden. He found a way to use a single function to change the multiple iterations into an easy knapsack problem. The technique identifies the plaintext message without finding the original superincreasing sequence. Knapsack ciphers with 100 weights and 20 iterations can be broken using less than 2 hours of computer time on a Cray 1S computer. This technique is expected to break a 1000-weights, 40-iterations cipher in 750 hours on the same computer [3].

Does all this rule out the possibility of the existence of a secure knapsack cryptosystem? Not necessarily. But certainly any new knapsack systems must find other ways, besides modular arithmetic, to hide private information. There is already a new knapsack cipher, proposed by B. Chor and R. Rivest, based on arithmetic in finite fields. It attempts to avoid the flaws identified by successful attacks on prior knapsack systems, and only time will determine its success or failure. It is interesting to note, in light of Edgar Allen Poe's quote, that when Chor-Rivest proposed their system they remarked, "At the moment, we do not know of any attacks capable of breaking this system in a reasonable amount of time."

REFERENCES

1. L. M. Adleman, "On Breaking the Generalized Knapsack Public-Key Cryptosystem," Proceedings 15th Annual ACM Symposium on Theory of Computing (1983) 402–412.
2. L. M. Adleman, G. Pomerance, and R. S. Rumely, "On Distinguishing Prime Numbers from Composite Numbers," Annals of Mathematics 117 (1983) 173–206.
3. E. Brickell, "Breaking Iterated Knapsacks," Lecture Notes in CS, 196, Advances in Cryptology: Proceedings of CRYPTO 84, Springer-Verlag, New York, 1985, pp. 342–357.
4. W. Diffie and M. E. Hellman, "New Directions in Cryptography," IEEE Transactions of Information Theory IT-22 (November 1976) 644–654.
5. J. Dixon, "Factorization and Primality Tests," American Mathematical Monthly 91 (1984) 333–352.
6. W. Francis, "A Standard Sample of Present-Day Edited American English for Use with Digital Computers," Linguistics Department, Brown University, 1964.
7. M. Gardner, "A New Kind of Cipher that Would Take Millions of Years to Break," Scientific American 237 (1977) 120–124.
8. M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, San Francisco, 1979.
9. A. Hodges, *Alan Turing: The Enigma*, Simon and Schuster, New York, 1983.
10. D. Kahn, *The Codebreakers, the Story of Secret Writing*, MacMillan, New York, 1967.
11. A. G. Konheim, *Cryptography: A Primer*, Wiley-Interscience, New York, 1981.
12. R. Merkle and M. Hellman, "Hiding Information and Signatures in Trapdoor Knapsacks," IEEE Transactions of Information Theory IT-24, No. 5 (September 1978) 525–530.
13. C. H. Meyer and S. M. Matyas, *Cryptography: A New Dimension in Computer Data Security*, Wiley-Interscience, New York, 1982.
14. G. L. Miller, "Riemann's Hypothesis and Tests for Primality," Journal of Computer and System Sciences 13, No. 3 (December 1976) 300–317.
15. C. Pomerance, "Analysis and Comparison of Some Integer Factoring Algorithms," Computational Methods in Number Theory, Math Centrum Tracts 154, Part I (1982) 89–139.
16. _____, "The Search for Prime Numbers," Scientific American 247 (December 1982) 122–130.
17. _____, *Lecture Notes on Primality Testing and Factoring*, MAA Notes, No. 4, Mathematical Association of America, Washington, D.C., 1984.
18. M. O. Rabin, "Probabilistic Algorithm for Testing Primality," Journal of Number Theory 12 (1980) 128–138.
19. H. Riesel, *Prime Numbers and Computer Methods for Factorization*, Progress in Mathematics 57, Birkhäuser, Boston, Massachusetts, 1985.
20. R. L. Rivest, "Remarks on a Proposed Cryptanalytic Attack on the M.I.T. Public-Key Cryptosystem," Cryptologia 2 (1978) 62–65.
21. R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Laboratory for Computer Science, M.I.T., LCS/TM-82 (April 1977).
22. _____, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM 21 : 2 (1978) 120–126.
23. A. Shamir, "The Cryptanalytic Security of Compact Knapsacks," M.I.T., LCS/TM-164 (1980).
24. _____, "A Polynomial Time Algorithm Testing for Breaking the Basic Merkle-Hellman Cryptosystems," Proceedings of the 23rd Annual Symposium of the Foundations of Computer Science (1982) 145–152.
25. C. E. Shannon, "Communications Theory of Secrecy Systems," Bell Sys. Tech. Journal 28 (1949) 656–715.
26. A. Sinkov, *Elementary Cryptanalysis—A Mathematical Approach*, New Mathematical Library, No. 22, Mathematical Association of America, 1966.
27. R. Solovay and V. Strassen, "A Fast Monte-Carlo Test for Primality," SIAM Journal on Computing 6 (March 1977) 84–85.
28. R. C. Williams and B. Schmid, "Some Remarks Concerning the M.I.T. Public-Key Cryptosystems," Science Report No. 91 (1979) Department of Computer Science, University of Manitoba, Winnipeg, Canada.