A Public-Key Cryptosystem Based On Algebraic Coding Theory

R. J. McEliece
Communications Systems Research Section

Using the fact that a fast decoding algorithm exists for a general Goppa code, while no such exists for a general linear code, we construct a public-key cryptosystem which appears quite secure while at the same time allowing extremely rapid data rates. This kind of cryptosystem is ideal for use in multi-user communication networks, such as those envisioned by NASA for the distribution of space-acquired data

I. Introduction

Recently, Diffie and Hellman (Ref. 3) introduced the notion of a *public-key cryptosystem* in which communication security is achieved without the need of periodic distribution of a secret key to the sender and receiver. This property makes such systems ideal for use in multi-user communication networks, such as those envisioned by NASA for the distribution of space-acquired data (Ref. 4). Later, Rivest, Shamir and Adleman (Ref. 7) explicitly exhibited such a system, using facts from number theory, and Merkle and Diffie (Ref. 6) exhibited one based on the known difficulty of the integer-packing "knapsack" problem. In this paper we propose a public key cryptosystem which is based on the theory of algebraic codes.

II. Description of the System

We base our system on the existence of *Goppa codes*. For the full theory of such codes the reader is referred to (Ref. 5, Chapter 8), but here we summarize the needed facts.

Corresponding to each irreducible polynomial of degree t over $GF(2^m)$, there exists a binary irreducible Goppa code of length $n = 2^m$, dimension $k \geq n - tm$, capable of correcting any pattern of t or fewer errors. Moreover, there exists a fast algorithm for decoding these codes. [Algorithm due to Patterson. See Ref. 5, problem 8.18. The running time is $0(nt)$].

Suppose the system designer picks a desirable value of n and t, and then randomly selects an irreducible polynomial of degree t over $GF(2^m)$. Since the probability that a randomly chosen polynomial of degree t is irreducible is about $1/t$, and since there is a fast algorithm for testing irreducibility (see Ref. 1, Chapter 6), this selection would be easy to do. Next, the system designer produces a $k \times n$ generator matrix G for the code, which could be in canonical, for example row-reduced echelon, form.

Having generated G, the system designer now "scrambles" G by selecting a random dense $k \times k$ nonsingular matrix S, and a random $n \times n$ permutation matrix P. He then computes

$G' = SGP$, which generates a linear code with the same rate and minimum distance as the code generated by G. We call G' the public generator matrix, since it will be made known to the outside world.

The system designer then publishes the following data encryption algorithm, which is to be used by anyone desiring to communicate to him in a secure fashion.

Algorithm E

Let the data to be encrypted be divided into k-bit blocks. If $\mathbf{u}$ is such a block, transmit the vector $\mathbf{x} = \mathbf{u}\, G' + \mathbf{z}$, where G' is the public generator matrix, and $\mathbf{z}$ is a locally generated random vector of length n and weight t. ▨

Having received $\mathbf{x}$, the system designer can recover $\mathbf{u}$ efficiently by using the following decryption algorithm:

Algorithm D

Compute $\mathbf{x}' = \mathbf{x}P^{-1}$, where P^{-1} is the inverse of the permutation matrix P. $\mathbf{x}'$ will then be a codeword in the Goppa code previously chosen. Using Patterson's algorithm, one then computes $\mathbf{u}S = \mathbf{u}'$. Finally, $\mathbf{u}$ is computed by $\mathbf{u} = \mathbf{u}'S^{-1}$. ▨

III. Discussion

It is clear that algorithms D and E can be implemented quite simply. What remains to be studied is the security of the system. What we need to determine, essentially, is how difficult it will be for an eavesdropper who knows G' and intercepts $\mathbf{x}$ to determine $\mathbf{u}$. It appears that an eavesdropper has two basic attacks to try; first, to try to recover G from G' and so to be able to use Patterson's algorithm. Second, he might attempt to recover $\mathbf{u}$ from $\mathbf{x}$ without learning G.

The first attack seems hopeless if n and t are large enough because there are so many possibilities for G, not to mention the possibilities for S and P.

The second attack seems perhaps more promising but the basic problem to be solved is that of decoding a more or less arbitrary (n, k) linear code in the presence of t errors. In a recent paper Berlekamp, McEliece, and van Tilborg (Ref. 2) proved that the general decoding problem for linear codes is NP – complete, so one certainly expects that if the code parameters are large enough, that this attack too will be infeasible.

In particular, suppose we chose $n = 1024 = 2^{10}$, $t = 50$; then there will be about 10^{149} possible Goppa polynomials, and an astronomical number of choices for S and P. The dimension of the code will be about $k \approx 1024 \text{-} 50 \cdot 10 \approx 524$. Hence, a brute-force approach to decoding based on comparing $\mathbf{x}$ to each codeword has a work factor of about $2^{524} \approx 10^{158}$; and a brute-force approach based on coset leaders has a work factor of about $2^{500} \approx 10^{151}$. A more promising attack is to select k of the n coordinates randomly in hope that none of the k are in error, and based on this assumption, to calculate $\mathbf{u}$. The probability of no error, however, is about

$$\left(1 - \frac{t}{n}\right)^{k},$$

and the amount of work involved in solving the k simultaneous equations in k unknown is about k^3. Hence, before finding $\mathbf{u}$ using this attack one expects a work factor of

$$k^3 \cdot \left(1 - \frac{t}{n}\right)^{-k}.$$

For $n = 1024$, $k = 524$, $t = 50$ this is about $10^{19} \approx 2^{65}$.

Of course, the above discussion proves nothing about other potential methods but it does suggest that this public key system is quite secure. One final remark: The algorithms E and D are very easy to implement using digital logic, and communication rates near 10^6 bits/second or more would be feasible. The number-theoretic system proposed by Rivest, et al. (Ref. 7) does not appear to be implementable at such speeds.

Finally we note that the decryption algorithm described in this article cannot be used as an encryption algorithm for producing unforgeable "signatures." This is because algorithm D will almost surely fail to produce any output at all unless its input is a vector within Hamming distance t of some codeword; and only a very small fraction of the 2^n possible binary vectors of length n have this property. For example, if $n = 1024$, $k = 524$, there are

$$2^{524} \cdot \sum_{k<50} \binom{1024}{k} = 2^{808.41}$$

vectors within distance 50 of a codeword. Thus, the probability that a randomly selected length 1024 vector can be decoded successfully is only about $2^{-215.59}$.

References

1. Berlekamp, E. R., *Algebraic Coding Theory*, New York, McGraw-Hill, 1968.

2. Berlekamp, E. R., McEliece, R. J., and van Tilborg, H., "On the Inherent Intractibility of Certain Coding Problems," IEEE Trans. Inform. Theory. IT-24 (1978), *in press.*

3. Diffie, W., and Hellman, M., "New Directions in Cryptography," IEEE Trans. Inform. Theory, IT-22 (1976), pp. 644-654.

4. Kurzhals, P. R., "New Directions in Space Electronics," *Astronautics and Aeronautics*, Feb. 1977, pp. 32-41.

5. McEliece, R. J., *The Theory of Information and Coding*, (Vol. 3 of *The Encyclopedia of Mathematics and Its Applications*.) Reading, Mass., Addison-Wesley, 1977.

6. Merkle, R. C., and Hellman, M. E., "Hiding Information and Receipts in Trap-Door Knapsacks," paper presented at IEEE International Symposium on Information Theory, Cornell University, October, 1977.

7. Rivest, R. L., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," Communications of the ACM 21 (1978), pp. 120-126.