and then send A the following pair of elements of $\mathbf{F}_q$:

$$(g^k,\ Pg^{ak}).$$

Notice that we can compute g^{ak} without knowing a, simply by raising g^a to the k-th power. Now A, who knows a, can recover P from this pair by raising the first element g^k to the a-th power and dividing the result into the second element (or, equivalently, raising g^k to the $(q-1-a)$-th power and multiplying by the second element). In other words, what we send A consists of a disguised form of the message — P is "wearing a mask" g^{ak} — along with a "clue," namely g^k, which can be used to take off the mask (but the clue can be used only by someone who knows a).

Someone who can solve the discrete log problem in $\mathbf{F}_q$ breaks the cryptosystem by finding the secret deciphering key a from the public enciphering key g^a. In theory, there could be a way to use knowledge of g^k and g^a to find g^{ak} — and hence break the cipher — without solving the discrete log problem. However, as we mentioned in our discussion of the Diffie–Hellman key exchange system, it is conjectured that there is no way to go from g^k and g^a to g^{ak} without essentially solving the discrete logarithm problem.

The Digital Signature Standard. In 1991 the U.S. government's National Institute of Standards and Technology (NIST) proposed a Digital Signature Standard (DSS). The role of DSS is expected to be analogous to that of the much older Data Encryption Standard (DES), i.e., it is supposed to provide a standard digital signature method for use by government and commercial organizations. But while DES is a classical ("private key") cryptosystem, in order to construct digital signatures it is necessary to use public key cryptography. NIST chose to base their signature scheme on the discrete log problem in a prime finite field. The DSS is very similar to a signature scheme that was originally proposed by Schnorr (see the references below). It is also similar to a signature scheme of ElGamal (see Exercise 9 below). We now describe how the DSS works.

To set up the scheme (in order later to be able to sign messages), each user Alice proceeds as follows: (1) she chooses a prime q of about 160 bits (to do this, she uses a random number generator and a primality test); (2) she then chooses a second prime p that is $\equiv 1\ (mod\ q)$ and has about 512 bits; (3) she chooses a generator of the unique cyclic subgroup of $\mathbf{F}_p^*$ of order q (by computing $g_0^{(p-1)/q}\ (mod\ p)$ for a random integer g_0; if this number is $\neq 1$, it will be a generator); (4) she takes a random integer x in the range $0 < x < q$ as her secret key, and sets her public key equal to $y = g^x\ (mod\ p)$.

Now suppose that Alice wants to sign a message. She first applies a hash function to her plaintext (see §1), obtaining an integer h in the range $0 < h < q$. She next picks a random integer k in the same range, computes $g^k\ (mod\ p)$, and sets r equal to the least nonnegative residue modulo q of the latter number (i.e., g^k is first computed modulo p, and the result is then