10. Scalars, division, affine cryptography

For a number k and $m \times n$ matrix A we write

$$kA \tag{10.1}$$

to denote the $m \times n$ matrix obtained from A by muktiplying each of its entries by k.

■ **Example 10.1**

$$-2\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} = \begin{pmatrix} -2 & -4 & -6 \\ -8 & -10 & -12 \end{pmatrix} \tag{10.2}$$

■

In this setting we refer to the number k as a *scalar*, and we refer to (10.1) as *scalar multiplication*.

10.1 Algebraic properties of scalar multiplication

The equalities

$$\begin{aligned} k(A+B) &= (kA)+(kB) \quad \text{(distributivity)} \\ (k+\ell)A &= kA+\ell A \quad \text{(distributivity)} \end{aligned}$$

hold for any numbers k, ℓ and any $m \times n$ matrices A, B. The equalities

$$\begin{aligned} k(AB) &= (kA)B \quad \text{(associativity)} \\ (k\ell)A &= k(\ell A) \quad \text{(associativity)} \\ k(AB) &= A(kB) \quad \text{(scalar commutativity)} \\ k(\ell A) &= \ell(kA) \quad \text{(scalar commutativity)} \end{aligned}$$

hold for any numbers k, ℓ and any $m \times n$ matrices A, B.

10.2 Identity matrices

Let A be an $n \times n$ matrix. Such a matrix is said to be *square*. A *diagonal entry* of A is an entry lying in the ith row and ith column for some $1 \leq i \leq n$. An entry of A is said to be *non diagonal* if it is

not a diagonal entry. An *identity matrix* is a square matrix whose diagonal entries are all equal to 1, and whose non diagonal entries are all equal to 0. We denote such a matrix by I_n, or just by I if the value of n is clear from the context.

■ **Example 10.2**

$$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{10.3}$$

$$I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{10.4}$$

$$I_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{10.5}$$

■

The equalities

$$IA = A = AI \tag{10.6}$$

hold for any $n \times n$ matrix A and $I = I_n$.

10.3 Inverse matrices

The following definition repeats the idea we used in clock arithmetic for introducing a notion of dibision into our new setting.

Definition 10.3.1 Let A be a square $n \times n$ matrix. An $n \times n$ matrix B is said to be an *inverse* to A if the equalitiies

$$AB = I = BA$$

hold.

■ **Example 10.3** The matrices

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 5 \\ 3 & 8 & 6 \end{pmatrix}, \qquad B = \begin{pmatrix} 10 & -12 & 5 \\ -3 & 3 & -1 \\ -1 & 2 & -1 \end{pmatrix}$$

satisfy

$$AB = I = BA \tag{10.7}$$

and so B is an inverse to A. Is there any other inverse to A? Well, if C were also an inverse to A we'd have:

$$C(AB) = C(I) \tag{10.8}$$

$$(CA)B = C \tag{10.9}$$

$$IB = C \tag{10.10}$$

$$B = C \tag{10.11}$$

■

This little calculation establishes the following fundamental result.

Theorem 10.3.1 A square matrix A has at most one inverse.

If a matrix A has an inverse then we denote this inverse by A^{-1}. Clearly some matrices, such as the zero matrix, have no inverse since we can never have $A0 = I$.

For the moment let us focus on 2×2 matrices. For an arbitrary 2×2 matrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{10.12}$$

we have:

$$A \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \tag{10.13}$$

$$= \begin{pmatrix} ad-bc & 0 \\ 0 & ad-bc \end{pmatrix} \tag{10.14}$$

$$= (ad-bc) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{10.15}$$

$$= (ad-bc)I \tag{10.16}$$

Similarly, we have:

$$\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} A = (ad-bc)I \tag{10.17}$$

Equations (10.16) and (10.17) establish the following formula for calculating inverses of 2×2 matrices.

Theorem 10.3.2 A matrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

has an inverse if and only if the number $ad - bc$ has an inverse. If this number has an inverse then

$$A^{-1} = (ad-bc)^{-1} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

■ **Example 10.4** The matrix

$$A = \begin{pmatrix} 1 & 4 \\ 5 & 7 \end{pmatrix} \tag{10.18}$$

has inverse

$$A^{-1} = (1 \times 7 - 5 \times 4)^{-1} \begin{pmatrix} 7 & -4 \\ -5 & 1 \end{pmatrix} \tag{10.19}$$

$$= \begin{pmatrix} \frac{-7}{13} & \frac{4}{13} \\ \frac{5}{13} & \frac{-1}{13} \end{pmatrix}. \tag{10.20}$$

■

10.4 Affine cryptography

The cryptosystem $f_E : \mathbb{Z}_N \to \mathbb{Z}_N, n \mapsto an + b$ introduced in Section 3.2 has the merit of being simple to implement, and efficient for enciphering long messages. But it has two serious drawbacks:

1. Frequency analysis of the letters in a long ciphertext would lead to a pair of linear simultaneous equations modulo N from which the enciphering and deciphering keys could likely be calculated very quickly. Here N denotes the length of the alphabet used.
2. The number of possibly enciphering keys $E = (a,b)$ is equal to $\phi(N)N$ where the Euler Phi function $\phi(N)$ determines the number of invertible numbers a on an N-hour clock. We refer to $\phi(N)N$ as the size of the *key space*. In our examples we used an alphabet with $N = 26$ letters, for which the key space has size:

$$\begin{aligned} \phi(26)26 &= \phi(2)\phi(13)26 && (10.21)\\ &= 12 \times 26 && (10.22)\\ &= 312 && (10.23) \end{aligned}$$

   For each of the 312 possible enciphering keys one could quickly compute the corresponding deciphering key using the Euclidean algorithm with formulae (3.13) and (3.14); all deciphering keys could be applied to any ciphertext to see which one yields meanigful plaintext. In more realistic examples there would be maybe one letter for each character on a computer key board. A standard Qwerty keyboard has around $N = 104$ characters. In this case the key space would have size:

$$\begin{aligned} \phi(104)104 &= \phi(13)\phi(2^3)104 && (10.24)\\ &= 12(2^3 - 2^2)104 && (10.25)\\ &= 4992 && (10.26) \end{aligned}$$

   Again, we could quickly run through all 4992 possible deciphering keys until we found one that produces meaningful plaintext.

In contrast, the RSA cryptosystem with suitably chosen enciphering and deciphering keys is a secure method for sending messages: frequecy analysis is not applicable since it is public key; it has an infinite number of possible keys. But it has the drawback that it takes a computer quite some time to encipher long messages. In practice, the RSA system is used only for sending short messages.

The simple cryptosystem of Section 3.2 is easily strengthened using matrix algebra. Instead of viewing a message as a sequence of letters, we can fix an integer $k \geq 1$ and regard plaintex as a sequence of k-tuples of letters.

For instance, when $k = 2$ we consider the plaintext

$$\text{HELLO\_WORLD} \tag{10.27}$$

as a sequence of 2-tuples:

$$\begin{pmatrix} \text{H} \\ \text{E} \end{pmatrix} \begin{pmatrix} \text{L} \\ \text{L} \end{pmatrix} \begin{pmatrix} \text{O} \\ \text{\_} \end{pmatrix} \begin{pmatrix} \text{W} \\ \text{O} \end{pmatrix} \begin{pmatrix} \text{R} \\ \text{L} \end{pmatrix} \begin{pmatrix} \text{D} \\ \text{\_} \end{pmatrix} \tag{10.28}$$

Using the correspondence $\text{A} \leftrightarrow 0, \ldots, \text{Z} \leftrightarrow 25, \text{\_} \leftrightarrow 26$ we can represent this sequence as a sequence of 2×1 matrices, or column vectors:

$$\begin{pmatrix} 7 \\ 4 \end{pmatrix} \begin{pmatrix} 11 \\ 11 \end{pmatrix} \begin{pmatrix} 14 \\ 26 \end{pmatrix} \begin{pmatrix} 22 \\ 14 \end{pmatrix} \begin{pmatrix} 17 \\ 11 \end{pmatrix} \begin{pmatrix} 3 \\ 26 \end{pmatrix} \tag{10.29}$$

And now for the exciting part! We can view (10.29) as a sequence of column vectors whose entries are from the set $\mathbb{Z}_{27}$ of integers modulo 27.

In the above discussion of matrix arithmetic we defined a matrix as an array of 'numbers'. Nowhere in the discussion did we insist that these be numbers from the set of reals $\mathbb{R}$. Everything

that we have said about matrices so far holds when the entries are taken from the integers $\mathbb{Z}_N$ modulo any fixed integer N. If we have broken the plaintext into message units consisting of k-tuples of letters, then we can encipher each k-tuple using a function

$$f_E\colon (\mathbb{Z}_N)^k \to (\mathbb{Z}_N)^k, v \mapsto Av+B \qquad \bmod N \tag{10.30}$$

where the enciphering key $E=(A,B)$ consists of an invertible $k\times k$ matrix A with entries taken from $\mathbb{Z}_N$, and a $k\times 1$ column vector B again with entries taken from $\mathbb{Z}_N$. The variable v denotes a $k\times 1$ column vector of numbers from $\mathbb{Z}_N$ corresponding to a k-tuple of letters.

Our proof of Theorem 10.3.2 continues to hold when we are working modulo N. So the requirement that the matrix

$$A=\begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad \bmod N \tag{10.31}$$

be invertible just means that the number $ad-bc$ must be invertible modulo N. By Theorem 2.3.1 this just means that we need the integer $ad-bc$ to be coprime to the integer N.

The deciphering function is

$$f_D\colon (\mathbb{Z}_N)^k \to (\mathbb{Z}_N)^k, v \mapsto \alpha v+\beta \qquad \bmod N \tag{10.32}$$

where the enciphering key $D=(\alpha,\beta)$ consists of a $k\times k$ matrix α and a $k\times 1$ column vector β. The values of α and β are determined by noting that the algebraic derivation of (3.13) and (3.14) carries over from clock arithmetic to matrix arithmetic to yield:

$$\alpha = A^{-1} \tag{10.33}$$

$$\beta = -\alpha B \tag{10.34}$$

What is the advantage to working with k-letter message units? One advantage is that for $k\geq 3$ there is no clear contender for the most frequent k-tuple of letters in English texts or texts in other languages. So frequency analysis can't be used to determine the enciphering or deciphering key. A second advantage is that for large k it has an extremely large key space. The cryptosystem (10.30) is known as an *affine cryptosystem*. It is a secure system when $k\geq 3$. It is also easy to implement on a computer and is quick at enciphering long pieces of plaintext. The question of how to send new deciphering and enciphering keys to users is easily answered these days: use a public key cryptosystem, such as the RSA system, to exchange keys. The cryptosystem considered in Section 3.2 is just an affine system with $k=1$.

The idea of using matrices to construct cryptosystems goes back to Lester S. Hill who, in 1929 invented what is now known as the *Hill cipher*. In short, the Hill cipher is an affine cryptosystem with $k=3$ and with the vector B in the enciphering key always taken to be the zero vector. When $B=0$ we say that the cryptosystem (10.30) is a *linear cryptosystem*. With modern computing power, linear systems are now considered to be insecure.

10.5 A puzzle

What is the size of the key space for the affine cryptosystem (10.30)? This is a hard question to ask in an introductory text! Is there some restriction that can be placed on N to make the question more tractable without making it so specific that it loses its mathematical interest?