6. From $23^2 \equiv 17 \pmod{2^7}$, find three other solutions of the quadratic congruence $x^2 \equiv 17 \pmod{2^7}$.
7. First determine the values of a for which the congruences below are solvable, and then find the solutions of these congruences:
   (a) $x^2 \equiv a \pmod{2^4}$.
   (b) $x^2 \equiv a \pmod{2^5}$.
   (c) $x^2 \equiv a \pmod{2^6}$.
8. For fixed $n > 1$, show that all the solvable congruences $x^2 \equiv a \pmod{n}$ with $\gcd(a, n) = 1$ have the same number of solutions.
9. (a) Without actually finding them, determine the number of solutions of the congruences $x^2 \equiv 3 \pmod{11^2 \cdot 23^2}$ and $x^2 \equiv 9 \pmod{2^3 \cdot 3 \cdot 5^2}$.
   (b) Solve the congruence $x^2 \equiv 9 \pmod{2^3 \cdot 3 \cdot 5^2}$.
10. (a) For an odd prime p, prove that the congruence $2x^2 + 1 \equiv 0 \pmod{p}$ has a solution if and only if $p \equiv 1$ or $3 \pmod 8$.
    (b) Solve the congruence $2x^2 + 1 \equiv 0 \pmod{11^2}$.
    [*Hint:* Consider integers of the form $x_0 + 11k$, where x_0 is a solution of $2x^2 + 1 \equiv 0 \pmod{11}$.]

CHAPTER 10

INTRODUCTION TO CRYPTOGRAPHY

I am fairly familiar with all forms of secret writings and am myself the author of a trifling manuscript on the subject.
SIR ARTHUR CONAN DOYLE

10.1 FROM CAESAR CIPHER TO PUBLIC KEY CRYPTOGRAPHY

Classically, the making and breaking of secret codes has usually been confined to diplomatic and military practices. With the growing quantity of digital data stored and communicated by electronic data-processing systems, organizations in both the public and commercial sectors have felt the need to protect information from unwanted intrusion. Indeed, the widespread use of electronic funds transfers has made privacy a pressing concern in most financial transactions. There thus has been a recent surge of interest by mathematicians and computer scientists in *cryptography* (from the Greek *kryptos* meaning *hidden* and *graphein* meaning *to write*), the science of making communications unintelligible to all except authorized parties. Cryptography is the only known practical means for protecting information transmitted through public communications networks, such as those using telephone lines, microwaves, or satellites.

In the language of cryptography, where codes are called *ciphers*, the information to be concealed is called *plaintext*. After transformation to a secret form, a message is called *ciphertext*. The process of converting from plaintext to ciphertext is said to be *encrypting* (or *enciphering*), whereas the reverse process of changing from ciphertext back to plaintext is called *decrypting* (or *deciphering*).

One of the earliest cryptographic systems was used by the great Roman emperor Julius Caesar around 50 B.C. Caesar wrote to Marcus Cicero using a rudimentary substitution cipher in which each letter of the alphabet is replaced by the letter that occurs three places down the alphabet, with the last three letters cycled back to the first three letters. If we write the ciphertext equivalent underneath the plaintext letter, the substitution alphabet for the *Caesar cipher* is given by

Plaintext: 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

Ciphertext: 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

For example, the plaintext message

CAESAR WAS GREAT

is transformed into the ciphertext

FDHVDU ZDV JUHDW

The Caesar cipher can be described easily using congruence theory. Any plaintext is first expressed numerically by translating the characters of the text into digits by means of some correspondence such as the following:

A	B	C	D	E	F	G	H	I	J	K	L	M
00	01	02	03	04	05	06	07	08	09	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

If P is the digital equivalent of a plaintext letter and C is the digital equivalent of the corresponding ciphertext letter, then

$$C \equiv P + 3 \pmod{26}$$

Thus, for instance, the letters of the message in Eq. (1) are converted to their equivalents:

02 00 04 18 00 17 22 00 18 06 17 04 00 19

Using the congruence $C \equiv P + 3 \pmod{26}$, this becomes the ciphertext

05 03 07 21 03 20 25 03 21 09 20 07 03 22

To recover the plaintext, the procedure is simply reversed by means of the congruence

$$P \equiv C - 3 \equiv C + 23 \pmod{26}$$

The Caesar cipher is very simple and, hence, extremely insecure. Caesar himself soon abandoned this scheme—not only because of its insecurity, but also because he did not trust Cicero, with whom he necessarily shared the secret of the cipher.

An encryption scheme in which each letter of the original message is replaced by the same cipher substitute is known as a *monoalphabetic cipher*. Such cryptographic systems are extremely vulnerable to statistical methods of attack because they preserve the frequency, or relative commonness, of individual letters. In a *polyalphabetic cipher,* a plaintext letter has more than one ciphertext equivalent: the

General fascination with cryptography had its initial impetus with the short story *The Gold Bug*, published in 1843 by the American writer Edgar Allan Poe. It is a fictional tale of the use of a table of letter frequencies to decipher directions for finding Captain Kidd's buried treasure. Poe fancied himself a cryptologist far beyond the ordinary. Writing for *Alexander's Weekly*, a Philadelphia newspaper, he once issued a claim that he could solve "forthwith" any monoalphabetic substitution cipher sent in by readers. The challenge was taken up by one G. W. Kulp, who submitted a 43-word ciphertext in longhand. Poe showed in a subsequent column that the entry was not genuine, but rather a "jargon of random characters having no meaning whatsoever." When Kulp's cipher submission was finally decoded in 1975, the reason for the difficulty became clear; the submission contained a major error on Kulp's part, along with 15 minor errors, which were most likely printer's mistakes in reading Kulp's longhand.

The most famous example of a polyalphabetic cipher was published by the French cryptographer Blaise de Vigenère (1523–1596) in his *Traicté de Chiffres* of 1586. To implement this system, the communicating parties agree on an easily remembered word or phrase. With the standard alphabet numbered from A = 00 to Z = 25, the digital equivalent of the keyword is repeated as many times as necessary beneath that of the plaintext message. The message is then enciphered by adding, modulo 26, each plaintext number to the one immediately beneath it. The process may be illustrated with the keyword READY, whose numerical version is 17 04 00 03 24. Repetitions of this sequence are arranged below the numerical plaintext of the message

ATTACK AT ONCE

to produce the array

00	19	19	00	02	10	00	19	14	13	02	04
17	04	00	03	24	17	04	00	03	24	17	04

When the columns are added modulo 26, the plaintext message is encrypted as

17 23 19 03 00 01 04 19 17 11 19 08

or, converted to letters,

RXTDAB ET RLTI

Notice that a given letter of plaintext is represented by different letters in ciphertext. The double T in the word ATTACK no longer appears as a double letter when ciphered, while the ciphertext letter R first corresponds to A and then to O in the original message.

In general, any sequence of n letters with numerical equivalents $b_1, b_2, \ldots, b_n$ $(00 \leq b_i \leq 25)$ will serve as the keyword. The plaintext message is expressed as successive blocks $P_1 P_2 \cdots P_n$ of n two-digit integers P_i, and then converted to ciphertext blocks $C_1 C_2 \cdots C_n$ by means of the congruences

$$C_i \equiv P_i + b_i \pmod{26} \qquad 1 \leq i \leq n$$

Decryption is carried out by using the relations

A weakness in Vigenère's approach is that once the length of the keyword has been determined, a coded message can be regarded as a number of separate monoalphabetic ciphers, each subject to straightforward frequency analysis. A variant to the continued repetition of the keyword is what is called a *running key*, a random assignment of ciphertext letters to plaintext letters. A favorite procedure for generating such keys is to use the text of a book, where both sender and recipient know the title of the book and the starting point of the appropriate lines. Because a running key cipher completely obscures the underlying structure of the original message, the system was long thought to be secure. But it does not, as *Scientific American* once claimed, produce ciphertext that is "impossible of translation."

A clever modification that Vigenère contrived for his polyalphabetic cipher is currently called the *autokey* ("automatic key"). This approach makes use of the plaintext message itself in constructing the encryption key. The idea is to start off the keyword with a short *seed* or *primer* (generally a single letter) followed by the plaintext, whose ending is truncated by the length of the seed. The autokey cipher enjoyed considerable popularity in the 16th and 17th centuries, since all it required of a legitimate pair of users was to remember the seed, which could easily be changed.

Let us give a simple example of the method.

Example 10.1. Assume that the message

ONE IF BY DAWN

is to be encrypted. Taking the letter K as the seed, the keyword becomes

KONEIFBYDAW

When both the plaintext and keyword are converted to numerical form, we obtain the array

14	13	04	08	05	01	24	03	00	22	13
10	14	13	04	08	05	01	24	03	00	22

Adding the integers in matching positions modulo 26 yields the ciphertext

24	01	17	12	13	06	25	01	03	22	09

or, changing back to letters:

YBR MN GZ BDWJ

Decipherment is achieved by returning to the numerical form of both the plaintext and its ciphertext. Suppose that the plaintext has digital equivalents $P_1P_2 \ldots P_n$ and the ciphertext $C_1C_2 \ldots C_n$. If S indicates the seed, then the first plaintext number is

$$P_1 = C_1 - S = 24 - 10 \equiv 14 \pmod{26}$$

Thus, the deciphering transformation becomes

$$P_k = C_k - P_{k-1} \pmod{26}, 2 \leq k \leq n$$

This recovers, for example, the integers

$$P_2 \equiv 01 - 14 = -13 \equiv 13 \pmod{26}$$

$$P_3 \equiv 17 - 13 \equiv 4 \pmod{26}$$

where, to maintain the two-digit format, the 4 is written 04.

A way to ensure greater security in alphabetic substitution ciphers was devised in 1929 by Lester Hill, an assistant professor of mathematics at Hunter College. Briefly, Hill's approach is to divide the plaintext message into blocks of n letters (possibly filling out the last block by adding "dummy" letters such as X's), and then to encrypt block by block using a system of n linear congruences in n variables. In its simplest form, when $n = 2$, the procedure takes two successive letters and transforms their numerical equivalents P_1P_2 into a block C_1C_2 of ciphertext numbers via the pair of congruences

$$C_1 \equiv aP_1 + bP_2 \pmod{26}$$

$$C_2 \equiv cP_1 + dP_2 \pmod{26}$$

To permit decipherment, the four coefficients a, b, c, d must be selected so the $\gcd(ad - bc, 26) = 1$.

Example 10.2. To illustrate Hill's cipher, let us use the congruences

$$C_1 \equiv 2P_1 + 3P_2 \pmod{26}$$

$$C_2 \equiv 5P_1 + 8P_2 \pmod{26}$$

to encrypt the message BUY NOW. The first block BU of two letters is numerically equivalent to 01 20. This is replaced by

$$2(01) + 3(20) \equiv 62 \equiv 10 \pmod{26}$$

$$5(01) + 8(20) \equiv 165 \equiv 09 \pmod{26}$$

Continuing two letters at a time, we find that the completed ciphertext is

10 09 09 16 16 12

which can be expressed alphabetically as KJJ QQM.

Decipherment requires solving the original system of congruences for P_1 and P_2 in terms of C_1 and C_2. It follows from the proof of Theorem 4.9 that the plaintext block P_1P_2 can be recovered from the ciphertext block C_1C_2 by means of the congruences

$$P_1 \equiv 8C_1 - 3C_2 \pmod{26}$$

$$P_2 \equiv -5C_1 + 2C_2 \pmod{26}$$

For the block 10 09 of ciphertext, we calculate

$$P_1 \equiv 8(10) - 3(09) \equiv 53 \equiv 01 \pmod{26}$$

$$P_2 \equiv -5(10) + 2(09) \equiv -32 \equiv 20 \pmod{26}$$

which is the same as the letter-pair BU. The remaining plaintext can be restored in a similar manner.

An influential nonalphabetic cipher was devised by Gilbert S. Verman in 1917 while he was employed by the American Telephone and Telegraph Company (AT&T). Verman was interested in safeguarding information sent by the newly developed teletypewriter. At that time, wire messages were transmitted in the Baudot code, a code named after its French inventor J. M. E. Baudot. Baudot represented each letter of the alphabet by a five-element sequence of two symbols. If we take the two symbols to be 1 and 0, then the complete table is given by

A = 11000	J = 11010	S = 10100
B = 10011	K = 11110	T = 00001
C = 01110	L = 01001	U = 11100
D = 10010	M = 00111	V = 01111
E = 10000	N = 00110	W = 11001
F = 10110	O = 00011	X = 10111
G = 01011	P = 01101	Y = 10101
H = 00101	Q = 11101	Z = 10001
I = 01100	R = 01010	

Any plaintext message such as

ACT NOW

would first be transformed into a sequence of binary digits:

11000011100000100110000111100 1

Verman's innovation was to take as the encryption key an arbitrary sequence of 1's and 0's with length the same as that of the numerical plaintext. A typical key might appear as

10100101110010001000111100 1011

where the digits could be chosen by flipping a coin with heads as 1 and tails as 0. Finally, the ciphertext is formed by adding modulo 2 the digits in equivalent places in the two binary strings. The result in this instance becomes

01100110010010101110111111 0010

A crucial point is that the intended recipient must possess in advance the encryption key, for then the numerical plaintext can be reconstructed by merely adding modulo 2 corresponding digits of the encryption key and ciphertext.

In the early applications of Verman's telegraph cipher, the keys were written on numbered sheets of paper and then bound into pads held by both correspondents. A sheet was torn out and destroyed after its key had been used just once. For this reason, the Verman enciphering procedure soon became known as the one-time system or one-time pad. The cryptographic strength of Verman's method of enciphering resided in the possibly extreme length of the encryption key and the absence of any pattern within its entries. This assured security that was attractive to the military or diplomatic services of many countries. In 1963, for instance, a teleprinter hot line was established between Washington and Moscow using a one-time tape.

In conventional cryptographic systems, such as Caesar's cipher, the sender and receiver jointly have a secret *key*. The sender uses the key to encrypt the plaintext to be sent, and the receiver uses the same key to decrypt the ciphertext obtained. Public-key cryptography differs from conventional cryptography in that it uses two keys, an encryption key and a decryption key. Although the two keys effect inverse operations and are therefore related, there is no easily computed method of deriving the decryption key from the encryption key. Thus, the encryption key can be made public without compromising the decryption key; each user can encrypt messages, but only the intended recipient (whose decryption key is kept secret) can decipher them. A major advantage of a public-key cryptosystem is that it is unnecessary for senders and receivers to exchange a key in advance of their decision to communicate with each other.

In 1977, R. Rivest, A. Shamir, and L. Adleman proposed a public-key cryptosystem that uses only elementary ideas from number theory. Their enciphering system is called *RSA*, after the initials of the algorithm's inventors. Its security depends on the assumption that in the current state of computer technology, the factorization of composite numbers with large prime factors is prohibitively time-consuming.

Each user of the RSA system chooses a pair of distinct primes, p and q, large enough that the factorization of their product $n = pq$, called the *enciphering modulus*, is beyond all current computational capabilities. For instance, one might pick p and q with 200 digits each, so that n has roughly 400 digits. Having selected n, the user then chooses a random positive integer k, the *enciphering exponent*, satisfying $\gcd(k, \phi(n)) = 1$. The pair (n, k) is placed in a public file, analogous to a telephone directory, as the user's personal encryption key. This allows anyone else in the communication network to encrypt and send a message to that individual. Notice that whereas n is openly revealed, the listed public key does not mention the factors p and q of n.

The encryption process begins with the conversion of the message to be sent into an integer M by means of a "digital alphabet" in which each letter, number, or punctuation mark of the plaintext is replaced by a two-digit integer. One standard procedure is to use the following assignment:

A = 00	K = 10	U = 20	1 = 30
B = 01	L = 11	V = 21	2 = 31
C = 02	M = 12	W = 22	3 = 32
D = 03	N = 13	X = 23	4 = 33
E = 04	O = 14	Y = 24	5 = 34
F = 05	P = 15	Z = 25	6 = 35
G = 06	Q = 16	, = 26	7 = 36
H = 07	R = 17	. = 27	8 = 37
I = 08	S = 18	? = 28	9 = 38
J = 09	T = 19	0 = 29	! = 39

with 99 indicating a space between words. In this scheme, the message

The brown fox is quick

is transformed into the numerical string

$$M = 19070499011714221399051423990818991620080210 27$$

It is assumed that the plaintext number $M < n$, where n is the enciphering modulus. Otherwise it would be impossible to distinguish M from any larger integer congruent to it modulo n. When the message is too long to be handled as a single number $M < n$, then M is broken up into blocks of digits $M_1, M_2, \ldots, M_s$ of the appropriate size. Each block is encrypted separately.

Looking up the intended recipient's encryption key (n, k) in the public directory, the sender disguises the plaintext number M as a ciphertext number r by raising M to the kth power and then reducing the result modulo n; that is,

$$M^k \equiv r \pmod{n}$$

A 200-character message can be encrypted in seconds on a high-speed computer. Recall that the public enciphering exponent k was originally selected so that $\gcd(k, \phi(n)) = 1$. Although there are many suitable choices for k, an obvious suggestion is to pick k to be any prime larger than both p and q.

At the other end, the authorized recipient deciphers the transmitted information by first determining the integer j, the secret *recovery exponent*, for which

$$kj \equiv 1 \pmod{\phi(n)}$$

Because $\gcd(k, \phi(n)) = 1$, this linear congruence has a unique solution modulo $\phi(n)$. In fact, the Euclidean algorithm produces j as a solution x to the equation

$$kx + \phi(n)y = 1$$

The recovery exponent can only be calculated by someone who knows both k and $\phi(n) = (p-1)(q-1)$ and, hence, knows the prime factors p and q of n. Thus, j is secure from an illegitimate third party whose knowledge is limited to the public key (n, k).

Matters have been arranged so that the recipient can now retrieve M from r by simply calculating r^j modulo n. Because $kj = 1 + \phi(n)t$ for some integer t, it follows that

$$\begin{aligned} r^j \equiv (M^k)^j &\equiv M^{1+\phi(n)t} \\ &\equiv M(M^{\phi(n)})^t \equiv M \cdot 1^t \equiv M \pmod{n} \end{aligned}$$

whenever $\gcd(M, n) = 1$. In other words, raising the ciphertext number to the jth power and reducing it modulo n recovers the original plaintext number M.

The assumption that $\gcd(M, n) = 1$ was made to use Euler's theorem. In the unlikely event that M and n are not relatively prime, a similar argument establishes that $r^j \equiv M \pmod{p}$ and $r^j \equiv M \pmod{q}$, which then yields the desired congruence $r^j \equiv M \pmod{n}$. We omit the details.

The major advantage of this ingenious procedure is that the encryption of a message does not require the knowledge of the two primes p and q, but only their product n; there is no need for anyone other than the receiver of the message ever to know the prime factors critical to the decryption process.

Example 10.3. For the reader to gain familiarity with the RSA public-key algorithm, let us work an example in detail. We first select two primes

$$p = 29 \qquad q = 53$$

of an unrealistically small size, to get an easy-to-handle illustration. In practice, p and q would be large enough so that the factorization of the nonsecret $n = pq$ is not feasible. Our enciphering modulus is $n = 29 \cdot 53 = 1537$ and $\phi(n) = 28 \cdot 52 = 1456$. Because $\gcd(47, 1456) = 1$, we may choose $k = 47$ to be the enciphering exponent. Then the recovery exponent, the unique integer j satisfying the congruence $kj \equiv 1 \pmod{\phi(n)}$, is $j = 31$. To encrypt the message

NO WAY

first translate each letter into its digital equivalent using the substitution mentioned earlier; this yields the plaintext number

$$M = 131499220024$$

We want each plaintext block to be an integer less than 1537. Given this restriction, it seems reasonable to split M into blocks of three digits each. The first block, 131, encrypts as the ciphertext number

$$131^{47} \equiv 570 \pmod{1537}$$

These are the first digits of the secret transmission. At the other end, knowing that the recovery exponent is $j = 31$, the authorized recipient begins to recover the plaintext number by computing

$$570^{31} \equiv 131 \pmod{1537}$$

The total ciphertext of our message is

0570 1222 0708 1341

For the RSA cryptosystem to be secure it must not be computationally feasible to recover the plaintext M from the information assumed to be known to a third party, namely, the listed public-key (n, k). The direct method of attack would be to attempt to factor n, an integer of huge magnitude; for once the factors are determined, the recovery exponent j can be calculated from $\phi(n) = (p-1)(q-1)$ and k. Our confidence in the RSA system rests on what is known as the work factor, the expected amount of computer time needed to factor the product of two large primes. Factoring is computationally more difficult than distinguishing between primes and composites. On today's fastest computers, a 200-digit number can routinely be tested for primality in less than 20 seconds, whereas the running time required to factor a composite number of the same size is prohibitive. It has been estimated that the quickest factoring algorithm known can use approximately $(1.2)10^{23}$ computer operations to resolve an integer with 200 digits into its prime factors; assuming that each operation takes 1 nanosecond (10^{-9} seconds), the factorization time would be about $(3.8)10^6$ years. Given unlimited computing time and some unimaginably efficient factoring algorithm, the RSA cryptosystem could be broken, but for the present it

appears to be quite safe. All we need do is choose larger primes p and q for the enciphering moduli, always staying ahead of the current state of the art in factoring integers.

A greater threat is posed by the use of widely distributed networks of computers, working simultaneously on pieces of data necessary for a factorization and communicating their results to a central site. This is seen in the factoring of RSA-129, one of the most famous problems in cryptography.

To demonstrate that their cryptosystem could withstand any attack on its security, the three inventors submitted a ciphertext message to *Scientific American*, with an offer of $100 to anyone who could decode it. The message depended on a 129-digit enciphering modulus that was the product of two primes of approximately the same length. This large number acquired the name RSA-129. Taking into account the most powerful factoring methods and fastest computers available at the time, it was estimated that at least 40 quadrillion years would be required to break down RSA-129 and decipher the message. However, by devoting enough computing power to the task the factorization was realized in 1994. A worldwide network of some 600 volunteers participated in the project, running more than 1600 computers over an 8-month period. What seemed utterly beyond reach in 1977 was accomplished a mere 17 years later. The plaintext message is the sentence

"The magic words are squeamish ossifrage."

(An ossifrage, by the way, is a kind of hawk.)

Drawn up in 1991, the 42 numbers in the RSA Challenge List serve as something of a test for recent advances in factorization methods. The latest factoring success showed that the 174-digit number (576 binary digits) RSA-576 could be written as the product of two primes having 87 digits each.

PROBLEMS 10.1

1. Encrypt the message *RETURN HOME* using the Caesar cipher.
2. If the Caesar cipher produced *KDSSB ELUWKGDB*, what is the plaintext message?
3. (a) A linear cipher is defined by the congruence $C \equiv aP + b \pmod{26}$, where a and b are integers with $\gcd(a, 26) = 1$. Show that the corresponding decrypting congruence is $P \equiv a'(C - b) \pmod{26}$, where the integer a' satisfies $aa' \equiv 1 \pmod{26}$.
   (b) Using the linear cipher $C \equiv 5P + 11 \pmod{26}$, encrypt the message *NUMBER THEORY IS EASY*.
   (c) Decrypt the message *RXQTGU HOZTKGH FJ KTMMTG*, which was produced using the linear cipher $C \equiv 3P + 7 \pmod{26}$.
4. In a lengthy ciphertext message, sent using a linear cipher $C \equiv aP + b \pmod{26}$, the most frequently occurring letter is Q and the second most frequent is J.
   (a) Break the cipher by determining the values of a and b.
   [*Hint:* The most often used letter in English text is E, followed by T.]
   (b) Write out the plaintext for the intercepted message *WCPQ JZQO MX*.
5. (a) Encipher the message *HAVE A NICE TRIP* using a Vigenère cipher with the keyword *MATH*.
   (b) The ciphertext *BS FMX KFSGR JAPWL* is known to have resulted from a Vigenère cipher whose keyword is *YES*. Obtain the deciphering congruences and read the message.

6. (a) Encipher the message HAPPY DAYS ARE HERE using the autokey cipher with seed Q.
   (b) Decipher the message BBOT XWBZ AWUVGK, which was produced by the autokey cipher with seed RX.
7. (a) Use the Hill cipher

$$C_1 \equiv 5P_1 + 2P_2 \pmod{26}$$

$$C_2 \equiv 3P_1 + 4P_2 \pmod{26}$$

   to encipher the message *GIVE THEM TIME*.
   (b) The ciphertext *ALXWU VADCOJO* has been enciphered with the cipher

$$C_1 \equiv 4P_1 + 11P_2 \pmod{26}$$

$$C_2 \equiv 3P_1 + 8P_2 \pmod{26}$$

   Derive the plaintext.
8. A long string of ciphertext resulting from a Hill cipher

$$C_1 \equiv aP_1 + bP_2 \pmod{26}$$

$$C_2 \equiv cP_1 + dP_2 \pmod{26}$$

   revealed that the most frequently occurring two-letter blocks were *HO* and *PP*, in that order.
   (a) Find the values of a, b, c, and d.
   [*Hint:* The most common two-letter blocks in the English language are *TH*, followed by *HE*.]
   (b) What is the plaintext for the intercepted message *PPIH HOG RAPVT?*
9. Suppose that the message GO SOX is to be enciphered using Verman's telegraph cipher.
   (a) Express the message in Baudot code.
   (b) If the enciphering key is

01110101111010101001100 10

   obtain the alphabetic form of the ciphertext.
10. A plaintext message expressed in Baudot code has been converted by the Verman cipher into the string

11000111000011101010010111111 1

   If it is known that the key used for encipherment was

01110101100101111000100 1101010

   recover the message in its alphabetic form.
11. If $n = pq = 274279$ and $\phi(n) = 272376$, find the primes p and q.
   [*Hint:* Note that

$$p + q = n - \phi(n) + 1$$

$$p - q = [(p + q)^2 - 4n]^{1/2}.]$$

12. When the RSA algorithm is based on the key $(n, k) = (3233, 37)$, what is the recovery exponent for the cryptosystem?
13. Encrypt the plaintext message *GOLD MEDAL* using the RSA algorithm with key $(n, k) = (2419, 3)$.

14. The ciphertext message produced by the RSA algorithim with key $(n, k) = (1643, 223)$ is

0833 0823 1130 0055 0329 1099

Determine the original plaintext message.
[*Hint:* The recovery exponent is $j = 7$.]

15. Decrypt the ciphertext

1369 1436 0119 0385 0434 1580 0690

that was encrypted using the RSA algorithm with key $(n, k) = (2419, 211)$.
[*Hint:* The recovery exponent is 11. Note that it may be necessary to fill out a plaintext block by adding zeros on the left.]

10.2 THE KNAPSACK CRYPTOSYSTEM

A public-key cryptosystem also can be based on the classic problem in combinatorics known as the *knapsack problem*, or the subset sum problem. This problem may be stated as follows: Given a knapsack of volume V and n items of various volumes $a_1, a_2, \ldots, a_n$, can a subset of these items be found that will completely fill the knapsack? There is an alternative formulation: For positive integers $a_1, a_2, \ldots, a_n$ and a sum V, solve the equation

$$V = a_1x_2 + a_2x_2 + \cdots + a_nx_n$$

where $x_i = 0$ or 1 for $i = 1, 2, \ldots, n$.

There might be no solution, or more than one solution, to the problem, depending on the choice of the sequence $a_1, a_2, \ldots, a_n$ and the integer V. For instance, the knapsack problem

$$22 = 3x_1 + 7x_2 + 9x_3 + 11x_4 + 20x_5$$

is not solvable; but

$$27 = 3x_1 + 7x_2 + 9x_3 + 11x_4 + 20x_5$$

has two distinct solutions, namely

$$x_2 = x_3 = x_4 = 1 \qquad x_1 = x_5 = 0$$

and

$$x_2 = x_5 = 1 \qquad x_1 = x_3 = x_4 = 0$$

Finding a solution to a randomly chosen knapsack problem is notoriously difficult. None of the known methods for attacking the problem are substantially less time-consuming than is conducting an exhaustive direct search, that is, by testing all the 2^n possibilities for $x_1, x_2, \ldots, x_n$. This is computationally impracticable for n greater than 100, or so.

However, if the sequence of integers $a_1, a_2, \ldots, a_n$ happens to have some special properties, the knapsack problem becomes much easier to solve. We call a sequence

$a_1, a_2, \ldots, a_n$ *superincreasing* when each a_i is larger than the sum of all the preceding ones; that is,

$$a_i > a_1 + a_2 + \cdots + a_{i-1} \qquad i = 2, 3, \ldots, n$$

A simple illustration of a superincreasing sequence is 1, 2, 4, 8, $\ldots$, 2^n, where $2^i > 2^i - 1 = 1 + 2 + 4 + \cdots + 2^{i-1}$. For the corresponding knapsack problem,

$$V = x_1 + 2x_2 + 4x_3 + \cdots + 2^nx_n \qquad V < 2^{n+1}$$

the unknowns x_i are just the digits in the binary expansion of V.

Knapsack problems based on superincreasing sequences are uniquely solvable whenever they are solvable at all, as our next example shows.

Example 10.4. Let us solve the superincreasing knapsack problem

$$28 = 3x_1 + 5x_2 + 11x_3 + 20x_4 + 41x_5$$

We start with the largest coefficient in this equation, namely 41. Because $41 > 28$, it cannot be part of our subset sum; hence $x_5 = 0$. The next-largest coefficient is 20, with $20 < 28$. Now the sum of the preceding coefficients is $3 + 5 + 11 < 28$, so that these cannot fill the knapsack; therefore 20 must be included in the sum, and so $x_4 = 1$. Knowing the values of x_4 and x_5, the original problem may be rewritten as

$$8 = 3x_1 + 5x_2 + 11x_3$$

A repetition of our earlier reasoning now determines whether 11 should be in our knapsack sum. In fact, the inequality $11 > 8$ forces us to take $x_3 = 0$. To clinch matters, we are reduced to solving the equation $8 = 3x_1 + 5x_2$, which has the obvious solution $x_1 = x_2 = 1$. This identifies a subset of 3, 5, 11, 20, 41 having the desired sum:

$$28 = 3 + 5 + 20$$

It is not difficult to see how the procedure described in Example 10.4 operates, in general. Suppose that we wish to solve the knapsack problem

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

where $a_1, a_2, \ldots, a_n$ is a superincreasing sequence of integers. Assume that V can be obtained by using some subset of the sequence, so that V is not larger than the sum $a_1 + a_2 + \cdots + a_n$. Working from right to left in our sequence, we begin by letting $x_n = 1$ if $V \geq a_n$ and $x_n = 0$ if $V < a_n$. Then obtain $x_{n-1}, x_{n-2}, \ldots, x_1$, in turn, by choosing

$$x_i = \begin{cases} 1 & \text{if } V - (a_{i+1}x_{i+1} + \cdots + a_nx_n) \geq a_i \\ 0 & \text{if } V - (a_{i+1}x_{i+1} + \cdots + a_nx_n) < a_i \end{cases}$$

With this algorithm, knapsack problems using superincreasing sequences can be solved quite readily.

A public-key cryptosystem based on the knapsack problem was devised by R. Merkle and M. Hellman in 1978. It works as follows. A typical user of the system starts by choosing a superincreasing sequence $a_1, a_2, \ldots, a_n$. Now select a modulus $m > 2a_n$ and a multiplier a, with $0 < a < m$ and $\gcd(a, m) = 1$. This ensures that

the congruence $ax \equiv 1 \pmod{m}$ has a unique solution, say, $x \equiv c \pmod{m}$. Finally, form the sequence of integers $b_1, b_2, \ldots, b_n$ defined by

$$b_i \equiv aa_i \pmod{m} \qquad i = 1, 2, \ldots, n$$

where $0 < b_i < m$. Carrying out this last transformation generally destroys the superincreasing property enjoyed by the a_i.

The user keeps secret the original sequence $a_1, a_2, \ldots, a_n$, and the numbers m and a, but publishes $b_1, b_2, \ldots, b_n$ in a public directory. Anyone wishing to send a message to the user employs the publicly available sequence as the encryption key.

The sender begins by converting the plaintext message into a string M of 0's and 1's using the binary equivalent of 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

For example, the message

First Place

would be converted into the numerical representation

$$M = 00101 \quad 01000 \quad 10001 \quad 10010 \quad 10011 \quad 01111 \quad 01011 \quad 00000 \quad 00010 \quad 00100$$

The string is then split into blocks of n binary digits, with the last block being filled out with 1's at the end, if necessary. The public encrypting sequence $b_1, b_2, \ldots, b_n$ is next used to transform a given plaintext block, say $x_1x_2\cdots x_n$, into the sum

$$S = b_1x_1 + b_2x_2 + \cdots + b_nx_n$$

The number S is the hidden information that the sender transmits over a communication channel, which is presumed to be insecure.

Notice that because each x_i is either 0 or 1, the problem of recreating the plaintext block from S is equivalent to solving an apparently difficult knapsack problem ("difficult" because the sequence $b_1, b_2, \ldots, b_n$ is not necessarily superincreasing). On first impression, the intended recipient and any eavesdropper are faced with the same task. However, with the aid of the private decryption key, the recipient can change the difficult knapsack problem into an easy one. No one without the private key can make this change.

Knowing c and m, the recipient computes

$$S' \equiv cS \pmod{m} \qquad 0 \le S' < m$$

or, expanding this,

$$\begin{aligned} S' &\equiv cb_1x_1 + cb_2x_2 + \cdots + cb_nx_n \pmod{m} \\ &\equiv caa_1x_1 + caa_2x_2 + \cdots + caa_nx_n \pmod{m} \end{aligned}$$

Now $ca \equiv 1 \pmod{m}$, so that the previous congruence becomes

$$S' \equiv a_1x_1 + a_2x_2 + \cdots + a_nx_n \pmod{m}$$

Because m was initially chosen to satisfy $m > 2a_n > a_1 + a_2 + \cdots + a_n$, we obtain $a_1x_1 + a_2x_2 + \cdots + a_nx_n < m$. In light of the condition $0 \le S' < m$, the equality

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

must hold. The solution to this superincreasing knapsack problem furnishes the solution to the difficult problem, and the plaintext block $x_1x_2\cdots x_n$ of n digits is thereby recovered from S.

To help make the technique clearer, we consider a small-scale example with $n = 5$.

Example 10.5. Suppose that a typical user of this cryptosystem selects as a secret key the superincreasing sequence 3, 5, 11, 20, 41, the modulus $m = 85$, and the multiplier $a = 44$. Each member of the superincreasing sequence is multiplied by 44 and reduced modulo 85 to yield 47, 50, 59, 30, 19. This is the encryption key that the user submits to the public directory.

Someone who wants to send a plaintext message to the user, such as

HELP US

first converts it into the following string of 0's and 1's:

$$M = 00111 \quad 00100 \quad 01011 \quad 01111 \quad 10100 \quad 10010$$

The string is then broken up into blocks of digits, in the current case blocks of length 5. Using the listed public key to encrypt, the sender transforms the successive blocks into

$$\begin{aligned} 108 &= 47\cdot 0 + 50\cdot 0 + 59\cdot 1 + 30\cdot 1 + 19\cdot 1 \\ 59 &= 47\cdot 0 + 50\cdot 0 + 59\cdot 1 + 30\cdot 0 + 19\cdot 0 \\ 99 &= 47\cdot 0 + 50\cdot 1 + 59\cdot 0 + 30\cdot 1 + 19\cdot 1 \\ 158 &= 47\cdot 0 + 50\cdot 1 + 59\cdot 1 + 30\cdot 1 + 19\cdot 1 \\ 106 &= 47\cdot 1 + 50\cdot 0 + 59\cdot 1 + 30\cdot 0 + 19\cdot 0 \\ 77 &= 47\cdot 1 + 50\cdot 0 + 59\cdot 0 + 30\cdot 1 + 19\cdot 0 \end{aligned}$$

The transmitted ciphertext consists of the sequence of positive integers

108 59 99 158 106 77

To read the message, the legitimate receiver first solves the congruence $44x \equiv 1 \pmod{85}$, yielding $x \equiv 29 \pmod{85}$. Then each ciphertext number is multiplied by 29 and reduced modulo 85, to produce a superincreasing knapsack problem. For instance,

108 is converted to 72, because $108 \cdot 29 \equiv 72 \pmod{85}$; and the corresponding knapsack problem is

$$72 = 3x_1 + 5x_2 + 11x_3 + 20x_4 + 41x_5$$

The procedure for handling superincreasing knapsack problems quickly produces the solution $x_1 = x_2 = 0, x_3 = x_4 = x_5 = 1$. In this way, the first block 00111 of the binary equivalent of the plaintext is recovered.

The Merkle-Hellman cryptosystem aroused a great deal of interest when it was first proposed, because it was based on a provably difficult problem. However, in 1982 A. Shamir invented a reasonably fast algorithm for solving knapsack problems that involved sequences $b_1, b_2, \ldots, b_n$, where $b_i \equiv aa_i \pmod{m}$ and $a_1, a_2, \ldots, a_n$ is superincreasing. The weakness of the system is that the public encryption key $b_1, b_2, \ldots, b_n$ is too special; multiplying by a and reducing modulo m does not completely disguise the sequence $a_1, a_2, \ldots, a_n$. The system can be made somewhat more secure by iterating the modular multiplication method with different values of a and m, so that the public and private sequences differ by several transformations. But even this construction was successfully broken in 1985. Although most variations of the Merkle-Hellman scheme have been shown to be insecure, there are a few that have, so far, resisted attack.

PROBLEMS 10.2

1. Obtain all solutions of the knapsack problem
$$21 = 2x_1 + 3x_2 + 5x_3 + 7x_4 + 9x_5 + 11x_6$$
2. Determine which of the sequences below are superincreasing:
   (a) 3, 13, 20, 37, 81.
   (b) 5, 13, 25, 42, 90.
   (c) 7, 27, 47, 97, 197, 397.
3. Find the unique solution of each of the following superincreasing knapsack problems:
   (a) $118 = 4x_1 + 5x_2 + 10x_3 + 20x_4 + 41x_5 + 99x_6$.
   (b) $51 = 3x_1 + 5x_2 + 9x_3 + 18x_4 + 37x_5$.
   (c) $54 = x_1 + 2x_2 + 5x_3 + 9x_4 + 18x_5 + 40x_6$.
4. Consider a sequence of positive integers $a_1, a_2, \ldots, a_n$, where $a_{i+1} > 2a_i$ for $i = 1, 2, \ldots, n-1$. Show that the sequence is superincreasing.
5. A user of the knapsack cryptosystem has the sequence 49, 32, 30, 43 as a listed encryption key. If the user's private key involves the modulus $m = 50$ and multiplier $a = 33$, determine the secret superincreasing sequence.
6. The ciphertext message produced by the knapsack cryptosystem employing the superincreasing sequence 1, 3, 5, 11, 35, modulus $m = 73$, and multiplier $a = 5$ is 55, 15, 124, 109, 25, 34. Obtain the plaintext message.
   [*Hint:* Note that $5 \cdot 44 \equiv 1 \pmod{73}$.]
7. A user of the knapsack cryptosystem has a private key consisting of the superincreasing sequence 2, 3, 7, 13, 27, modulus $m = 60$, and multiplier $a = 7$.
   (a) Find the user's listed public key.
   (b) With the aid of the public key, encrypt the message *SEND MONEY*.

10.3 AN APPLICATION OF PRIMITIVE ROOTS TO CRYPTOGRAPHY

Most modern cryptographic schemes rely on the presumed difficulty of solving some particular number theoretic problem within a reasonable length of time. For instance, the security underlying the widely used RSA cryptosystem discussed in Section 10.1 is the sheer effort required to factor large numbers. In 1985, Taher ElGamal introduced a method of encrypting messages based on a version of the so-called discrete logarithm problem: that is, the problem of finding the power $0 < x < \phi(n)$, if it exists, which satisfies the congruence $r^x \equiv y \pmod{n}$ for given r, y, and n. The exponent x is said to be discrete logarithm of y to the base r, modulo n. The advantage of requiring that the base r be a primitive root of prime number n is the assurance that y will always have a well-defined discrete logarithm. The logarithm could be found by exhaustive search; that is, by calculating the successive powers of r until $y \equiv r^x \pmod{n}$ is reached. Of course, this would generally not be practical for a large modulus n of several hundred digits.

Example 8.4 indicates that, say, the discrete logarithm of 7 to the base 2 modulo 13 is 11; expressed otherwise, 11 is the smallest positive integer x for which $2^x \equiv 7 \pmod{13}$. In that example, we used the classical notation $11 = \text{ind}_2 7 \pmod{13}$ and spoke of 11 as being the index of 7, rather than employing the more current terminology.

The ElGamal cryptosystem, like the RSA system, requires that each user possess both a public and a private (secret) key. The means needed to transmit a ciphered message between parties is announced openly, even published in a directory. However, deciphering can be done only by the intended recipient using a private key. Because knowledge of the public key and the method of encipherment is not sufficient to discover the other key, confidential information can be communicated over an insecure channel.

A typical user of this system begins by selecting a prime number p along with one of its primitive roots r. Then an integer k, where $2 \leq k \leq p-2$, is randomly chosen to serve as the secret key; thereafter,

$$a \equiv r^k \pmod{p} \quad 0 \leq a \leq p-1$$

is calculated. The triple of integers (p, r, a) becomes the person's public key, made available to all others for cryptographic purposes. The value of the exponent k is never revealed. For an unauthorized party to discover k would entail solving a discrete logarithm problem that would be nearly intractable for large values of a and p.

Before looking at the enciphering procedure, we illustrate the selection of the public key.

Example 10.6. Suppose that an individual begins by picking the prime $p = 113$ and its smallest primitive root $r = 3$. The choice $k = 37$ is then made for the integer satisfying $2 \leq k \leq 111$. It remains to calculate $a \equiv 3^{37} \pmod{113}$. The exponentiation can be readily accomplished by the technique of repeated squaring, reducing