A Note on Digital Signatures

R. J. McEliece
Consultant, Department of Mathematics
University of Illinois

This article demonstrates the fact that any one-way function can be used to generate unforgeable digital signatures in a conceptually simple and easily implementable manner.

1. One of the most intriguing possibilities raised by Diffie and Hellman (Ref. 1) in their seminal paper on cryptography is that of devising digital signature systems. Until now, however, the only entirely satisfactory signature systems which have been generated using the Diffie-Hellman approach depend on the existence of a "trap door one-way permutation" – and apparently the only such function of this kind presently known is the number-theoretic mapping of Rivest, Shamir, and Adleman (Ref. 2). In this note, we will show that any one-way function (and one-way functions, in contrast to trapdoor one-way permutations, are quite easy to find) can be used to produce signature systems which are likely to be satisfactory for many applications.

2. Suppose a certain individual A wishes to transmit a sequence of n messages to another individual B. These messages will be transmitted digitally, and A wishes to be able to "sign" his messages in such a way that B will be certain that A actually was the sender. (For our purposes the actual content of the messages is unimportant; but, for example, the messages could be checks drawn against A's bank account at bank B.)

We suggest the following way for A to generate digital signatures. Let f be a "one-way" function with domain X and range Y. (This is an imprecise notion, but roughly it means that for a given $x \in X$, $f(x)$ is easy to compute, but for a given $y \in Y$, the equation $f(x) = y$ is overwhelmingly difficult to solve for x. We will give explicit examples in the next section.) Prior to the beginning of correspondence between A and B, A randomly selects n elements of X, $(x_1, x_2, \ldots, x_n)$. This list must be kept secret. He then computes $y_i = f(x_i)$, $i = 1, 2, \ldots, n$, and transmits the list $(y_1, y_2, \ldots, y_n)$ to B. This list need not be kept secret, and indeed in some circumstances it may be desirable to make it publicly known, in order to avoid later disputes. The signature attached to the ith message in the correspondence between A and B is simply x_i. B can verify that a given received signature x attached to what purports to be the ith message from A is authentic by computing $f(x)$; the message is accepted if and only if $f(x) = y_i$.

In this scheme, the actual verification is computationally simple, since by assumption, given x, $f(x)$ is easy to compute. On the other hand, a potential forger who does not know the x_i's would have to solve the equation $f(x) = y_i$ for some i in order to attach a valid signature x_i to a message of his own. But since f is assumed to be a one-way function it would be computationally infeasible for him to do this. In the next section we discuss the selection of the function f.

3. It is easy to concoct horribly complicated functions f which might be one-way functions, but what is really needed is

a function f which is *provably* one-way. We can almost, but not quite rigorously, accomplish this by invoking the theory of *NP*-completeness. We cite only two examples, but many more could be given.

Example 1. Let C by an arbitrary $N/2 \times N$ matrix with nonnegative integer entries. Let X be the set of N-dimensional vectors with entries 0 and 1. Then for $x \in X$, the function $f(x) = Cx^T$ is, by the theory of *NP*-completeness, very likely to be a one-way function, provided N and the entries of C are large enough. For in the first place the function itself is quite easy to compute – it is a simple integer matrix-vector multiplication. On the other hand, the complexity of solving the equation $Cx^T = y$ is, by the theory of *NP*-completeness, almost surely an exponential function of N. (This is because the "integer programming" problem is *NP*-complete (Ref. 3).

Example 2. Let $a = (a_1, \ldots, a_N)$ be a list of large nonnegative integers, and again let X be the set of N-dimensional 0-1 vectors. The function $f(x) = \Sigma x_i a_i$, where $x = (x_1, \ldots, x_N)$ is again likely to be one-way, because the corresponding combinatorial problem (solve $a \cdot x = y$ for x; the "knapsack" problem) is known to be *NP*-complete. Again, see Ref. 3 for details. (The knapsack problem was also used by Merkle and Hellman (Ref. 4) to design a public-key cryptosystem.)

References

1. Diffie, W., and Hellman, M., "New Directions in Cryptography," *IEEE Trans. Inform. Th.*, IT-22, pp. 144-654, 1976.
2. Rivest, R., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, 21, pp. 120-126, 1978.
3. Karp, R., "Reducibility Among Combinatorial Problems," pp. 85-103, in *Complexity of Computer Computations*, Plenum Press, New York, 1972.
4. Merkle, R., and Hellman, M., "Hiding Information and Signatures in Trapdoor Knapsacks," *IEEE Trans. Inform. Th.*, IT-24, pp. 525-530, 1978.