subject security," *J. Ass. Comput. Mach.*, vol. 24, pp. 455–464, 1977.
[7] R. C. Merkle, "Protocols for public key cryptography," BNR Tech. Rep. Palo Alto, CA, 1980.
[8] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Comm. ACM*, vol. 2, pp. 993–999, 1978.
[9] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *J. Ass. Comput. Mach.*, vol. 27, pp. 228–234, 1980.
[10] G. J. Popek and C. S. Kline, "Encryption protocols, public key algorithms, and digital signatures in computer networks," in *Foundations of Secure Computation*, R. A. Demillo *et al.*, Eds. New York: Academic, 1978.
[11] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Comm. ACM*, vol. 21, pp. 120–126, 1978.

A Modular Approach to Key Safeguarding

CHARLES ASMUTH AND JOHN BLOOM

***Abstract*—A method is proposed for a key safeguarding scheme (threshold scheme) in which the shadows are congruence classes of a number associated with the original key. A variation of this scheme provides efficient error detection and even exposes deliberate tampering. Certain underlying similarities of this scheme with Shamir's interpolation method make it possible to incorporate these protective features in that method as well.**

I. Introduction

We CONSIDER the following problem. Given a key x, one wishes to decompose it into shadows $y_1, \cdots, y_n$, in such a way that the key x is recoverable from any r of the y_i, but essentially no information is derivable from s or any fewer y_i. (See [1], also [4].) We will refer to any method that accomplishes this as a "key safeguarding scheme." Such schemes are also called threshold schemes and have uses other than key safeguarding.

The value of such a scheme depends on a number of features. Some of these are

1) the efficiency with which keys are decomposed and recovered,
2) the sensitivity of the method to random error or deliberate tampering,
3) the relation between r, s, and n.

To have $r = s + 1$ would be best. This is the sharpest possible arrangement. However, one might consider sacrificing some of this sharpness if there were compensating improvements in some other feature, e.g., speed.

The polynomial interpolation method of Shamir [4] is one of maximum sharpness. A set of numbers $\{x_0, x_1, \cdots, x_n\}$ in some field is chosen. A polynomial P of degree $r - 1$ is constructed so that $P(x_0) = x$. The numbers $y_i = P(x_i)$ for $i = 1$ to n are the shadows. The key is recovered by evaluating a Lagrange interpolating polynomial at x_0. As we shall see, this method is somewhat sensitive to errors. Also, key recovery by the usual interpolation formula requires $O(r \log^2 r)$ operations. The modular method of this paper requires only $O(r)$ operations. It also is maximally sharp. Furthermore, it is easily modified to include the option of checking the validity of the shadows before recovery of the key.

II. The Basic Method

A set of integers $\{p, m_1 < m_2 < \cdots < m_n\}$ is chosen subject to the following:

1) $(m_i, m_j) = 1$ for $i \neq j$,
2) $(p, m_i) = 1$ for all i,
3) $\prod_{i=1}^{r} m_i > p \prod_{i=1}^{r-1} m_{n-i+1}$.

Here, as before, n denotes the number of shadows. Any r shadows will suffice for key recovery. Estimates of the density of primes show that one could easily find primes m_i to satisfy 3). To find composite m_i is still easier. Finally, let $M = \prod_{i=1}^{r} m_i$.

The decomposition process begins with the key x; we assume that $0 \leqslant x < p$. Let $y = x + Ap$ where A is an arbitrary integer subject to the condition $0 \leqslant y < M$. Then let $y_i \equiv y (\text{mod } m_i)$ be the shadows.

Manuscript received March 10, 1981; revised July 20, 1982. This paper was presented at the National Telecommunications Conference, Houston, TX, December 3, 1980.

C. Asmuth is with the Department of Mathematics, Texas A & M University, College Station, TX 77843.
J. Bloom is with Chevron Oilfield Research Corporation, La Habra, CA.

0018-9448/83/0300-0208$01.00 ©1983 IEEE

To recover x, it clearly suffices to find y. If $y_{i_1}, y_{i_2}, \cdots, y_{i_r}$ are known, then by the Chinese remainder theorem, y is known modulo $N_1 = \prod_{j=1}^{r} m_{i_j}$. As $N_1 \geqslant M$ this uniquely determines y and thus x. On the other hand, if only $r-1$ shadows were known, essentially no information about the key can be recovered. If $y_{i_1}, \cdots, y_{i_{r-1}}$ are known, then all we have is $y(\bmod N_2)$ where $N_2 = \prod_{j=1}^{r-1} m_{i_j}$. Since $M/N_2 > p$ and $(N_2, p) = 1$, the collection of numbers n_i with $n_i \equiv y(\bmod N_2)$ and $n_i \leqslant M$ cover all congruence classes mod p, with each class containing at most one more or one less n_i than any other class. Thus no useful information (even probabilistic) is available without r shadows.

III. Specific Algorithms

The construction of y and the set of shadows $\{y_1, \cdots, y_n\}$ is straightforward. One needs only to decide upon a method for selecting a random integer A in $[0, (M/p) - 1]$. The reconstruction algorithm requires some discussion.

Assume now that r shadows are to be used for key recovery. Let these be denoted $y_{i_1}, \cdots, y_{i_r}$. Let $w_j = \prod_{k \leqslant j} m_k$. Thus in theory, knowledge of $y_{i_1}, \cdots, y_{i_r}$ determines $y \bmod w_r$. We define a sequence z_j recursively; $z_1 = y_1$, $z_{j+1} = z_j + a_j w_j \equiv y_{i_{j+1}} (\bmod m_{i_{j+1}})$. Once the a_j have been found, only $r-1$ multiplications and r additions are required to find $z_r \equiv y(\bmod w_r)$. Since $w_j > M$ by arrangement, we take for y the remainder obtained by dividing z_r by w_j. Since only congruences classes of the $z_j(\bmod w_j)$ are used, it is profitable to replace z_j with its remainder $(\bmod w_j)$ whenever the value of z_j obtained is substantially larger than w_j.

To find the a_j, one notices that they satisfy $a_j \equiv (y_{i_{j+1}} - z_j) w_j^{-1} (\bmod m_{i_{j+1}})$. Once the appropriate w_j^{-1} are known, it still requires only $O(r)$ operations to recover the key. To compute w_j^{-1} using the Euclidean algorithm requires at most $O(\log m_{i_{j+1}})$ operations. This can be improved at the expense of storage space by keeping a table of integers u_i such that $u_i m_i \equiv 1(\bmod \prod_{j=1}^{i-1} m_j)$ for $i = 2, 3, \cdots, n$. Now $u_{i_{j+1}} m_{i_{j+1}} \equiv 1(\bmod w_j)$ so long division yields an integer b_j such that $u_{i_{j+1}} m_{i_{j+1}} = 1 + b_j w_j$, thus $b_j \equiv w_j^{-1} (\bmod m_{i_{j+1}})$. Hence, $3r$ operations will find all w_j^{-1}.

IV. Validity Checking

The probability of two distinct sets of r shadows yielding the same incorrect key is extremely small. Thus in the methods of [1] and [4], one can be reasonably confident of not mistaking an incorrect key for a correct one provided more than r shadows are available. However, the problem of finding a set of error-free shadows can easily be unmanageable under otherwise reasonable circumstances. Consider the case in which $n = 30$, $r = 20$, and six of the shadows are in error. The chances of a random selection of 20 shadows being error free is $\binom{24}{20} \Big/ \binom{30}{20}$ which is less than $1/2800$. This might sometimes be unacceptable.

A slight modification of the modular scheme permits checking the shadows y_i in advance and eliminating those found to be in error. The idea is to weaken condition 1) of Section II. If y_i and y_j are known and $\gcd(m_i, m_j) = q_{ij}$, then we have $y_i \equiv y_j (\bmod q_{ij})$ if both shadows are correct. A random error in y_i would change its congruence class modulo most if not all of the q_{ij}. Thus the error-free shadows would be in general agreement with each other. Those shadows in error would stand out conspicuously and be discarded.

What follows is a method of constructing the moduli m_i which will even defeat deliberate tampering up to a point. For some positive integer $k < n$, choose $\binom{n}{k}$ pairwise relatively prime integers. The integer $q_{i_1, \cdots, i_k}$ corresponds to the set $\{i_1, i_2, \cdots, i_k\}$. The modulus $m_j = \prod_{j \in \{i_1, \cdots, i_k\}} q_{i_1, \cdots, i_k}$. If condition 3) in Section II is changed to require that P times the lcm of any $r-1$ moduli is less than the lcm of any r moduli, then the procedure in Section II may be used essentially unchanged. For $k = 1$ this reduces to the basic method of that section. When $k = 2$, each pair of moduli share one q, and each q occurs in two different moduli. Thus the true value of any y_i can be uniquely determined by the other shadows. Choosing $k = 2$ should in general be adequate protection against random error.

The possibility of deliberate tampering remains when $k = 2$, but cooperation by holders of two shadows is required. The shadows y_i and y_j are classes modulo m_i and m_j which share the factor q_{ij}. This factor appears in no other modulus. From the Chinese remainder theorem, one can see that it is possible for both y_i and y_j to be altered in such a way that $y_i \equiv y_j (\bmod q_{ij})$ and both remain unchanged modulo the other moduli (which are relatively prime to q_{ij}). This would not necessarily cause a false key to be accepted as the answer, but it would hinder the recovery process somewhat.

In general, a scheme using $\binom{n}{k}$ factors to build n moduli will work to expose cooperative tampering by any group of fewer than k conspirators. This is because no factor q would be shared exclusively by fewer than k moduli.

Given a set of shadows y_i, those shadows to be used in the key recovery process can be selected in a number of ways. One possibility is to pick one or more y_i at random, presuming each of them to be error free. Each of these is a "seed" from which one attempts to grow a set of mutually compatible shadows by adding them one at a time to the set.

Alternatively, one might compute a score for each shadow equal to the number of other shadows with which it is compatible. The recovery process would then use the shadows with the highest scores.

The key recovery itself may be accomplished using the equations of Section III replacing m_i with the factors q and using the residues of the y_i modulo the various q dividing m_i. When $k > 2$ this scheme seems unmanageable due to the large number $\binom{n}{k}$ of q. Instead one should probably use the process of Section III replacing accumulated products of m_i with accumulated lcm's and the congruence relations divided through by gcd's.

V. Choosing the q

To construct a scheme with n shadows, $m_1, m_2, \cdots, m_n$, and k-fold protection, one needs $\binom{n}{k}$ numbers $q_{i_1,\cdots,i_k}$ $(1 \leqslant i_1 \leqslant \cdots \leqslant i_k \leqslant n)$ which are pairwise relatively prime. A scheme of maximum sharpness requires the following condition: for any subsets of $\{1,2,\cdots,n\}$ of the form $\{j_1,\cdots,j_r\}$ and $\{k_1,\cdots,k_{r-1}\}$, we have

$$\operatorname{lcm}\left[m_{j_1},\cdots,m_{j_r}\right] > p \operatorname{lcm}\left[m_{k_1},\cdots,m_{k_r}\right]. \tag{1}$$

To show that such a choice is even possible, consider the set of primes in the interval $[a, b]$, and pick the q from that set. To insure enough q values a rough estimate is

$$\frac{b-a}{\log b} > \binom{n}{k}. \tag{2}$$

For schemes of maximum sharpness we have $s = r - 1$. The least common multiple of s moduli equals the product of all q divided by the $\binom{n-s}{k}$ omitted ones. If $a \leqslant q_1 < q_2 < \cdots < q\binom{n}{k} \leqslant b$ then (1) is guaranteed by

$$\prod_{i \leqslant \binom{n-s}{k}} q_i > p\left(\prod_{j > \binom{n}{k} - \binom{n-r}{k}} q_j\right), \tag{3}$$

and this may in turn be replaced by the weaker inequality

$$a^{\binom{n-s}{k}} > pb^{\binom{n-r}{k}}. \tag{4}$$

Letting $a = b(1-\epsilon)$, (2) becomes $\epsilon b/\log b > \binom{n}{k}$, and (4) becomes

$$\left[\binom{n-s}{k} - \binom{n-r}{k}\right] \log b > \log p + \binom{n-r+1}{k} \log\left(\frac{1}{1-\epsilon}\right).$$

For fixed n, k, p, and r one can easily satisfy these conditions when b is large and ϵ is small.

Some comments are in order. First, while (2) is roughly correct, guaranteed theoretical bounds for numbers of primes yield inequalities such as $b/(\log b + 2) - a/(\log a - 4) > \binom{n}{k}$, or, if $[a, b]$ is disjoint from $[e^{100}, e^{500}]$, $b/\log b - a/(\log a - 2) > \binom{n}{k}$ (Rosser [3]). Since the distribution of primes is erratic and even the best theoretical estimates are poor, it is best in practice to refer to actual lists of primes. Secondly, the q need only be relatively prime. For large a and small ϵ, sets of relatively prime q twice as large as $(b-a)/\log b$ probably exist. Finally, for large a and small ϵ, little is lost in using (4). If it is desired that the q be as small as possible, then (4) is significantly stronger than necessary, and one pays for that elsewhere. A slight improvement is obtained from the estimate (3). An even larger value of p can be obtained by choosing all moduli to have both large and small factors q. Condition (1) should then be checked directly.

This improvement is illustrated by the case $n = 6$, $k = 2$, $r = 3$. Let $\{q_1, q_2, \cdots, q_{15}\} = \{25, 26, 27, 29, \cdots, 67, 71\}$, a set of 15 pairwise relatively prime integers. Condition (4) requires $p < 682$; (3) requires $p < 2012$; a scheme for forming the moduli exists which allows $p < 20503$ by (1).

One can estimate the size of the moduli for various values of k. The ratio $\log m/\log p$ is the extra work required in using a k-fold scheme. For large a and small ϵ, all factors q are roughly the same size, say Q. All moduli m are roughly the same size as $Q^{\binom{n-1}{k-1}-\binom{r}{k}}$. For any modulus m, $\log m$ is approximately $((n-1)\cdots(n-k+1)/(r)(r-1)\cdots(r-k+2))\log p$ (for $k = 1$, $\log p \doteq \log m$). For $n = 2r$ and $n \gg k$ this is roughly $\log m \doteq 2^{k-1}\log p$. Specifically, a scheme with $n = 20$, $r = 10$, $k = 3$ can be implemented with the values of the q taken to be primes between 87 037 and 99 991. Here $p \doteq 10^{215}$, and each modulus is about the size of 10^{855}. This takes approximately four times as long as an unprotected scheme with p the same size but is easily worth the time if some errors can be expected among the shadows.

VI. Generalizations

In all that has been done, one can replace the ring of integers $\boldsymbol{Z}$ by any Euclidean domain since such structures have built-in mechanisms for computing gcd's, inverses, etc. Euclidean domains other than the integers include certain rings of algebraic numbers and certain polynomial rings.

Rings of algebraic numbers do not appear to offer any special advantages over $\boldsymbol{Z}$. Using polynomial rings over a field amounts to the interpolation scheme in [4]. In particular, the value of $p(x_j)$ is just the residue to p modulo the ideal generated by $(x - x_j)$, a prime ideal in the ring of polynomials in x.

The methods of Section III may be applied to polynomials as well; using stored inverses modulo some ideal results in faster key recovery than the usual Lagrange interpolation formulas. One can also incorporate the validity checking methods of Section IV into the interpolation scheme (there the q are binomials $(x - x_j)$; the y_j are polynomials of higher degree). Thus the advantages of using $\boldsymbol{Z}$ lie in the fact that it is slightly faster and in that adding validity checking does not substantially complicate the recovery procedure.

References

[1] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. Nat. Computer Conf.* AFIPS Conf. Proc., vol. 48, 1979, pp. 313–317.

[2] ——, "One time pads are key safeguarding schemes, not cryptosystems. Fast key safeguarding schemes (threshold schemes) exist," in *Proc. 1980 Symp. Security and Privacy*.

[3] B. Rosser, "Explicit bounds for some functions of prime numbers," *Amer. J. Math.*, vol. 63, pp. 211–232, 1941.

[4] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, pp. 612–613, Nov. 1979.