time for all neighbors m of j and hence Z_j will become $(S+1)$. Since j has no nodes at hop-distance $(S+1)$, $\langle 7 \rangle$ will hold and this completes the proof of the lemma.

Lemma MH-1 a) and Lemma MH-2 a), b) are exactly Theorem MH-1 and this completes the proof of the theorem.

References

[1] R. G. Gallager, "A shortest path routing algorithm with automatic resynch," unpublished note, March 1976.

[2] A. Segall, P. M. Merlin, and R. G. Gallager, "A recoverable protocol for loop-free distributed routing," *Proc. ICC*, June 1978.

[3] S. G. Finn, "Resynch procedures and a failsafe network protocol *IEEE Trans. Comm.*, vol. COM-27, no. 6, pp. 840–846, June 1979.

[4] P. M. Merlin and A. Segall, ‘A failsafe distributed routing protocol *IEEE Trans. Comm.*, vol. COM-27, no. 9, pp. 1280–1288, Sept. 1979.

[5] P. M. Merlin, "Specification and validation of protocols," *IEEE Trans. Comm.*, vol. COM-27, no. 11, pp. 1671–1681, Nov. 1979.

[6] A. Segall, "Optimal distributed routing for virtual line-switched data networks, *IEEE Trans. Comm.*, vol. COM-27, no. 1, pp. 201–209, Jan. 1979.

[7] ——, "Advances in verifiable failsafe routing procedures, *IEEE Trans. Comm.*, vol. COM-29, no. 4, pp. 491–497, Apr. 1981.

[8] ——, "Distributed network protocols," EE Publ. 414, Dept. Elec. Eng., Technion–I.I.T., July 1981; also MIT Rep. LIDS-P-1015.

On Secret Sharing Systems

EHUD D. KARNIN, STUDENT MEMBER, IEEE, JONATHAN W. GREENE, STUDENT MEMBER, IEEE, AND MARTIN E. HELLMAN, FELLOW, IEEE

Abstract—A "secret sharing system" permits a secret to be shared among n trustees in such a way that any k of them can recover the secret, but any $k-1$ have complete uncertainty about it. A linear coding scheme for secret sharing is exhibited which subsumes the polynomial interpolation method proposed by Shamir and can also be viewed as a deterministic version of Blakley's probabilistic method. Bounds on the maximum value of n for a given k and secret size are derived for any system, linear or nonlinear. The proposed scheme achieves the lower bound which, for practical purposes, differs insignificantly from the upper bound. The scheme may be extended to protect several secrets. Methods to protect against deliberate tampering by any of the trustees are also presented.

I. Introduction

CRYPTOGRAPHY is extremely useful for making data files unintelligible to anyone who does not possess the secret key in which they were enciphered. But what happens if the legitimate owner of the file loses the key or is himself lost through incapacity or death?

There is a clear need for providing a backup copy of the key to protect against these eventualities. A safe deposit box can easily store a backup copy of the key on a punch card or similar data storage medium since most keys will be between 50 and 1000 bits long. But even a safe deposit box is vulnerable (e.g., to the "silverfish threat," named for an insect which eats punch cards), so it may be advantageous to provide multiple backup copies. To guard against simultaneous destruction, these copies should be stored in physically separated safe deposit boxes. Letting v_i denote the information stored in the ith safe deposit box and letting s denote the secret key,

$$v_1 = v_2 = \cdots = v_n = s.$$

The secret s can be recovered even if $n-1$ pieces have been destroyed, but theft of even one piece compromises the secret.

A different approach protects against the threat of theft, but aggravates the "silverfish threat." Divide the secret key s into n pieces $v_1, v_2, \cdots, v_n$ in a manner such that no information about s is learned from any $n-1$ pieces. This can be accomplished by letting v_1 to v_{n-1} be independent random variables, uniformly distributed over $\boldsymbol{S}$, the set of all possible secret keys, and letting

$$v_n = s + (v_1 + v_2 + \cdots + v_{n-1}) (\mathrm{mod}\ q),$$

where $q = |\boldsymbol{S}|$ is the cardinality of $\boldsymbol{S}$. As a small example, when $q = 2$, v_n is the exclusive or of s with v_1 through v_{n-1}. While a 1-bit key is of no value, the technique can be applied to successive bits of the key.

The advantage of this method is also a disadvantage: if even one of the v_i is destroyed, the legitimate owner is unable to reconstruct the key from the remaining backup information.

Motivated by a desire to protect against both threats, several researchers have investigated the following secret sharing problem.

Manuscript received February 13, 1981; revised February 25, 1982. This work was supported in part by the Joint Services Electronics Program under Contract DAAG29-81-0057 and in part by the National Science Foundation under Contract ENG-10173. This paper was presented at the International Symposium on Information Theory, Santa Monica, CA, February 9–12, 1981.

The authors are with the Department of Electrical Engineering, Stanford University, Stanford, CA, 94305.

Divide a secret s into n pieces $v_1, v_2, \cdots, v_n$, each chosen from a set V, such that the following conditions are satisfied.

C1) The secret s is recoverable from any k pieces ($k \leq n$).

C2) Knowledge of $k-1$ or fewer pieces provides absolutely no information about s.

If the system is not to involve data expansion we also require:

C3) $|\boldsymbol{V}| \leq |\boldsymbol{S}|$. That is, each piece v_i is to be no longer than s.

Any such system will be referred to as a "k-out-of-n secret sharing system." The technique applies not only to protecting backup copies of a key in safe deposit boxes, but also to sharing any secret among n trustees in such a way that any k of them can reconstruct the secret, but any $k-1$ or fewer of them cannot learn anything about it.

Restating these requirements using the notation of information theory we have for any set of k indices $\{i_1, i_2, \cdots, i_k\}$:

$$H(s \mid v_{i_1}, v_{i_2}, \cdots, v_{i_k}) = 0 \tag{1}$$

and

$$H(s \mid v_{i_1}, v_{i_2}, \cdots, v_{i_{k-1}}) = H(s). \tag{2}$$

Theorem 1: For conditions C1) and C2) to hold it is necessary that

$$H(v_i) \geq H(s), \qquad i = 1, 2, \cdots, n.$$

Note: If s is uniformly distributed over $\boldsymbol{S}$ then Theorem 1 implies that $|\boldsymbol{V}| \geq |\boldsymbol{S}|$, and condition C3) can be replaced by $|\boldsymbol{V}| = |\boldsymbol{S}|$. For an arbitrary distribution C3) and Theorem 1 imply that

$$H(v_i) = H(s), \qquad i = 1, 2, \cdots, n. \tag{3}$$

Proof: We have

$$H(v_{i_k}) \geq I(s; v_{i_k} \mid v_{i_1}, \cdots, v_{i_{k-1}})$$

because a random variable cannot provide more information than it has uncertainty. From (1) and (2) above we see that $I(s; v_{i_k} \mid v_{i_1}, \cdots, v_{i_{k-1}}) = H(s)$. Combining the last two expressions and replacing the dummy variable i_k by i complete the proof. Q.E.D.

Blakley [1] was the first to publish an approach to solving the secret sharing problem. His is a probabilistic approach based on linear projective geometry. Each v_i specifies a hyperplane and the secret s is the (hopefully) unique point of intersection of the n hyperplanes. Blakley satisfies C1) by specifying more hyperplanes than are needed. He gives a probabilistic argument which indicates that C2) should also hold.

Shamir [2] has also published a solution, and bases his approach on the observation that any k distinct points (x, y) suffice to determine the coefficients of the $(k-1)$st degree polynomial

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}. \tag{4}$$

By giving each trustee a different (x, y) point, $v_i = (i, y(i))$, and by setting $a_0 = y(0) = s$, Shamir is able to satisfy condition C1). By dealing in a finite field $\boldsymbol{F}$ with $|\boldsymbol{F}| = |\boldsymbol{S}|$, he is able to satisfy condition C2) as well.

Because there are only $|\boldsymbol{F}|$ distinct x values and $s = y(0)$, Shamir's system is limited to a number of trustees (or pieces)

$$n \leq |\boldsymbol{F}| - 1. \tag{5}$$

Otherwise one of the v_i would have to equal s, or two of the v_i would be mathematically equivalent. Either way C1) or C2) would fail to hold. (The only exception is when $k = 1$. Then each trustee knows the whole secret, each v_i is equal to s, and n can be as large as desired, independent of $|\boldsymbol{F}|$.)

The approach to secret sharing developed in Section II of this paper can be viewed as a deterministic version of Blakley's approach and includes Shamir's method as a special case.

Section III establishes upper and lower bounds on the maximum value of n for given values of k and $|\boldsymbol{S}|$, with any system, linear or nonlinear.

Section IV generalizes C1) and C2) to the situation where there are l secrets $s_1, s_2, \cdots, s_l$ to be protected, and it is required that for any $1 \leq j \leq l$ and for any set of k indices $\{i_1, i_2, \cdots, i_k\}$,

$$H(s_j \mid v_{i_1}, v_{i_2}, \cdots, v_{i_k}) = 0 \tag{6}$$

and

$$H(s_j \mid v_{i_1}, v_{i_2}, \cdots, v_{i_{k-1}}) = H(s_j). \tag{7}$$

That is any k trustees can determine all l secrets perfectly, but any $k-1$ trustees have no information about any particular secret. (The $k-1$ trustees might have significant information about two or more secrets taken as a pair.)

This generalization has the advantage of allowing several secrets to be protected with the same amount of data as is usually needed to protect one secret by itself. Or a large secret might be partitioned into l pieces and these protected with a smaller amount of data than is needed to protect the entire secret.

In Section V we show that for some parameter values our system is computationally more efficient than previous techniques.

Section VI discusses the problem of detecting tampering by one of the trustees. A solution based on one-way functions is suggested.

II. Requirements for a Secret Sharing System

Our method for secret sharing grew out of the following probabilistic reasoning: Let $\boldsymbol{u}$ be a totally random 300-bit binary row vector. Let each of the v_i be a 100-bit binary row vector consisting of 100 random parity checks on $\boldsymbol{u}$. That is, each parity check bit is the exclusive or of a random subset of the 300 bits of $\boldsymbol{u}$.

Then any three v_i are related to $\boldsymbol{u}$ by a random 300-by-300 binary matrix. As shown in [3], approximately 29 percent of the n-by-n binary matrices are nonsingular over GF(2), the binary field, and most of the rest have small rank defect. It is therefore likely that any three $\boldsymbol{v}_i$ allow $\boldsymbol{u}$ to be reconstructed nearly perfectly. Yet any two $\boldsymbol{v}_i$ must leave at least 100 bits of uncertainty about $\boldsymbol{u}$: 200 bits of information (the two $\boldsymbol{v}_i$) must leave at least 100 bits of uncertainty about the 300-bit vector $\boldsymbol{u}$.

This would give rise to a "three-out-of-n secret sharing system," except for two facts:

1) Condition C1) is not exactly met because $\boldsymbol{u}$ is not always perfectly reconstructable from three of the $\boldsymbol{v}_i$;
2) Condition C2) is not met. Any two pieces $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ leave approximately 100 bits of uncertainty about $\boldsymbol{u}$, but $\boldsymbol{u}$ is 300 bits long.

Before treating these problems, we note that the method generalizes to any "k-out-of-n" system and to any secret size. For example, if one desires a 7-out-of-10 system with at least 40 bits of uncertainty if six or fewer pieces are known, then make $\boldsymbol{u}$ ($7 \times 40 = 280$)-bits long, and let each $\boldsymbol{v}_i$ be 40 random parity checks on $\boldsymbol{u}$.

To meet condition C1) we will use deterministic codes rather than a random coding argument. But, unlike many other areas of information theory, the random coding argument is applicable in practice because the effort required to decode a randomly chosen code is reasonable, involving only a matrix inversion. Contrast this with the effort required to decode a random, linear error correcting code [5].

Meeting condition C2) is exactly the "key distillation" problem solved in [3] and [4]. In key distillation, one is given a partially uncertain random variable ($\boldsymbol{u}$ in the above discussion) and must find a new, totally uncertain random variable (the secret $\boldsymbol{s}$) which is a function of $\boldsymbol{u}$. Further, the uncertainty of the new random variable must be *the same* as that of the partially uncertain variable. The new variable has all of the uncertainty of the original variable, but in concentrated or "distilled" form. Put in terms of the secret sharing problem, we require

$$H(\boldsymbol{s} \mid \boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \cdots, \boldsymbol{v}_{i_{k-1}}) = H(\boldsymbol{u} \mid \boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \cdots, \boldsymbol{v}_{i_{k-1}}) \quad (8)$$

and

$$H(\boldsymbol{u} \mid \boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \cdots, \boldsymbol{v}_{i_{k-1}}) = H(\boldsymbol{s}). \quad (9)$$

In [3] and [4] it is shown that random, linear projections of $\boldsymbol{u}$ often have the desired distillation property. Each of the $\boldsymbol{v}_i$ is also a random linear projection of $\boldsymbol{u}$, so we can consider $\boldsymbol{s}$ as a new, $(n+1)$st "piece" of $\boldsymbol{u}$ and denote it by $\boldsymbol{v}_0$. If the key distillation is perfect, $\boldsymbol{s}$ is the same length as each $\boldsymbol{v}_i$.

Theorem 2: Associating $\boldsymbol{s}$ with $\boldsymbol{v}_0$, and generalizing from GF(2) to any finite field GF(q), the problem of finding a secret sharing system of the form

$$\boldsymbol{v}_i = \boldsymbol{u}_i A_i \quad (10)$$

is equivalent to the following.

Find a set of $n+1$ matrices over GF(q), $\{A_0, A_1, A_2, \cdots, A_n\}$, each of dimension km-by-m, such that every set of k of the A_i has full rank, km. (m is the secret size and k is the number of pieces required to reconstruct the secret. The dimension of $\boldsymbol{u}$ is km.)

Note: Although we will usually deal with q being prime, this theorem and all subsequent statements are correct when q is a power of a prime.

Example: Here is a 2-out-of-4 system with a 2-bit secret $\boldsymbol{s}$:

$$A_0 = \begin{bmatrix} 10 \\ 01 \\ 00 \\ 00 \end{bmatrix} \quad A_1 = \begin{bmatrix} 00 \\ 00 \\ 10 \\ 01 \end{bmatrix} \quad A_2 = \begin{bmatrix} 10 \\ 11 \\ 10 \\ 01 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 01 \\ 10 \\ 10 \\ 01 \end{bmatrix} \quad A_4 = \begin{bmatrix} 10 \\ 01 \\ 11 \\ 01 \end{bmatrix}.$$

In this example $\boldsymbol{s} = (u_1, u_2)$, $\boldsymbol{v}_1 = (u_3, u_4)$, $\boldsymbol{v}_2 = (u_1 + u_2 + u_3, u_2 + u_4)$, etc. It is not hard to check that any single trustee knows nothing about $\boldsymbol{s}$ and that any two trustees can solve for $\boldsymbol{s} = (u_1, u_2)$.

Proof of Theorem 2: When $\boldsymbol{s}$ is the secret value to be protected, choose $\boldsymbol{u}$ uniformly at random from $U(\boldsymbol{s})$, the set of vectors which satisfy $\boldsymbol{s} = \boldsymbol{u}A_0$. For all $\boldsymbol{s}$, the cardinality of $U(\boldsymbol{s})$ is $q^{(k-1)m}$ so

$$H(\boldsymbol{u}) = H(\boldsymbol{s}) + (k-1)m \cdot \log(q). \quad (11)$$

First we show that if every set of k of the A_i has full rank then conditions C1) and C2) are met.

It is easiest to deal with condition C1) first. If k trustees collaborate they can reconstruct $\boldsymbol{u}$ because, by the full-rank assumption, their pooled information is related to $\boldsymbol{u}$ by an invertible matrix. Knowing $\boldsymbol{u}$ they can then compute $\boldsymbol{s} = \boldsymbol{u}A_0$. (The A_i are public information.)

Condition C2) is established by noting that if $k-1$ trustees were to collaborate and were also told the secret, they could reconstruct $\boldsymbol{u}$ perfectly. This follows from the full-rank assumption, using A_0 as one of the matrices. Therefore

$$H(\boldsymbol{u} \mid \boldsymbol{s}, \boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \cdots, \boldsymbol{v}_{i_{k-1}}) = 0$$

so

$$H(\boldsymbol{u} \mid \boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \cdots, \boldsymbol{v}_{i_{k-1}}) = I(\boldsymbol{u}; \boldsymbol{s} \mid \boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \cdots, \boldsymbol{v}_{i_{k-1}}). \quad (12)$$

Then

$$\begin{aligned} H(\boldsymbol{s} \mid \boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \cdots, \boldsymbol{v}_{i_{k-1}}) &\geq I(\boldsymbol{u}; \boldsymbol{s} \mid \boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \cdots, \boldsymbol{v}_{i_{k-1}}) \\ &= H(\boldsymbol{u} \mid \boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \cdots, \boldsymbol{v}_{i_{k-1}}) \\ &\geq H(\boldsymbol{u}) - (k-1)H(\boldsymbol{v}_i) \\ &\geq H(\boldsymbol{u}) - (k-1)m \cdot \log(q) \\ &= H(\boldsymbol{s}), \end{aligned}$$

which is the desired result. The first inequality just states that the information provided by a random variable is at most equal to its uncertainty. The next equality follows from (12) above. The next inequalities follow from $H(\boldsymbol{u} \mid \boldsymbol{v} \geq H(\boldsymbol{u}) - H(\boldsymbol{v})$ and $H(\boldsymbol{v}_i) \leq \text{length}(\boldsymbol{v}_i) = m \cdot \log(q)$. The final equality follows from (11). Thus condition C2) is met as well.

We now show the converse: If there is a set of k of the A_i with a dependence then either condition C1) or condition C2) must be violated. Because the distribution on $\boldsymbol{s}$ is arbitrary, and we are seeking a counter example, we can assume $\boldsymbol{s}$ is uniformly distributed and $H(\boldsymbol{s}) = m \cdot \log(q)$.

If any columns of A_0 are involved in the dependent subset, then there is a set of $k-1$ trustees who can reconstruct part of $\boldsymbol{s}$ (the bits corresponding to the dependent columns), in violation of C2). If the dependent subset does not involve A_0 then at least one bit known by some trustee is also known if a subset of $k-1$ other trustees were to collaborate. But these $k-1$ trustees have $m \cdot \log(q)$ bits of uncertainty about $\boldsymbol{s}$ by condition C2), and when the kth trustee joined them he could not reduce the uncertainty by $m \cdot \log(q)$, the length of his piece of the secret, because at least one bit of it was already known by the dependence assumption. There would then be a subset of k trustees who could not reconstruct $\boldsymbol{s}$, violating condition C1). Q.E.D.

Theorem 3: The secret $\boldsymbol{s}$ can be taken to be the first m components of $\boldsymbol{u}$ without loss of generality. Further, $\boldsymbol{v}_1$ can be taken to be the next m components of $\boldsymbol{u}$, $\boldsymbol{v}_2$ can be taken to be the next m components of $\boldsymbol{u}, \cdots,$ and $\boldsymbol{v}_{k-1}$ can be taken to be the last m components of $\boldsymbol{u}$.

Note: The example following Theorem 2 is of this form.

Proof: Given a secret sharing system using matrices $\{A_0, A_1, \cdots, A_n\}$, let T be the invertible km-by-km matrix consisting of $[A_0, A_1, \cdots, A_{k-1}]$. ($T$ must have full rank by Theorem 2). Letting

$$A_i' = T^{-1}A_i, \qquad i = 0, 1, 2, \cdots, n,$$

we then have A_i' of the desired form for $i = 0, 1, \cdots, k-1$. The new secret sharing system has the same properties as the old one, because they are related by the invertible transformation T. Equivalently, we have defined a new vector $\boldsymbol{u}' = \boldsymbol{u}T$ and the A_i' matrices give the $\boldsymbol{v}_i$ in terms of $\boldsymbol{u}'$ instead of in terms of $\boldsymbol{u}$. That is,

$$\boldsymbol{v}_i = \boldsymbol{u}A_i = \boldsymbol{u}TT^{-1}A_i = \boldsymbol{u}'A_i'. \qquad \text{Q.E.D.}$$

Up to this point we considered the secret as a string of m symbols over GF(q). An alternative description might be: Let s be a one component (scalar) secret over GF(q^m). Then $\boldsymbol{u}$ is a k-dimensional row vector, the matrices $\{A_i\}$ become k-dimensional column vectors $\{\boldsymbol{a}_i\}$ and v_i is the inner (dot) product of $\boldsymbol{a}_i$ with $\boldsymbol{u}$.

Going from a scalar system in GF(q^m) to a vector system in GF(q) is always possible because a linear relation of the form $y = ax$, where y, a, and x are all in GF(q^m), is also a (matrix) linear relation between m-dimensional vectors in GF(q). While the converse is not true, the fairly tight upper and lower bounds we shall derive in Section III on the maximum value of n, for a given k, are the same for both types of systems. The upper and lower bounds will be shown to practically coincide, hence in what follows we shall mainly treat the scalar systems over GF(q^m). Applying Theorem 2, and with this restriction, we can restate the problem of finding a k-out-of-n secret sharing system as follows.

Problem Statement: Find a (k-by-$n+1$)-dimensional matrix G whose $n+1$ columns form a set of k-dimensional vectors, $\{\boldsymbol{a}_0, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_n\}$, over GF($q^m$) such that any k of the columns are linearly independent. Theorem 3 allows us to assume without loss of generality that G has a k-by-k identity matrix for its first k columns, i.e., is in a systematic form.

In this framework,

$$\boldsymbol{v} = \boldsymbol{u}G \tag{13}$$

with $v_0 = u_1 = s$, the secret to be protected. The similarity between (10) and the encoding relation for a linear (parity check) code is evident. G is the generator matrix for a linear code which can correct any $(n+1)-k$ erasures (k-out-of-n trustees can reconstruct $\boldsymbol{u}$ and thence s), but which provides no information about the first component of $\boldsymbol{u}$ when there are $(n+1)-(k-1)$ or more erasures ($k-1$ trustees know nothing about s).

While the second requirement is a somewhat unusual for a code, results from coding theory are useful in this context. In Section III we use an upper bound on the minimum distance of nonlinear codes to show that nonlinear and linear secret sharing systems obey the same upper bound on $n_{\max}$, the maximum number of trustees, for given values of $|\boldsymbol{S}|$ and k.

Generalizing from linear to nonlinear secret sharing systems, we allow the secret s to be any function of a random variable $u \in \boldsymbol{U}$, and each trustee is given a subset $\boldsymbol{V}_i$ of $\boldsymbol{U}$ in which u lies. Letting $\boldsymbol{S}_i$ be the image of $\boldsymbol{V}_i$ under the mapping from u to s, conditions C1), C2), and C3) become the following.

C1′) The intersection of any k subsets $\boldsymbol{S}_i$ is the same single point, namely the secret s.

C2′) The intersection of any $k-1$ subsets $\boldsymbol{S}_i$ is the entire space $\boldsymbol{S}$.

C3′) For each i, $|\boldsymbol{V}_i| \leq |\boldsymbol{S}|$.

III. Bounds on the Maximum Value of n

We now address the following question: Given a secret set of cardinality $|\boldsymbol{S}|$ and k, the number of trustees required to recover the secret, what is the maximum number of trustees, $n_{\max}$? The next theorems give bounds for $n_{\max}$, and thus also demonstrate the existence of all types of secret sharing systems described in the previous section.

Theorem 4: Given $|\boldsymbol{S}| = q^m$ and k, a one component secret sharing system of the form $\boldsymbol{v} = \boldsymbol{u}G$ has

$$q^m \leq n_{\max} \leq q^m + k - 2, \qquad q^m > k, \tag{14}$$

$$n_{\max} = k, \qquad q^m \leq k. \tag{15}$$

Proof: We first show the lower bound of (14). Let α be a primitive element of GF(q^m) and denote the identity of the field by 1. The following matrix,

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & \alpha & \alpha^2 & \cdots & \alpha^{q^m-1} \\ 0 & 0 & \alpha^2 & \alpha^4 & \cdots & \alpha^{(q^m-1)2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & \alpha^{k-1} & \alpha^{2(k-1)} & \cdots & \alpha^{(q^m-1)(k-1)} \end{bmatrix} \tag{16}$$

is a k-by-$q^m + 1$ matrix with any k columns linearly independent. This is because the determinant of the associated k-by-k matrix is a Vandermonde determinant (or a product of nonzero elements by such a determinant), and $\alpha, \alpha^2, \cdots, \alpha^{q^m-1}$ are all different elements of GF(q^m). Hence we can always find a matrix G with number of columns

$$(n+1) \geq q^m + 1$$

or

$$n_{\max} \geq q^m.$$

To upper bound $n_{\max}$ suppose we have found a satisfactory k-by-$(n+1)$ matrix G. By Theorem 3, we may assume without loss of generality that G is in systematic form, namely

$$G = (I \mid P), \tag{17}$$

where I is a k-by-k identity matrix and P is k by $(n+1-k)$. We claim that every j-by-j minor of P, $j = 1, 2, \cdots, \min(k, n+1-k)$, is nonzero. To see this pick any j columns of P and any $k-j$ columns of I. The resulting matrix should be nonsingular, which implies that the j-by-j minor is nonzero. In particular all the entries of P must be nonzero.

Without introducing any dependencies among the columns of G, we can divide each row by its first entry in P, and then divide each column by its first entry. We thus obtain a matrix of the form:

$$(I \mid P) = \left[\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 0 & 1 & X & \cdots & X \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 & X & \cdots & X \end{array}\right]. \tag{18}$$

Consider now a two-by-two minor of P formed by two entries from the first row and two entries from the second. Since it is nonzero, the two entries from the second row must be different. This is true for any two entries, so all $n+1-k$ entries of the second row must be different. There are only $q^m - 1$ different nonzero elements in GF(q^m); hence

$$n + 1 - k \leq q^m - 1$$

which implies the upper bound in (14).

Similarly, since all two-by-two minors from the first and second columns of P are nonzero, all entries in the second column must be different. Since there are k entries, this is possible only if $k \leq q^m - 1$. Otherwise, i.e., for $k \geq q^m$, P has no second column and so

$$n + 1 - k = 1$$

which implies (15). Q.E.D.

Note 1: Suppose one chooses G as in (16), deleting the second column. Then the components of $\boldsymbol{v}$ in $\boldsymbol{v} = \boldsymbol{u}G$ can also be evaluated as

$$v_i = D(\alpha^i), \qquad i = 1, 2, \cdots, n,$$

where

$$D(x) = u_1 + u_2 x + u_3 x^2 + \cdots + u_k x^{k-1}$$

and $v_0 = u_1 = s$. But this is exactly Shamir's polynomial interpolation scheme (compare with (4)), which is thus shown to be included as a special case of our method.

Note 2: Over any finite field GF(q^m) the k-by-$(k+1)$ matrix

$$G = \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 \end{bmatrix}$$

exists, and has the desired properties for a "k-out-of-k secret sharing system." (This is the system mentioned in the introduction as a means of protecting against theft while aggravating the "silverfish threat.") This achieves the upper bound (15) which applies to "small secrets," i.e., $q^m \leq k$. Note that polynomial interpolation schemes cannot generate $k = n$ secret sharing systems when $q^m \leq k$, because of the restriction (5).

Corollary 1: Given $|S| = q^m$ and k, the bounds (14) and (15) apply to an m-component secret sharing system of the form $\boldsymbol{v}_i = \boldsymbol{u}A_i$ where entries are over GF(q). (See description in Theorem 2.)

Proof: The lower bound in (11) is obvious, since existence of a k-by-n matrix G over GF(q^m) implies existence of an appropriate km-by-nm matrix over GF(q) (as discussed in Section II). In analogy to what we have done while proving Theorem 4, G can be represented as the following block matrix:

$$(I \mid P) = \left[\begin{array}{cccc|cccc} I_m & 0 & \cdots & 0 & I_m & I_m & \cdots & I_m \\ 0 & I_m & \cdots & 0 & I_m & X & \cdots & X \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & I_m & I_m & X & \cdots & X \end{array}\right], \tag{19}$$

where I_m is an m-by-m identity matrix, and X stands for an m-by-m matrix. Pick any $2m$-by-$2m$ submatrix of P consisting of four m-by-m blocks of the first and second block rows. This $2m$-by-$2m$ matrix should be nonsingular, and in particular the first and the $(m+1)$st columns should differ. Hence all first (m component) columns of the blocks in the second block row of G are distinct. Since they must also be nonzero, their number is bounded by $q^m - 1$, which implies (14).

A similar argument applied to the first columns of the m-by-m blocks in the second block-column of P yields (15). Q.E.D.

The difference between the upper and lower bounds we established in (14) is $k - 2$, which is negligible with respect to $n_{\max}$. Even for a very small secret of only 20 bits, $n_{\max} > 10^6$ is obtained, which far exceeds any reasonable k.

Next we show that even the most general secret sharing system does not yield a substantial increase in $n_{\max}$.

Theorem 5: Any linear or nonlinear secret sharing system which satisfies C1)–C3) (see Section II) has

$$n_{\max} \leq |S| + k - 2. \tag{20}$$

Proof: Each trustee is given a symbol v_i which is an element of an alphabet having at most $|\mathbf{S}|$ letters, because of C3). Consider

$$\boldsymbol{v} = (v_1, v_2, \cdots, v_n)$$

as a codeword with n components. We would like to establish that the distance of this code is $n - k + 1$.

By C1) $n - k$ erasures still enable us to recover s. What we must show is that not only s, but the whole codeword can be restored, i.e.,

$$H(v_{i_{k+1}} | v_{i_1}, v_{i_2}, \cdots, v_{i_k}) = 0, \tag{21}$$

for any $(k + 1)$st piece.

First we notice that

$$\begin{aligned} H(v_{i_{k+1}} | v_{i_1}, \cdots, v_{i_k}) &\leq H(v_{i_{k+1}}, s | v_{i_1}, \cdots, v_{i_k}) \\ &= H(v_{i_{k+1}} | s, v_{i_1}, \cdots, v_{i_k}) \\ &\quad + H(s | v_{i_1}, \cdots, v_{i_k}) \\ &= H(v_{i_{k+1}} | s, v_{i_1}, \cdots, v_{i_k}), \end{aligned}$$

where the last equality follows from (1). The right-hand side can be bounded further by

$$H(v_{i_{k+1}} | s, v_{i_1}, \cdots, v_{i_{k-1}}, v_{i_k}) \leq H(v_{i_{k+1}} | s, v_{i_1}, \cdots, v_{i_{k-1}}),$$

hence to prove (21) it suffices to show that

$$H(v_{i_k} | s, v_{i_1}, \cdots, v_{i_{k-1}}) = 0. \tag{22}$$

We rewrite (1) and (2) as

$$H(s, v_{i_1}, \cdots, v_{i_{k-1}}, v_{i_k}) - H(v_{i_1}, \cdots, v_{i_{k-1}}, v_{i_k}) = 0$$

and

$$H(s, v_{i_1}, \cdots, v_{i_{k-1}}) - H(v_{i_1}, \cdots, v_{i_{k-1}}) = H(s).$$

Subtracting these equations we obtain

$$H(v_{i_k} | s, v_{i_1}, \cdots, v_{i_{k-1}}) - H(v_{i_k} | v_{i_1}, \cdots, v_{i_{k-1}}) = -H(s).$$

Since

$$H(v_{i_k} | v_{i_1}, \cdots, v_{i_{k-1}}) \leq H(v_{i_k}) \leq H(s),$$

where the last inequality is due to C3) or (3), we get

$$H(v_{i_k} | s, v_{i_1}, \cdots, v_{i_{k-1}}) \leq 0,$$

which establishes (22) and thus (21).

Obviously the code cannot correct more than $n - k$ erasures, because this would violate C2). Hence we have shown that any secret sharing system which satisfies C1)–C3) is also a code of distance $n - k + 1$, i.e., it is a maximum distance (n, k) code.

Maximum distance codes (not necessarily linear) were first studied by Singleton [5], and n was shown there to be bounded by

$$n_{\max} \leq |\mathbf{S}| + k - 1.$$

Since one of the symbols is the secret itself, we get (20). Q.E.D.

IV. Protecting More than One Secret

Suppose we have a set of $n + 1$ matrices over GF(q), $\{A_0, A_1, \cdots, A_n\}$, each of dimension km-by-m, and any k of the A_i have full rank. Then as we have shown in Theorem 2, $\boldsymbol{v}_i = \boldsymbol{u}A_i$ is a k-out-of-n secret sharing system, where $\boldsymbol{s} = \boldsymbol{v}_0$ and $\boldsymbol{v}_i$, $i = 1, 2, \cdots, n$, is the information available to the ith trustee. It is clear that there is no difference in kind between $\boldsymbol{v}_0$ and any other projection $\boldsymbol{v}_i$ of $\boldsymbol{u}$. Hence each $\boldsymbol{v}_i$ can be considered equally well as a secret as long as it is not given to one of the trustees.

We conclude that as many as $(n_{\max} + 1) - n$ projections can be kept secret in the following sense.

Any k trustees can recover all the projections, and any $k - 1$ trustees have absolutely no information about any *particular* projection. (Obviously $k - 1$ trustees know as much as $(k - 1)m \cdot \log(q)$ bits about $\boldsymbol{u}$, thus having significant information about the set of secrets when considered as one entity.)

As an application of these facts consider a k-out-of-n secret sharing system which can protect l secrets when $1 \leq l \leq k$. This can be accomplished by letting

$$\boldsymbol{u} = (\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_{l-1}, \boldsymbol{u}_l, \cdots, \boldsymbol{u}_{k-1}) \tag{23}$$

and

$$A_i = [0 \cdots 0 I_m 0 \cdots 0]^T, \qquad 0 \leq i \leq l - 1, \tag{24}$$

with I_m, the m-by-m identity matrix as the $(i + 1)$st block. $\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_{l-1}$ are the secrets to be protected, and $\boldsymbol{u}_l, \cdots, \boldsymbol{u}_{k-1}$, is a string of $(k - l)m$ independent random variables each drawn according to a uniform distribution on GF(q). We recall that $\{A_0, A_1, \cdots, A_n\}$ can always be chosen so that (24) is satisfied, due to Theorem 3.

Alternatively, one may use this procedure to protect a large secret by dividing smaller pieces among the trustees. If $m \cdot \log(q)$ bits of uncertainty seems adequate, a secret $\boldsymbol{s}$ of length lm can be protected; just let

$$(\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_{l-1}) = \boldsymbol{s}$$

and choose $\boldsymbol{u}$ and A_i as in (23) and (24).

As an example, a 1000-bit secret can be protected by giving each of the n ($n \geq 10$) trustees just 100 bits. Any nine of them still have 100 bits of uncertainty about each segment of s.

V. Computational Aspects

This section analyzes the computational requirements of our secret sharing system. When the k trustees collaborate to recover the secret, they have to solve k linear equations with k unknowns, (we consider the one component case in $\mathrm{GF}(q^m)$). For an arbitrary A_i, such as in [1], this takes $O(k^3)$ operations, while Shamir's polynomial interpolation method [2] requires $O(k \cdot \log^2 k)$ operations.

However, there are cases for which our method is more efficient. Suppose we wish to protect one secret and "silverfish" are rare; i.e., $n - k$ can be a small number. By choosing G as a systematic matrix with $n + 1$ columns, the worst case is when the collaborating k trustees correspond to the last k columns of G. The resulting matrix may be reduced to the form

$$\begin{bmatrix} I & Y \\ 0 & X \end{bmatrix}$$

merely by interchanging rows and columns. I is an identity matrix, and the dimensions of X are $(n + 1 - k)$ by $(n + 1 - k)$. Hence it takes only $O((n - k + 1)^3)$ operations (in the worst case) to recover the secret.

VI. Detecting Deliberate Tampering

Suppose one or more of the trustees deliberately changes his piece of information. If this trustee is one of the k who collaborate to recover the secret, an erroneous value of s results. Unfortunately, within the framework described so far, this is inevitable as the next theorem explains.

Theorem 6: No secret sharing system which satisfies conditions C1)–C3) can detect tampering.

Proof: As discussed in the proof of Theorem 5, a k-out-of-n secret sharing system which satisfies C1)–C3) is an (n, k) code, with minimum distance $d = n - k + 1$. By a basic theorem in coding theory (see e.g., [7, pp. 140–141]) such a code can correct c errors and detect t errors if and only if

$$c + t \leq d - 1.$$

(A corrected error is counted as a detected error, too.) A detected error is equivalent to an erasure, or the absence of a trustee in our case. Allowing $n - k$ trustees to be absent means

$$t = n - k = d - 1.$$

Hence further detection (or correction) of errors, due to deliberate tampering or other reasons, is impossible. Q.E.D.

We can clearly detect and even correct tampering by increasing the number of required trustees beyond k which was originally required, but this violates C1), C2).

We suggest a different approach to handle the tampering problem. Let F be a one-way function [8], and let

$$s' = F(s). \tag{25}$$

The image of the secret, s', as well as the function F are public. Once k trustees compute the secret, they can check the image of their result against s' and thus detect tampering.

We implicitly assume here that the secret is uniformly distributed in a set $\boldsymbol{S}$, where $|\boldsymbol{S}|$ is a large number. (Or alternatively, $H(s)$ is large.) Otherwise an exhaustive search over all $|\boldsymbol{S}|$ possible values is feasible and compromises s.

As an example, a 56-bit secret may be applied to the key port of a data encryption standard (DES) [8] device, while the plaintext and the resulting ciphertext are public. Clearly using a one-way function changes the status of the secret sharing system from unconditionally secure to computationally secure only.

References

[1] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. Amer. Fed. Inform. Proc. Soc.* 1979 *NCC*, vol. 48, pp. 313–317, June 1979.

[2] A. Shamir, "How to share a secret," in *Comm. Assoc. Comput. Mach.*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[3] A. Carleial and M. E. Hellman, "A note on Wyner's wiretap channel," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 387–390, May 1977.

[4] R. Kahn, "Privacy in multiuser information theory," Ph.D. dissertation, Dept. Electrical Engineering, Stanford University, Stanford, CA, 1979.

[5] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, "On the inherent intractability of certain coding schemes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 384–386, May 1978.

[6] R. C. Singleton, "Maximum distance q-nary codes," *IEEE Trans. Inform. Theory*, vol. IT-10, pp. 116–118, April 1964.

[7] R. J. McEliece, "The theory of information and coding," in *The Encyclopedia of Mathematics and its Applications*, vol. 3. Reading, MA: Addison-Wesley, 1977.

[8] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 644–654, Nov. 1976.