has made public $2^{29} = 45$, and so Bernardo can also find the key $B = 21$ by raising 45 to the b-th power (his secret exponent is $b = 19$). Of course, there is no security in working with such a small field; an outsider could easily find the discrete logarithm to the base 2 of 12 or 45 modulo 53. And in any case there is no security in using a shift encryption of single-letter message units. But this example illustrates the mechanics of the Diffie–Hellman key exchange system.

The Massey–Omura cryptosystem for message transmission. We suppose that everyone has agreed upon a finite field $\mathbf{F}_q$, which is fixed and publicly known. Each user of the system secretly selects a random integer e between 0 and $q-1$ such that $g.c.d.(e, q-1) = 1$ and, using the Euclidean algorithm, computes its inverse $d = e^{-1} \ mod \ q-1$, i.e., $de \equiv 1 \ mod \ q-1$. If user A (Alice) wants to send a message P to Bob, first she sends him the element P^{e_A}. This means nothing to Bob, who, not knowing d_A (or e_A, for that matter), cannot recover P. But, without attempting to make sense of it, he raises it to *his* e_B, and sends $P^{e_A e_B}$ back to Alice. The third step is for Alice to unravel the message part of the way by raising to the d_A-th power; because $P^{d_A e_A} = P$ (by Proposition II.1.1), this means that she returns P^{e_B} to Bob, who can read the message by raising this to the d_B-th power.

The idea behind this system is rather simple, and it can be generalized to settings where one is using other processes besides exponentiation in finite fields. However, some words of caution are in order. First of all, notice that it is absolutely necessary to use a good signature scheme along with the Massey–Omura system. Otherwise, any person C who is not supposed to know the message P could pretend to be Bob, returning to Alice $P^{e_A e_C}$; not knowing that an intruder was using his own e_C, she would proceed to raise to the d_A and make it possible for C to read the message. Thus, the message $P^{e_A e_B}$ from Bob to Alice must be accompanied by some authentification, i.e., some message in some signature scheme which only Bob could have sent.

In the second place, it is important that, after a user such as B or C has deciphered various messages P, and so knows various pairs (P, P^{e_A}), he cannot use that information to determine e_A. That is, suppose Bob could solve the discrete log problem in $\mathbf{F}_q^*$, thereby determining from P and P^{e_A} what e_A must be. In that case he could quickly compute $d_A = e_A^{-1} \ mod \ q-1$ and then intercept and read all future messages from Alice, whether intended for him or not.

The ElGamal cryptosystem. We start by fixing a very large finite field $\mathbf{F}_q$ and an element $g \in \mathbf{F}_q^*$ (preferably, but not necessarily, a generator). We suppose that we are using plaintext message units with numerical equivalents P in $\mathbf{F}_q$. Each user A randomly chooses an integer $a = a_A$, say in the range $0 < a < q-1$. This integer a is the secret deciphering key. The public enciphering key is the element $g^a \in \mathbf{F}_q$.

To send a message P to the user A, we choose an integer k at random,