1 Introduction

Information and the communication of that information comprise the nerve system of civilization, and civilization depends on the availability of reliable methods for the protection of information from intruders and adversaries. There are many ways the collection and communication of information needs to be protected and made trustworthy. The requirements are central to the orderly functioning of society and may include secrecy, integrity, nonrepudiation, authentication, covertness, copy resistance, certification, authorization, and ownership protection. These various topics can be regarded as more or less distinct requirements, although of course there are considerable overlaps. Together they form the topic of secure communications. At the center of these various topics, as well as at the heart of this book, is the classical topic of cryptography.

Communication and cryptography are closely related topics in the general field of telecommunication. Communication is the process of exchanging data and messages. By itself, the term communication carries an active, positive tone and suggests cooperation and openness. Yet the process of communication does have its competitive, defensive side. The nature of social and economic interaction can impose a great variety of subtle requirements on the structure of a communication system to ensure various forms of security, privacy, and trustworthiness.

Secrecy and authentication are complementary functions in a communication system. Secrecy is the function that ensures that a message cannot be understood by an eavesdropper. Authentication is the function that ensures that the message originated with the indicated source of that message. The purpose of authentication is to verify the source of the message. It does not verify the identity of the transmitter of a message, a function known as identification. Authentication can be provided by a digital signature that must be impervious to forgery. This requirement leads to connections with cryptography. Cryptography consists of the study, development, and implementation of methods for protecting data; cryptanalysis is the study, development, and implementation of methods for attacking and breaking cryptosystems. Taken together, the two subjects of cryptography and cryptanalysis form the topic of cryptology.

The elementary notion of a classical *cryptosystem* is familiar to many puzzle solvers who enjoy recreational cryptograms. Recreational cryptograms are a widespread feature of newspapers and are popular with puzzle solvers. Because each letter of the

alphabet is represented in a cryptogram by another letter, a solution amounts to finding which of 26! possible permutations is the "key." These recreational cryptograms usually appear with the word spaces intact, which gives useful information to the solver and allows the puzzles to be solved quite simply. However, even without the word spaces, the cryptograms are still rather easy to solve. We conclude from this observation that one does not actually try all 26! possible permutations. This would be a formidable task. Instead, the puzzle solver uses some kind of hierarchical structure in which the permutation is deduced by a sequence of inferences. In the language of this book, the recreational *cryptanalyst* attacks the cryptosystem using prior knowledge of both the structure of the cryptosystem and the linguistic structure and content of the likely encrypted messages.

1.1 Classical cryptography

A message $(x_1, x_2, \ldots, x_n)$, herein called a *plaintext message*, consists of a sequence of n symbols from a given finite alphabet $\mathcal{A}$ of size $\#\mathcal{A}$. For convenience, we may often regard the length n of the plaintext message to be fixed in advance in order to avoid certain uninteresting distractions that can arise when treating variable-length messages. Then the message $(x_1, x_2, \ldots, x_n)$ is an element of $\mathcal{A}^n$, but not every element of $\mathcal{A}^n$ need be a legitimate plaintext message of the given application. Let $\mathcal{M} \subset \mathcal{A}^n$ denote the set of legitimate plaintext messages. When appropriate, the set $\mathcal{M}$ is called a *natural language* or, more simply, a *language*. This statement implies that, in general, not every element of $\mathcal{A}^n$ is a plaintext message. This is denoted by $\mathcal{M} \neq \mathcal{A}^n$. Indeed, the cardinality of $\mathcal{M}$ is usually much smaller than the cardinality of $\mathcal{A}^n$, which we write as $\#\mathcal{M} \ll \#\mathcal{A}^n$. This observation plays an important role in the information-theoretic approach to cryptography.

For the simplest model of the English language, $m = 26$, all letters are upper case, and there is no symbol for a space. Whenever we require messages to have a fixed length n, we may pad a shorter message at the end with copies of a filler symbol, such as the symbol Z (or a blank space or other alternative null symbol), to make the message have the standard length n. For this simple model of English, $\mathcal{A}^n = 26^n$, but the legitimate plaintext messages, as determined by $\mathcal{M}$, are far fewer.

Ciphers may be classified as either of two types: *block ciphers* and *stream ciphers*. A block cipher of blocklength n first segments a longer plaintext message $\boldsymbol{x}$, say of length N, into N/n blocks (or segments); each block has length n, and the message is now written $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N/n}\}$, where $\boldsymbol{x}_\ell$ is the ℓth block of the message. For this purpose, we require that n divides N. Each plaintext block $\boldsymbol{x}_\ell$, for $\ell = 1, \ldots, N/n$, consists of n sequential symbols of the message, with each symbol from the given alphabet $\mathcal{A}$. Thus $\boldsymbol{x}_\ell \in \mathcal{A}^n$.

For many elementary block ciphers, $n = 1$. This means that symbols are encrypted independently, although by a common encryption rule. Such ciphers are far too trivial to be used in serious applications, although they are popular in recreational applications, and will be described here.

A *block encryption function* is a map, denoted $e(\boldsymbol{x})$, that maps a message block $\boldsymbol{x}$ of length n into another block, often also of length n, called a *ciphertext block*, and denoted $\boldsymbol{y} = e(\boldsymbol{x})$. Thus $\boldsymbol{y} \in \mathcal{A}^n$. A *key* is an element k from a set $\mathcal{K}$ called the *keyspace*. The function of a key $k \in \mathcal{K}$ is to specify an encryption function, now denoted $e_k(\boldsymbol{x})$, from a predefined set of encryption functions indexed by k. A *block encryptor* is a collection of maps, denoted $\{e_k(\boldsymbol{x}), k \in \mathcal{K}\}$, forming the set of encryption functions. A *block decryptor* is the collection of inverse maps $\{d_k(\boldsymbol{y}) = e_k^{-1}(\boldsymbol{y})\}$, also indexed by k. Thus for each k, $d_k(e_k(\boldsymbol{x})) = \boldsymbol{x}$. This arrangement, in which both the encryptor and the decryptor have access to the same key k, is called a *symmetric-key cryptosystem*. We will see eventually, and perhaps surprisingly, that some cryptosystems have a different key at the encryptor and the decryptor. This is called an *asymmetric-key cryptosystem*. In this case, each decryption key, which itself must be kept secret, is associated with a corresponding encryption key, which can be made public. This encryption key needs to be suitably published only once by that decryptor or by a trusted proxy. Thereafter, the decryptor or the proxy can be completely passive. Indeed, the proxy can be dissolved. It is no longer needed for this purpose. In contrast, in order to form a key using a symmetric cryptosystem, the decryptor must actively interact with every transmitter that intends to send one or more encrypted messages. For practical reasons, this interaction needed to create a secret symmetric key is usually public, and so it is referred to as a *public key exchange*. A public key exchange is different from a *public encryption key*, though both are public.

Another advantage of an asymmetric-key cryptosystem is that the leaking of the encryption key does not compromise the system because, in fact, the encryption key is public. Of course, this consideration requires that neither the decryption key nor the plaintext can be deduced from the public encryption key and the ciphertext. In particular, this system is vulnerable if it is used with a message space so small that it is possible to simply encrypt every possible message with the public encryption key until the given ciphertext is observed, thereby revealing the actual plaintext corresponding to that ciphertext. This possibility is countered by padding all short messages with randomly generated bits. In this way, inadequate entropy in the message space is supplemented with the additional entropy of the random padding.

The *cipherspace*, or *cryptspace*, of the key k is the set $e_k(\mathcal{M})$. The cipherspace is contained in $\mathcal{A}^N$ whenever the alphabet and blocklength of the ciphertext are the same as the alphabet and blocklength of the plaintext. It is clear that $d_k(e_k(\mathcal{M})) = \mathcal{M}$. However, in general, $d_{k'}(e_k(\mathcal{M})) \neq \mathcal{M}$ for $k' \neq k$. Indeed, it may be true that $d_{k'}(e_k(\mathcal{M})) \cap \mathcal{M} = \phi$ for every $k' \neq k$. In that case, in principle, the encryption is not secure. It is vulnerable to a *direct attack*. Simply compute $d_{k'}(\boldsymbol{y})$ for every k' and choose

that k' for which $d_{k'}(\boldsymbol{y}) \in \mathcal{M}$. Of course, if $\#\mathcal{K}$, the cardinality of the keyspace $\mathcal{K}$, is large enough, a direct attack may not be feasible because of excessive computations. Thus from a practical point of view, secrecy may depend on computational resources.

In the contrary case, it may be that $d_{k'}(e_k(\mathcal{M})) = \mathcal{M}$ for all $k' \neq k$ and for all k. This is an example of perfect secrecy. For any ciphertext $\boldsymbol{y}$ and any plaintext $\boldsymbol{x}$ there is a key k for which $e_k(\boldsymbol{x}) = \boldsymbol{y}$. *Any* plaintext message could be the right one. Only if the key is known or partially known can the plaintext be deduced, or partially deduced, from the ciphertext.

The only cryptosystem proven to have perfect secrecy is the one-time pad. Suppose that both the encryptor and the decryptor have an identical copy of a long, random, binary sequence $\boldsymbol{u} = (u_\ell, \ell = 0, \ldots)$, called a one-time pad. Let $\boldsymbol{x} = (x_\ell, \ell = 0, \ldots,)$ be the plaintext represented in the form of a binary sequence. The encryptor transmits $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{u}$ (where $+$ denotes componentwise modulo-two addition) and the decryptor computes $\boldsymbol{y} + \boldsymbol{u}$. Because $\boldsymbol{u} + \boldsymbol{u} = \boldsymbol{0}$ modulo two, the decryptor has recovered the plaintext $\boldsymbol{x}$. The adversary, or eavesdropper, sees only $\boldsymbol{y}$, which is an unintelligible ciphertext with its alphabet and blocklength the same as that of the plaintext. It is impossible to recover $\boldsymbol{x}$, in whole or in part, without knowing $\boldsymbol{u}$.

The one-time pad will not be fully secure if the same random sequence $\boldsymbol{u}$ of the pad is used twice. This is because, $\boldsymbol{u} + \boldsymbol{u} = \boldsymbol{0}$ modulo two, so if the two messages $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are encrypted with the same binary key, then $\boldsymbol{y}_1 = \boldsymbol{x}_1 + \boldsymbol{u}$ and $\boldsymbol{y}_2 = \boldsymbol{x}_2 + \boldsymbol{u}$, from which the cryptanalyst can compute that $\boldsymbol{y}_1 + \boldsymbol{y}_2 = \boldsymbol{x}_1 + \boldsymbol{x}_2$. Because $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{M}$, and $\mathcal{M}$ is presumably sparse in $\mathcal{A}^n$, it is possible with sufficient computational resources to list all $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ for which $\boldsymbol{x}_1 + \boldsymbol{x}_2$ equals $\boldsymbol{y}_1 + \boldsymbol{y}_2$. We may expect this list of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ to be much smaller than $|\mathcal{M}|^2$. In this way, the two ciphertexts have been partially decrypted by reducing them to a list of possible pairs of plaintexts.

Although the one-time pad is secure at the cryptographic level, it may still fail against *side-channel attacks*. Its weakness is that it requires both the encryptor and the decryptor to have an identical copy of the one-time pad. In practice, this means that the problem of cryptographic security is replaced by the problem of the distribution and protection of the one-time pad. This would be an equivalent problem unless the circumstances for the distribution of the one-time pad could be made different from the circumstances for the distribution of the ciphertext. For example, a trusted courier can physically carry the one-time pad from one user to the other, or the one-time pad could be distributed at an earlier time, in a different situation, and over a channel known to be secure.

Classical methods of key distribution are not practical on a large public network. Accordingly, new methods of public key exchange or public key agreement have been developed in recent years both for message secrecy and for message signatures. Surprisingly, such methods are in use for creating secret keys between two strangers in full view of adversarial third parties. These modern methods depend on the computational intractability of the adversarial cryptanalyst's task.

The practice of cryptography requires that formal procedures for using a given cryptographic technique be in place so that the key is not divulged by improper use. These methods are referred to as *cryptographic protocols*, which often can be understood without understanding the cryptographic techniques themselves. The protocols are variously studied under the name *information security*, rather than *information secrecy*, which more usually refers to the cryptography. The subject of this book falls primarily under the heading of information secrecy, but the methods of information security are close at hand, and also require discussion. Indeed, it is often not possible to draw a clear distinction between the tasks of secrecy and security.

1.2 Notions of cryptographic secrecy

Classical cryptography relies on the distribution of a secret key over a secure channel. This is unacceptable in modern applications because communication routinely takes place over large public networks between strangers, with no opportunity to prearrange the distribution of a secret key. Secure communication between two such parties will require that the two parties set up their key over a public channel and in the clear. This is both possible and routinely done. More precisely, it is generally believed that the methods now in common use for doing this are secure. It is possible – or so we believe – for two parties to arrange a secret key, one that only they will know, using only communications over a public channel in full view of a sophisticated adversary. Such methods can only give computational or practical secrecy, never perfect secrecy. They can be broken by an adversary with infinite computational resources. Much of this book is devoted to the study of methods of doing this, and to ways of attacking such methods.

Early attempts to define cryptographic security were based on the notion of *perfect secrecy* or *perfect security*, a notion that is only assured by a one-time pad and is usually impossible to meet in everyday situations. This notion has now been supplemented by alternative and more practical definitions of cryptography motivated by the question "When is a cryptographic system computationally secure?" The notion of computational security underlies the modern methods of public-key cryptography, but the notion of perfect secrecy remains as a stronger and more desirable requirement that is rarely met. A computationally secure system may be perfectly insecure when judged by a more rigid standard.

We will define four notions of secrecy against an attack in which the cryptanalyst has only the ciphertext, but always has full knowledge of the method of encryption. Only the key is unknown to the cryptanalyst.

1. Perfect or information-theoretic secrecy: The ciphertext gives no information about the plaintext.

2. Unconditional secrecy: It is impossible to break the ciphertext with infinite computational resources, even with the best *possible* algorithm.
3. Computational secrecy: It is intractable to break the ciphertext with "practical" computational resources, even with the best *possible* algorithm.
4. Practical secrecy: It is intractable to break the ciphertext with "practical" computational resources using the best *known* algorithm.

These four notions of secrecy are listed in the order of their decreasing apparent weakness. We would like the statements we make about cryptographic systems to be as strong as possible. However, we are usually unable to make a statement that a given cryptosystem is perfectly secure – because most are not. We can state only what we know, which is that for many cryptography systems, the system is perfectly insecure from a theoretical point of view, meaning that each of these systems can be broken in unlimited computational time or with unlimited computational resources. Moreover, we do not often know the complexity of the best possible algorithm for a given task. Usually we do not know the best possible algorithm. Statements about the best possible algorithm are either not known or are heavily qualified. Usually, we are content with practical secrecy even though this notion is informal and imprecise. The set of known algorithms changes with time, and the notion of a practical computation changes with time as well. Thus, to claim practical secrecy is to claim nothing of permanent truth.

Any discussion of practical secrecy should include some consideration of the desired longevity of the secrecy. One will use a stronger system for a message that must remain secret for decades than for a message that must remain secret for months. One should assume that, except for the one-time pad, every cryptography system can be broken eventually, though perhaps not for centuries.

A cryptography system also may be vulnerable to various kinds of side-channel attacks. A side-channel attack uses a variety of "externalities" to gain information about a message or a key through the "backdoor." The perceived relationship between the source of the message and the user of the message, or the time and circumstances of message transmission, may be useful to the cryptanalyst. Any measurement regarding the amount of time taken for an encryption or decryption computation may give some information about the key. Such information, even when it is meager, may be useful to the cryptanalyst when combined with other sources of information. It is possible that a faint echo of the plaintext message will be present on one or more wires leaving the encryption device, even perhaps unintentionally leaked in this way on the power cord. Even the electromagnetic or infrared signature of the encryption device may be scrutinized to gain information about the encryption key, however slight. This side-channel information might be used to supplement a direct attack.

Recurring message patterns of the user, such as starting every message with the date or a formal salutation, have been used to successfully attack a cryptosystem. For such reasons, some systems alter the key for each encrypted message with an unencrypted

but randomly chosen number to further randomize the message space. The secret key is modified by adding the random number to it, then appending this random number unencrypted as an attachment to the ciphertext. That random number is added to the secret key for encrypting and for decrypting that specific message. In this way, although a new key is not exchanged, each message is effectively encrypted with a different key in order to further hide recurring user patterns that may be present. Of course, the secrecy resides only in the original key, but certain attacks are foiled in this way.

In the end, it may be that the best way for an adversary to circumvent a modern cryptosystem is to find a way to enter the encryptor before the encryption takes place or to enter the decryptor after the decryption takes place, and to so read the plaintext directly. A totally secure lock on the front door does no real good if the windows are unprotected. This kind of vulnerability to an intruder is outside of the scope of this book.

1.3 Block ciphers

A *block cipher* segments a message into plaintext blocks of a fixed length, encrypts each plaintext block into a corresponding ciphertext block of a fixed length, then concatenates the encrypted blocks to form the encrypted message. Generally, the plaintext blocks and the encrypted blocks have the same length. We will usually assume that this is the case, though it is not always so. The decryptor reverses this procedure, breaking the concatenated stream of ciphertext blocks into individual blocks and decrypting these blocks one by one. The decrypted plaintext blocks are then concatenated to recover the original message.

We will describe elementary block ciphers in this section, mostly by way of examples. These elementary block ciphers are much too weak to be useful in any serious application. Some elementary ciphers may be useful for providing rudimentary privacy, but are easily broken by straightforward computational methods. They serve here only as a way to introduce the subject of cryptography and its terminology, to suggest its history, and to motivate the search for secure methods.

The simpler of the elementary block ciphers have a blocklength n equal to one. Each block x_ℓ of the plaintext is only a single symbol of the alphabet $\mathcal{A}$, and each block $y_\ell = e_k(x_\ell)$ of the ciphertext is also a single symbol of the alphabet $\mathcal{A}$. The message $(x_1, \ldots, x_n)$ then is encrypted, one symbol at a time, to produce the ciphertext $(y_1, \ldots, y_n) = (e_k(x_1), \ldots, e_k(x_n))$. In the simplest case, the key k and the encryption function e_k remain the same for all symbols of the message.

For an elementary block cipher of blocklength n equal to one, an encryption function will only be a permutation of the alphabet $\mathcal{A}$, i.e., the $q = \#\mathcal{A}$ symbols of the alphabet $\mathcal{A}$ are encrypted by a permutation of the alphabet, denoted by $\sigma_k : \mathcal{A} \to \mathcal{A}$, or $x \mapsto \sigma_k(x)$. Each key k specifies one permutation. Thus the keyspace $\mathcal{K}$ is a subset of $\mathcal{S}_q$, where $\mathcal{S}_q$ is the group of permutations on a set of q elements.

Some elementary ciphers have standard names. A *substitution cipher*[1] of blocklength one (not to be confused with a permutation cipher to be discussed later) is a cipher in which the set of keys is the set of all permutations of an alphabet of size $q = \#\mathcal{A}$. Thus $\mathcal{K} = \mathcal{S}_q$, and $\#\mathcal{K} = q!$. It is common to refer to $q!$ as the cardinality of the key space, and the size of the equivalent binary key needed to specify a permutation is $\log_2(q!)$ bits. For example, if $q = 26$, the size of the equivalent binary key is $\log_2(26!) \approx 90$, so the equivalent binary key size is about 90 bits. A single key of the recreational cryptogram is the permutation

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

There are approximately 2^{90} such substitution keys on an alphabet of 26 letters. The set of recreational cryptograms is an example of a substitution cipher on an alphabet with 26 letters.

For a larger example of a substitution cipher, a standard keyboard is often regarded as having 256 characters. Then the alphabet has size 256. Thus there are 256! permutations on this set, so a substitution cipher that used all possible keys would have an equivalent key size of $\log_2(256!)$ bits. This is a very large number, and it would require a binary word with more than a thousand bits to specify one key from this set. Accordingly, one will usually choose to restrict the keyspace in some way, as determined by the details of the encryption algorithm. The actual key will be much smaller than $\log_2(256!)$ bits, and so the set of allowable permutations will be much smaller as well.

All block ciphers of blocklength one are actually substitution ciphers. The named block ciphers of blocklength one are defined by restricting the set of permutations to those that can be described by some specialized simpler rule. A *shift cipher* is a special case of a substitution cipher. A shift cipher consists only of permutations that are cyclic shifts of alphabet $\mathcal{A}$. Let $\mathcal{A}$ be represented by the set of integers with addition modulo q, denoted $\boldsymbol{Z}_q$. If $\mathcal{A}$ is the roman alphabet, then $q = 26$ and the letters of $\mathcal{A}$ can be represented by the integers from 1 to 26. Then the encryption function is $e_k(x_i) = x_i + k \pmod{26}$. The keyspace of the shift cipher has size 26 and the equivalent binary key size is $\log_2 26$, which is less than five bits. Because the keyspace is so small, the shift cipher is not secure, though it may be used as a privacy cipher.

An *affine cipher* of blocklength one is given by $e_k(x_i) = ax_i + b \pmod{26}$. So that this function can be inverted, $a^{-1} \pmod{26}$ must exist, but b can be any element of $\boldsymbol{Z}_{26}$. Elementary number theory, which is reviewed in the next chapter, states that for the inverse a^{-1} to exist, a must satisfy $\mathrm{GCD}(a, q) = 1$. The set of a that have inverses under modulo-q arithmetic is denoted $\boldsymbol{Z}_q^*$. For the roman alphabet, $q = 26$ and $\boldsymbol{Z}_{26}^* = 12$.

[1] A substitution cipher permutes the letters of the alphabet. A permutation cipher permutes the letters of the message block.

Therefore the affine cipher $e_k(x_i) = ax_i + b \pmod{26}$ has $12 \cdot 26$ keys (12 choices for a and 26 choices for b), and the equivalent binary key size is less than nine bits. Of these, the trivial key with $a = 1$ and $b = 0$, and perhaps other trivial keys, should be avoided.

Block ciphers with a blocklength n larger than one can be defined similarly. If each symbol is from an alphabet $\mathcal{A}$ of size q, then each block of length n is from an alphabet $\mathcal{A}^n$ of size q^n. Then $\mathcal{K}$ is a subset of $\mathcal{S}_{q^n}$, the group of permutations on q^n elements. It may even be that $\mathcal{K}$ is equal to $\mathcal{S}_{q^n}$. Then the cipher is called a *block substitution cipher*.

For example, the simplest model of the English language has $q = 26$. If $n = 2$, then there are $26^2 = 676$ pairs of letters. A block substitution cipher on blocks of length two replaces each pair of letters with a substitute pair of letters. This block cipher is a set of simple look-up tables, each with 676 entries, one entry for each pair of letters, and one such look-up table for each key. To break this blocklength-two substitution cipher is much harder than it is to break the elementary recreational cryptogram of blocklength one described earlier. Because there are $(26^2)!$ permutations on pairs of letters, there are $(26^2)!$ distinct blocklength-two substitution ciphers on an alphabet of size 26. This is an immense number, not to be confused with $(26!)^2$. If all of these permutations are indexed by keys in a keyspace $\mathcal{K}$, then the size of the keyspace is $(26^2)!$. One cannot attack this cipher by exhaustively trying all keys; there are far too many. Because $(26)^2!$ is an immense number – too large to index the keys – one should expect that only a subset of these permutations would be used as keys in a given practical cipher, and that restricted keyspace would be defined by a tractable rule for defining the key. A cryptanalyst would use knowledge of this restricted keyspace, were it known, but this knowledge will presumably be useless in a well-designed cipher.

There are elementary block ciphers of this kind that have standard names: A *Vigenère cipher* is a componentwise additive cipher on blocks of length n symbols using an additive key of length n symbols. For example, let $SNOW$ denote a Vigenère key with blocklength $n = 4$. This key has the numerical equivalent (19, 15, 16, 23). To encrypt the word "ball," add componentwise modulo 26 the numerical equivalents $(2, 1, 12, 12) + (19, 15, 16, 23) = (21, 16, 2, 9)$, which becomes "UPBI." To decrypt, again using the key "SNOW," write $(21, 16, 2, 9) - (19, 15, 16, 23) = (2, 1, 12, 12)$, which recovers the word "ball."

A *Hill cipher* of blocklength two has the form

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \boldsymbol{M} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

under modulo-q arithmetic, where $\boldsymbol{M}$ is an invertible matrix modulo q. The matrix will be invertible if $(\det \boldsymbol{M})^{-1}$ exists in $\boldsymbol{Z}_q$, and this inverse exists if the greatest common divisor of $\det \boldsymbol{M}$ and q is equal to one. The Hill cipher provides our first example of a cipher that uses number theory in a nontrivial way. For example, to specify a Hill

cipher of blocklength two requires a two-by-two matrix that is invertible modulo 26. The matrix

$$\boldsymbol{M} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

satisfies GCD(det $\boldsymbol{M}$, 26) = 2, so this $\boldsymbol{M}$ cannot be used to key a Hill cipher. The matrix

$$\boldsymbol{M} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}$$

is suitable because now GCD(det $\boldsymbol{M}$, 26) = 1. Therefore

$$\boldsymbol{M}^{-1} = \begin{bmatrix} 21 & 2 \\ 3 & 25 \end{bmatrix}$$

modulo 26. To encrypt the word "ball" with this Hill cipher, the numerical equivalent, (2, 1, 12, 12), is encrypted two letters at a time as

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} 12 \\ 12 \end{bmatrix},$$

modulo 26. Then $y = (4, 11, 10, 18)$, which becomes "DKJR" in the ciphertext. In this way, the plaintext "ball" is represented by the ciphertext "DKJR."

To decrypt, write

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 21 & 2 \\ 3 & 25 \end{bmatrix} \begin{bmatrix} 4 \\ 11 \end{bmatrix}$$

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 21 & 2 \\ 3 & 25 \end{bmatrix} \begin{bmatrix} 10 \\ 18 \end{bmatrix},$$

modulo 26. Then $x = (2, 1, 12, 12)$, which reduces to the word "ball," thereby recovering the plaintext message.

A *permutation cipher* of blocklength N is given by

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \boldsymbol{M} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix},$$

where $\boldsymbol{M}$ is an N-by-N matrix with a single one in every column and a single one in every row and all other elements are equal to zero. Such a matrix is called a *permutation*