with a different key, namely, the pair $(a^{-1}, -a^{-1}b)$. (In some cryptosystems, the deciphering algorithm, as well as the key, is different from the enciphering algorithm.) We shall always suppose that the deciphering and enciphering algorithms are publicly known, and that it is the keys K_E and K_D which can be concealed.

Let us suppose that someone wishes to communicate secretly using the above affine cryptosystem $C \equiv aP + b$. We saw in § III.1 that it is not hard to break the system if one uses single-letter message units in an N-letter alphabet. It is a little more difficult to break the system if one uses digraphs, which can be regarded as symbols in an N^2-letter alphabet. It would be safer to use blocks of k letters, which have numerical equivalents in $\mathbf{Z}/N^k\mathbf{Z}$. At least for $k > 3$ it is not easy to use frequency analysis, since the number of possible k-letter blocks is very large, and one will find many that are close contenders for the title of most frequently occurring k-graph. If we want to increase k, we must be concerned about the length of time it takes to do various arithmetic tasks (the most important one being finding a^{-1} by the Euclidean algorithm) involved in setting up our keys and carrying out the necessary transformations every time we send a message or our friend at the other end deciphers a message from us. That is, it is useful to have big-O estimates for the order of magnitude of time (as the parameters increase, i.e., as the cryptosystem becomes "larger") that it takes to: encipher (knowing K_E), decipher (knowing K_D), or break the code by enciphering without knowledge of K_E or deciphering without knowledge of K_D.

In all of the examples in Chapter III — and in all of the cryptosystems used historically until about fifteen years ago — it is not really necessary to specify the deciphering key once the enciphering key (and the general algorithms) are known. Even if we are working with large numbers — such as N^k with k fairly large — it is possible to determine the deciphering key from the enciphering key using an order of magnitude of time which is roughly the same as that needed to implement the various algorithms. For example, in the case of an affine enciphering transformation of $\mathbf{Z}/N^k\mathbf{Z}$, once we know the enciphering key $K_E = (a, b)$ we can compute the deciphering key $K_D = (a^{-1} \bmod N^k, -a^{-1}b \bmod N^k)$ by the Euclidean algorithm in $O(log^3(N^k))$ bit operations.

Thus, with a traditional cryptosystem anyone who knew enough to decipher messages could, with little or no extra effort, determine the enciphering key. Indeed, it was considered naive or foolish to think that someone who had broken a cipher might nevertheless not know the enciphering key. We see this in the following passage from the autobiography of a well-known historical personality:

> Five or six weeks later, she [Madame d'Urfé] asked me if I had deciphered the manuscript which had the transmutation procedure. I told her that I had.