THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING
ASPECTS OF ELECTRICAL COMMUNICATION

Volume 53 March 1974 Number 3

Codes Which Detect Deception

By E. N. GILBERT, Mrs. F. J. MacWILLIAMS, and N. J. A. SLOANE

(Manuscript received May 15, 1973)

We consider a new kind of coding problem, which has applications in a variety of situations. A message x is to be encoded using a key m to form an encrypted message $y = \Phi(x, m)$, which is then supplied to a user G. G knows m and so can calculate x. It is desired to choose $\Phi(\cdot, \cdot)$ so as to protect G against B, who knows x, y, and $\Phi(\cdot, \cdot)$ (but not m); B may substitute a false message y' for y. It is shown that if the key can take K values, then an optimal strategy for B secures him a probability of an undetected substitution $\geqq K^{-\frac{1}{2}}$. Several encoding functions $\Phi(\cdot, \cdot)$ are given, some of which achieve this bound.

I. INTRODUCTION

The gambling casino has often supplied a vivid and concrete setting for problems in probability theory,[1] stochastic processes,[2] hypothesis testing,[3] information theory,[4] and coding theory,[5] and we shall use it to describe our problem.

There are two main participants, the owner of the casino G (standing for good guy) and the manager B (the bad guy). B has been reporting the daily takings from the slot machines to be less than they actually are and keeping the difference for himself. To prevent this, G proposes to install in each slot machine a key generator of which he possesses an exact duplicate and an encoder which will encrypt the day's takings x using a key m to produce an encrypted message

$$y = \Phi(x, m). \tag{1}$$

(See Figs. 1 and 2.) The device will punch y onto a paper tape. At suitable intervals B will mail the tape to G, who will calculate x from y and m. From time to time G will visit the casino to change the key generator. We assume that B knows x and $\Phi(\cdot,\cdot)$ (but cannot change them), and y (which he can change), but does not know m. G knows y, m, and $\Phi(\cdot,\cdot)$.

If B attempts to give G a false message y_o', there may be no x' satisfying $y_o' = \Phi(x', m)$, and then G will discover B's deception. But if B can solve (1) for m, then he can successfully substitute a false message x' by giving G the correctly encrypted message $y' = \Phi(x', m)$. The problem is to design $\Phi(\cdot,\cdot)$ so as to make it as difficult as possible for B to deceive G without being caught.

Clearly, the problem is applicable to other situations (vending machines, cash registers, etc.) and in fact was first presented to us by G. J. Simmons of Sandia Corporation in connection with monitoring the production of certain materials in the interests of arms limitation.

The problem resembles the one normally encountered in cryptography in that a key m is used to encrypt a clear text x into an encoded form $y = \Phi(x, m)$. But there is an important difference. Since B knows x already, many of the standard cryptographic codes would allow B to recover the key m.

To prevent B from using (1) to learn the key, G must construct $\Phi(\cdot,\cdot)$ so that (1) has several solutions m. Then B will probably pick a wrong key m_o and G will discover that B's encrypted mesage y_o' is incompatible with the correct key. As one might expect, to provide many solutions to (1) G must use a large number K of possible keys.

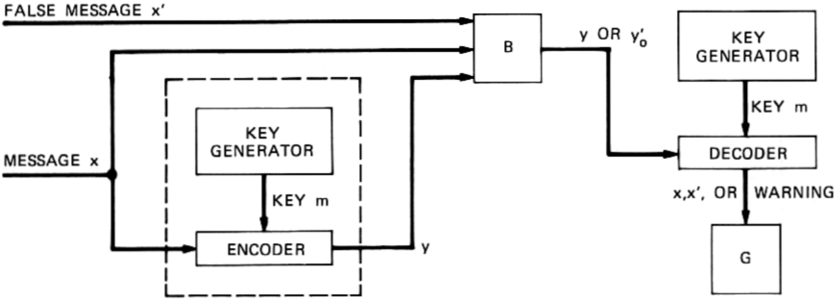

Fig. 1—Encoding to detect substitution.

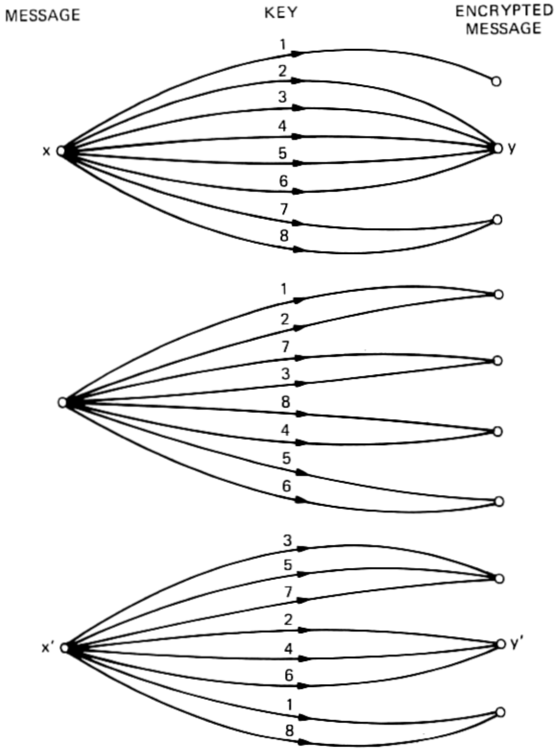

Fig. 2—Diagram of a code.

When B tries to substitute a false message, his probability of escaping detection will be called p_o. The probability p_o for an optimal B strategy will be called p_o^*. We will show that $p_o^* \geqq K^{-\frac{1}{2}}$. Although Section IV will construct a code which is best-possible in the sense of achieving $p_o^* = K^{-\frac{1}{2}}$, this equality can be achieved only by severely restricting the number N of possible messages x. More useful codes must compromise among three conflicting goals for G: small p_o^*, small K, and large N. We give two such codes, one random (Section VII) and one systematic (Section VIII).

Throughout most of this paper we imagine that B has a particular, but unknown, false message x' to substitute for x. We assume that x is equally likely to be any one of the N possibilities and that B picks x' at random from the remaining $N - 1$ messages. Then p_o is an average of the probabilities $p_o(x, y, x')$ of success when B substitutes a given x' for given x, knowing y.

In Section IX, B uses a different strategy. There B is content to succeed in *any* deception. Given x and y, B now picks x' to maximize the chance of escaping detection. Merely keeping p_o small does not protect G against this if individual terms $p_o(x, y, x')$ are large. With proper design, the systematic code of Section VIII still defeats B.

II. THE AUTHENTICATOR

A convenient special form for the encryption (1) is

$$y = (x; z), \tag{2}$$

i.e., y is the clear text x followed by a string z of extra digits or letters. Here z is some function of x and m. G will use z to test the received message y for authenticity. For this reason z will be called an *authenticator*.

Although (2) is a special case of (1), nothing is lost by restricting the encryption to this special form. Indeed, if some other $\Phi_o(x, m)$ in (1) provides a good code, one can always create a code of the form (2) by taking $z = \Phi_o(x, y)$, i.e.,

$$y = \Phi(x, m) = [x; \Phi_o(x, m)].$$

Including x as part of y cannot help B; he knows x already. Giving x to G explicitly cannot hinder him in detecting a deception by B. Thus the new code is at least as good for G as the old one.

Whether or not to use a code of the form (2) is purely a matter of convenience. However, the form (2) has a special property which we can now require without loss for all codes. It is that different clear text messages x_1, x_2 cannot be encoded into the same y, i.e.,

$$\Phi(m_1, x_1) \neq \Phi(m_2, x_2) \tag{3}$$

holds for all m_1, m_2 if $x_1 \neq x_2$. Then a typical code has a *diagram* like Fig. 2 which portrays clear messages x as points in the left column and encrypted messages y as points in the right column. The lines directed from left to right are labeled by the key names $1, \cdots, K$ to show how these keys encode each x into a y. Because of (3) the encrypted messages y fall into disjoint clusters, each cluster containing all possible images of a particular x.

III. PROBABILITY OF DECEPTION

B successfully deceives G with probability $p_o \geqq K^{-1}$ just by guessing a key m_o at random with all K keys equally likely. Better strategies use B's knowledge of x and y to restrict his guess to keys satisfying (1).

Usually B need not guess $m_o = m$, the correct key. B still succeeds if

$$\Phi(x', m_o) = \Phi(x', m). \tag{4}$$

In Fig. 2, B would pick m_o to be one of 2, 3, 4, 5, or 6; if $m = 2$ then the guesses $m_o = 2$, 4, or 6 all succeed.

An important qualitative feature of a code is the size of the bundle of lines leading from the message x to the encrypted message y in the code diagram (Fig. 2). G must make these bundles large enough to prevent B from guessing m with high probability. But if the bundles are too large, B will succeed often because many keys m_o satisfy (4). In compromising between the two extreme bundle sizes, G cannot limit B to a probability $p_o = 1/K$. In fact, we now show that B can always use a strategy which succeeds with probability

$$p_o \geqq K^{-\frac{1}{2}}. \tag{5}$$

In order to prove (5) we will have to place some natural restrictions on the behavior of G and B.

(*a*) B does not attempt to deceive G by replacing x by $x' = x$. If we allowed B that kind of "deception," B could succeed with probability $p_o = 1$ and (5) would be a weak result.

(*b*) All N messages x are equally likely. Although this requirement could be relaxed, some condition like it must be imposed to forbid G from using one particular message x_1 almost exclusively. In that case G could let all keys encrypt x_1 to the same y_1 but give all other messages x' K distinct encrypted forms. B would then have $p_o < K^{-\frac{1}{2}}$ but G would receive little information from each message.

(*c*) Another restriction on G might be that he use the K keys at random, equally likely and independent of x. We won't need this restriction on G to prove (5). If G uses the keys in any other way he only helps B increase p_o.

(*d*) We will prove that (5) holds even if B picks x' at random from the $N - 1$ messages different from x, all equally likely. This only strengthens (5) because there may be better strategies for B.

Knowing how the message x, x', and key m are distributed, we can compute the joint probability $P(x, y, x')$. This probability is the weight used in averaging $p_o(x, y, x')$ to get

$$p_o = \sum_{x,y,x'} P(x, y, x')p_o(x, y, x'), \tag{6}$$

as mentioned in Section I. The probability $p_o(x, y, x')$, that B succeeds in substituting x', knowing x and y, depends on how B uses x, y, x' to determine a false encrypted message y_o'. B knows the function $\Phi(\cdot,\cdot)$ and the key distribution. From these, he can compute the conditional probability distribution $P(y'|x, y, x')$ of the correctly encrypted false message $y' = \Phi(x', m)$. B maximizes his chance of success by using a false message y_o' which maximizes $P(y'|x, y, x')$. Then B achieves

$$p_o(x, y, x') = \operatorname*{Max}_{y'} P(y'|x, y, x') \tag{7}$$

and maximizes p_o in (6). Since (7) is optimal for B we give the corresponding p_o value a special name p_o^*.

As a preliminary to (5) we now relate p_o^* to the average uncertainty U which B has about the correctly encrypted false message y'. U is a conditional entropy

$$\begin{aligned} U &= H(y'|x, y, x') \\ &= -\sum_{x,y,x',y'} P(x, y, x', y') \log P(y'|x, y, x'). \end{aligned} \tag{8}$$

Lemma: If B chooses y_o' to make (7) hold, then

$$p_o = p_o^* \geqq 2^{-U}. \tag{9}$$

Equality holds in (9) if and only if all the possible encrypted messages y' for each (x, y, x') having $P(x, y, x) \neq 0$ are equally likely and there are exactly 2^U such y'.

The proof does not require restrictions (a), (b), (c), or (d). Use (7) to write $P(y'|x, y, x') \leqq p_o(x, y, x')$ in (8). Sum on y' and use the convexity of the function $-\log p$ to get

$$\begin{aligned} U &\geqq -\sum_{x,y,x'} P(x, y, x') \log p_o(x, y, x') \\ &\geqq -\log \sum_{x,y,x'} P(x, y, x') p_o(x, y, x'). \end{aligned}$$

Now (9) follows from (6).

The derivation used two inequalities. Both must become equalities if equality holds in (9). $P(y'|x, y, x') = p_o(x, y, x')$ requires all possible y' to be equally likely for given x, y, x'. In the convexity argument, equality requires all $-\log p_o(x, y, x')$ terms to be equal to U.

We now bound p_o^* in terms of the uncertainty $H(m)$ associated with the choice of key.

Theorem 1: Suppose (7) and restrictions (a), (b), (d) all hold. Then

$$p_o = p_o^* \geqq 2^{-\frac{1}{2}H(m)}. \tag{10}$$

First note that y' is determined by $y' = \Phi(m, x')$ if m, x' are known. Then y' contains less information than (m, x'):

$$U = H(y'|x, y, x') \leqq H(m, x'|x, y, x') = H(m|x, y, x'). \tag{11}$$

But the conditional probability for m given x, y, x' depends only on x, y, so (11) becomes

$$U \leqq H(m|x, y). \tag{12}$$

Also,

$$H(m) \geqq H(m|x) = H(m, y|x) = H(y|x) + H(m|x, y)$$

so (12) provides

$$U \leqq H(m) - H(y|x). \tag{13}$$

But

$$U = H(y'|x, y, x') \leqq H(y'|x').$$

Because of constraint (d), x' is equally likely to be any one of the N messages. Then, by (b), x and x' have the same distribution, $H(y'|x') = H(y|x)$, and finally

$$U \leqq H(y|x). \tag{14}$$

Now compare (13) and (14). If $H(y|x) \leqq \frac{1}{2}H(m)$, then $U \leqq \frac{1}{2}H(m)$ follows from (14). If $H(y|x) \geqq \frac{1}{2}H(m)$, then $U \leqq \frac{1}{2}H(m)$ follows from (13). In either case, (10) follows from the lemma.

Remark: The bound (10) implies (5), and in fact reduces to (5) when restriction (c) holds.

IV. PROJECTIVE PLANE CODES

Since p_o^* is the largest probability of success obtainable by B, a code for which equality holds in (10) guarantees G the minimum p_o against optimal behavior by B. This section designs such a code. We now assume that G behaves according to (c) of Section III, for that will make

$$p_o^* = K^{-\frac{1}{2}}.$$

If equality is to hold in (10), all the inequalities used in proving Theorem 1 must become equalities. We now review these inequalities to obtain requirements on the code.

The requirements are most easily stated in terms of the bundles of keys in the code diagram, Fig. 2.

(*i*) Every pair of bundles, from x_1 to y_1 and x_2 to y_2, with $x_2 \neq x_1$, have exactly one key in common.

(*ii*) Every bundle contains $K^{\frac{1}{2}}$ keys.

(*iii*) There are $K^{\frac{1}{2}}$ bundles at each x.

To prove (*i*), (*ii*), (*iii*), begin with (11) and write $H(y'|x, y, x') = H(m, x'|x, y, x')$. If, for some x, y, x', more than one key m satisfied $y' = \Phi(m, x')$ then there would be more conditional uncertainty about the pair (m, x') than about y'. Thus equality in (11) requires

(*i'*) Every pair of bundles, from x_1 to y_1 and x_2 to y_2, $x_2 \neq x_1$, have at most one key in common.

Equality in (9) requires that the keys in any bundle from x to y be distributed equally over $2^U = 2^{\frac{1}{2}H(m)} = K^{\frac{1}{2}}$ images y' of any x'. Each of these keys leads from x' to a different y' [by (*i'*)]. Then the bundle x to y has $K^{\frac{1}{2}}$ keys, which proves (*ii*). Now (*iii*) follows from (*ii*) because there are only K keys. Requirements (*ii*) and (*iii*) also guarantee $H(y|x) = \frac{1}{2}\log K = \frac{1}{2}H(m)$, which is needed for equality in (13) and (14).

To strengthen (*i'*) to (*i*) consider the $K^{\frac{1}{2}}$ bundles leaving x and the $K^{\frac{1}{2}}$ bundles leaving x'. There are $K^{\frac{1}{2}} \cdot K^{\frac{1}{2}} = K$ pairs of bundles. (*i'*) permits each pair to have at most one key in common. But each key is common to some pair. Since there are K keys, (*i*) must hold.

One can find trivial codes which satisfy (*i*), (*ii*), (*iii*) but which have only a few messages x. For instance, the K keys might be arranged in a $K^{\frac{1}{2}} \times K^{\frac{1}{2}}$ square matrix and each row (or column) be designated as the bundle for a distinct encrypted form of x_1 (or x_2). Since this code has $N = 2$ it is not very useful. In order to force N to be large we need another requirement.

Since (*i*) requires a pair (m_1, m_2) of different keys to belong to at most one bundle, the number of pairs of keys having a common bundle is $N\binom{K^{\frac{1}{2}}}{2}$. This number must be no greater than the unrestricted number of pairs of keys $\binom{K}{2}$, so that

$$\tfrac{1}{2}NK^{\frac{1}{2}}(K^{\frac{1}{2}} - 1) \leqq \tfrac{1}{2}K(K - 1)$$
$$N \leqq K^{\frac{1}{2}} + 1. \qquad (15)$$

The condition for equality in (15) is

(*iv*) Every pair (m_1, m_2) of different keys belongs to exactly one common bundle.

We now add requirement (*iv*) in order to have a code with the largest possible N. Note that even for this code (15) indicates only about half as many message bits as key bits.

A code satisfying (*i*), (*ii*), (*iii*), (*iv*) can be constructed from any finite projective plane. Recall that a projective plane is a set of points and lines in which:

(*v*) Each pair of different lines has a unique point in common, and
(*vi*) Each pair of different points belongs to a unique line.

The most easily visualized projective plane is an infinite one based on the surface of a sphere. The lines and points of this projective plane are the great circles and pairs of diametrically opposite points on the sphere. A well-known technique (see Refs. 6, 7) uses a Galois field $GF(q)$, where q is a prime power, to construct a projective plane having $q^2 + q + 1$ points and $q^2 + q + 1$ lines.

The code will be obtained by using certain points and lines of a projective plane as the names of messages, keys, and bundles. First pick any line S to serve a special role. Using the sphere as a model, we call S the *equator*. Points on the equator will represent messages x. Points not on the equator will represent keys m. Lines other than the equator represent encrypted messages y (bundles). Each x and m determines a unique line (not S because it contains m) which we use as the name of y in (1).

Figure 3 shows the projective plane constructed from $GF(2)$. It has $2^2 + 2 + 1 = 7$ points. Six of the seven lines are shown as straight lines and the seventh, which we may take as the equator S, is a circle.

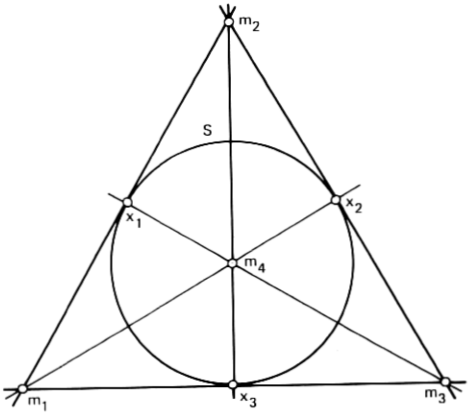

Fig. 3—A projective plane.

The three points on S are the messages and the remaining four are keys. The six straight lines are bundles, containing two keys each.

One can easily verify (*i*) and (*iv*) using (*v*) and (*vi*). Moreover, in the projective plane based on $GF(q)$, $q + 1$ lines pass through each point and $q + 1$ points lie on each line. Each line different from S contains one message x and q keys.

Each x lies on S and on q other lines. Then (*ii*) and (*iii*) hold if

$$q = K^{\frac{1}{2}}. \tag{16}$$

The equator contains $N = 1 + q = 1 + K^{\frac{1}{2}}$ points, as we expect from (15), and (*iv*) holds. When G uses this code, B will know that m is one of q keys on the line y. For any $x' \neq x$, these keys lie on q different lines through x' and B has $p_o^* = 1/q = K^{-\frac{1}{2}}$.

A Galois field $GF(q)$ exists if and only if q is a power of a prime, $q = p^n$. Then (16) requires K to be an even power of a prime: $K = p^{2n}$ in this design.

V. IMPLEMENTATION

This section simplifies the code of Section IV into a form that is easily realized by a logic circuit.

The usual construction for a projective plane begins by defining the points as vectors, having three components taken from $GF(q)$. Two vectors $\mathbf{v}_1$, $\mathbf{v}_2$ are regarded as two names for the same point if they differ only by a scalar multiple, i.e., if $\mathbf{v}_2 = \alpha\mathbf{v}_1$ for some $\alpha \in GF(q)$. The zero vector (0, 0, 0) is not used as a point. Lines are sets of points satisfying a linear homogeneous constraint. A line L can then be described by a nonzero vector $\mathbf{L} = (a, b, c)$ with the understanding that the points on L are the vectors $\mathbf{v} = (r, s, t)$ satisfying

$$\mathbf{L}\cdot\mathbf{v} = ar + bs + ct = 0.$$

Take the equator to be the line specified by the vector $\mathbf{S} = (0, 0, 1)$. Then messages x are points having third coordinate zero. By applying appropriate scalar multipliers, each x can be written either as (0, 1, 0) or as $(1, s, 0)$ with $s \in GF(q)$. The remaining points, which can be written in the standard form $(i, j, 1)$, are the q^2 keys.

To make the logic circuit as simple as possible we agree not to use (0, 1, 0) as a message. There remain $N = q = K^{\frac{1}{2}}$ messages, all of the form $(1, s, 0)$. The q lines through $(1, s, 0)$ all have vectors $(-s, 1, c)$ where

$$si - j = c \tag{17}$$

holds for all keys $(i, j, 1)$ on the line.

Only a single element s of $GF(q)$ need be transmitted to specify the vector $(1, s, 0)$ and hence x. Likewise, the key input in Fig. 1 requires only the pair (i, j). The encrypted message y [a line with vector $(-s, 1, c)$] can be transmitted just as a pair (s, c). That amounts to using c as an authenticator z. The encoder is a computer which uses (17) to produce the authenticator value c from the inputs s, i, j. G uses a similar computer to test that his received s, c and known i, j satisfy (17).

For example, the code obtained from the projective plane of Fig. 3 is:

message	*key*	*encrypted message*
0	00 or 01	00
	10 or 11	01
1	00 or 11	10
	01 or 10	11

Again the code obtained from the projective plane with 13 points based on $GF(3) = \{0, 1, 2\}$ is:

message	*key*	*encrypted message*
0	00, 01, 02	00
	10, 11, 12	01
	20, 21, 22	02
1	00, 12, 21	10
	01, 10, 22	11
	02, 11, 20	12
2	00, 11, 22	20
	02, 10, 21	21
	01, 12, 20	22

Tables for constructing larger Galois fields will be found in Refs. 8, 9, 10, and circuits for doing arithmetic in these fields in Refs. 10, 11, 12. A field $GF(2^b)$ is convenient if the message originates in binary form. Then x and z each consist of b binary digits while $2b$ digits (b for i and b for j) are required for the key.

VI. BLOCK DESIGNS

Projective planes are special cases of more complicated structures called balanced incomplete block designs (BIBD). The technique used in Section IV generalizes directly to produce new codes based on

BIBD's. The new codes do not achieve $p_o^* = K^{-\frac{1}{2}}$, but they provide good solutions for some new values of K not of the form p^{2n}.

A (b, v, r, k, λ) *BIBD* is another system of points and sets of points. The sets are now called *blocks* instead of lines. There are v points in total and each block contains exactly k points. Each point belongs to r blocks and each pair of points is a subset of λ blocks. These conditions determine the number b of blocks. For $bk = vr$ and $r(k-1) = \lambda(v-1)$ must hold in a BIBD (Ref. 6, p. 96; Ref. 7, p. 100).

Examples:

(1) The projective plane formed from $GF(q)$ (see Section IV): $b = v = q^2 + q + 1$, $r = k = q + 1$, $\lambda = 1$.

(2) The affine plane formed from $GF(q)$ (Ref. 7, p. 176): $b = q^2 + q$, $v = q^2$, $r = q + 1$, $k = q$, $\lambda = 1$.

(3) Many other examples are known: see, for example, Refs. 6, 7, 13, 14, and recent volumes of the journals *Sankhya*, *Annals of Mathematical Statistics*, and the *Journal of Combinatorial Theory*.

Given any BIBD with $\lambda = 1$, we may form a code as follows. Proceeding as in Section IV, we select a particular block S to serve as the "equator." Points on S will represent messages x. Points not on S will represent keys m. Blocks other than the equator represent encrypted messages y (bundles). Each x and m determines a unique block different from S which we use as the name of the y in (1).

There are $N = k$ messages, $K = v - k$ keys, $b - 1$ encrypted messages, and $k - 1$ keys per bundle. Since $\lambda = 1$, the $k - 1$ keys in the bundle from x to y belong to distinct bundles leaving x'. Then $p_o^* = 1/(k-1) = 1/(N-1)$.

When the BIBD is a projective plane these formulas become again $K = q^2$, $N = 1 + K^{\frac{1}{2}}$, and $p_o^* = K^{-\frac{1}{2}}$. For affine planes $K = q^2 - q$, $N = q < 1 + K^{\frac{1}{2}}$, and $p_o^* = 1/(q-1) > K^{-\frac{1}{2}}$. Thus, for given K, the affine plane has both smaller N and larger p_o^* than one would expect from the projective plane. The larger p_o^* should be expected since (*ii*), (*iii*) fail.

To have (*ii*), (*iii*) hold, r and k should be as close as possible. In most known BIBD's other than the projective and affine planes, r and k are considerably different. For example, consider the BIBD with parameters $b = 195$, $v = 91$, $r = 15$, $k = 7$, $\lambda = 1$ (number 111 in Hall's list[7]). The code obtained from this design has $K = 84$ keys, $N = 7$ messages, and $p_o^* = \frac{1}{6}$. For comparison, the projective plane code based on $GF(9)$ is superior on all counts, having $K = 81$, $N = 10$, and $p_o^* = \frac{1}{9}$.

VII. RANDOM CODES

The projective plane code in Section IV obtains $p_o^* = K^{-\frac{1}{2}}$, the smallest possible value, but it has only $N = 1 + K^{\frac{1}{2}}$ messages. Codes with $N \gg K$ have more interest. To see how large the corresponding p_o^* might be, this section examines a code constructed at random. Now N can be made as large as desired. The main result will be that p_o^* still need not exceed $K^{-\frac{1}{2}}$ by a large factor.

The random code will have one free parameter A. Each x is allowed A possible encoded forms y. For each of the K keys the y in (1) is chosen at random from the A possibilities, all equally likely. The K choices are made independently. It may well happen that one of the A possibilities is never chosen in the K trials. In that case the code diagram, Fig. 2, will show fewer than A bundles from x. The code has a p_o^* which depends on the random choices. We will look for the expected value $E(p_o^*)$. Specific codes, with the given N and K and having p_o^* less than this expectation, surely exist.

All the data about $\Phi(\cdot, \cdot)$ that B needs when substituting x' for x are contained in a table showing how the encrypted messages y, y' depend on the key m. Figure 4 shows a convenient table as an $A \times A$ array of cells, each cell containing a list of all keys which determine a (y, y') pair. Figure 4 corresponds to the pair of messages labeled x, x' in Fig. 2. Let $\nu(y, y')$ be the number of keys in the (y, y') cell.

Knowing y, B examines the corresponding column in Fig. 4. Since the K keys are equally likely,

$$P(y' \mid x, y, x') = \nu(y, y') / \sum_{y_1} \nu(y, y_1). \tag{18}$$

The optimal strategy, by which B achieves (7), is to pick y_o' to maximize $\nu(y, y')$. In Fig. 4 the row y_o' intersects the y column in a cell with the largest number of keys. There may be $k > 1$ such cells in the y column, in which case B may as well pick one of the k rows equally likely, at random.

$E(p_o^*)$ can now be described as the solution to a distribution problem. Imagine that the correct key is key #1 and that it occupies the cell in column 1 and row 1. Distribute the $K - 1$ remaining keys at random

		y	
		3,5	7
y′		2,4,6	
	1		8

Fig. 4—Table of keys.

over the A^2 cells. Let $p_{n,k}$ be the probability that the (1, 1) cell contains $\nu(1, 1) = n$ keys, that $k - 1$ other cells in column 1 contains n keys, and that moreover all of the $A - k$ remaining cells in column 1 contain fewer than n keys. Then

$$E(p_o^*) = \sum_{n,k} k^{-1} p_{n,k} \tag{19}$$

is the probability that B picks the first row for y_o'.

The exact formula for $p_{n,k}$ is cumbersome. It is not hard to simulate the distribution experiment on a computer in order to estimate $E(p_o^*)$ when K is less than a few hundred. This has been done, but only as a check on the simpler approximate calculation which follows.

When A is large, each key has a small probability A^{-2} of belonging to the cell (y, y'). After a large number $K - 1$ of independent trials, the number $\nu(y, y')$ of keys in the cell will have approximately a Poisson distribution with mean

$$\lambda = (K - 1)/A^2. \tag{20}$$

Accordingly, we treat numbers $\nu(y, y')$ as independent Poisson random variables with mean λ. The number $\nu(1, 1)$ is special because we started the distribution by placing key #1 in cell (1, 1); $\nu(1, 1) - 1$ is the Poisson variable for this cell. Poisson approximation has the disadvantage that the total number of keys $\sum_{y,y'} \nu(y, y')$ is itself a random variable. However, the mean number of keys is K and there is high probability that there will be close to K keys if K is large. The effect of this approximation should be worse for small K than for large K. The Poisson approximation and the simulation do give the same $E(p_o^*)$ to within a few percent even for $K = 25$.

Table I — $E(p_0^*)$ for random designs

$\lambda =$	$\frac{1}{16}$	$\frac{1}{4}$	1	4	16	$K^{-\frac{1}{2}}$
$K = 25$		0.47	0.44	0.54		0.2
64	0.46	0.34	0.32	0.38	0.57	0.125
100	0.40	0.29	0.27	0.32	0.46	0.1
256	0.27	0.21	0.19	0.22	0.32	0.06
400	0.23	0.17	0.16	0.18	0.26	0.05
1,024	0.15	0.12	0.11	0.12		0.03
4,096	0.087	0.069	0.062	0.068	0.092	0.015
10,000	0.062	0.047	0.042	0.046	0.061	0.01
40,000	0.036	0.026	0.023	0.024	0.032	0.005
100,000	0.025	0.018	0.015	0.016	0.021	0.003
1,045,576	0.0084	0.0063	0.0054	0.0055	0.0069	0.001

To simplify writing an expression for $p_{n,k}$, let b_n and B_n denote the probabilities that a Poisson random variable has value exactly n or at most n.

$$b_n = \lambda^n e^{-n}/n!$$
$$B_n = b_0 + b_1 + \cdots + b_n.$$

Then

$$p_{n,k} = b_{n-1} b_n^{k-1} B_{n-1}^{A-k} \binom{A-1}{k-1}. \tag{21}$$

In (21), b_{n-1} is the probability that cell (1, 1) contains n keys, $b_n^{k-1} B_{n-1}^{A-k}$ is the probability that a particular set of $k-1$ other cells have n keys but all $A-k$ others have $n-1$ keys or less, and the binomial coefficient counts the different sets of $k-1$ cells. Now insert (21) into (19) and sum on k to get

$$E(p_o^*) = \sum_{n=1}^{\infty} (n/\lambda A)\{B_n^A - B_{n-1}^A\}. \tag{22}$$

Table I gives values of $E(p_o^*)$, computed from (22). For fixed K, a broad minimum of $E(p_o^*)$ occurs near $\lambda = 1$. Then (20) shows that the minimum occurs when $A = K^{\frac{1}{3}}$, approximately. Thus, even when G designs his code by random means, he should pick A to make (*ii*) and (*iii*) of Section IV hold as nearly as possible.

Although (22) is only an approximate solution to the problem, it is also a generating function for the exact solution. Let $e(K)$ denote the exact expected value of p_o^* when the number of keys is K. Instead of $e(K)$, eq. (22) provides

$$\sum_K \frac{(\lambda A^2)^{K-1}}{(K-1)!} \exp(-\lambda A^2) e(K),$$

i.e., a sum of terms $e(K)$ weighted by the probability that the Poisson experiment produces $K-1$ keys in addition to key #1. In principle, one could multiply the sum in (22) by $\exp(\lambda A^2)$, expand the result into a series in powers of λ, and identify the coefficient of λ^{K-1} as $A^{2(K-1)} e(K)/(K-1)!$. The result for $e(K)$ is unpleasant and (22) is accurate enough. In an experiment to estimate $e(64)$, 2000 trials were made for each of $\lambda = \frac{1}{4}$, 1, 4. The fractions of trials in which B succeeded were 0.31, 0.30, 0.37.

VIII. SYSTEMATIC CODES

This section constructs a systematic code with large N by means of another generalization of the projective plane code of Section IV.

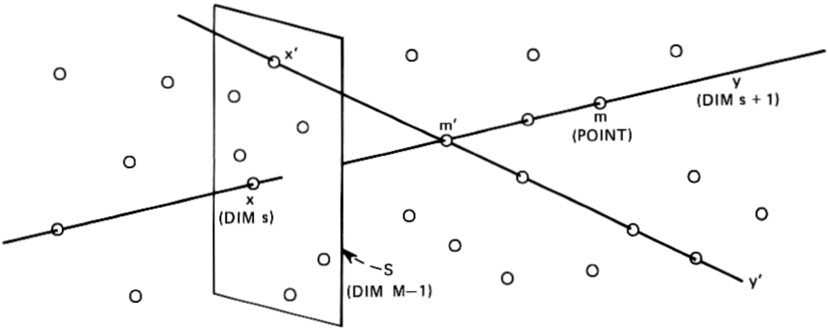

Fig. 5—Code designed from projective space of dimension M.

Unlike the random code, which had N as a free parameter, this code will specify a particular N. That disadvantage is offset by a smaller value of $E(p_o^*)$ and by a more important advantage discussed in Section IX.

Figure 5 will illustrate the code design. Given a field $GF(q)$, one can construct a projective space $PG(M, q)$ of dimension M in which points are again equivalence classes of nonzero vectors, now having $M + 1$ components. $M = 3$ in Fig. 5. The number of points is

$$f(M) = (q^{M+1} - 1)/(q - 1) = 1 + q + \cdots + q^M. \qquad (23)$$

Each set of points satisfying a system of $M - D$ independent linear homogeneous equations is a D-dimensional subspace $PG(D, q)$ containing $f(D)$ of the points of $PG(M, q)$. The number of D-dimensional subspaces of $PG(M, q)$ is[15]

$$\begin{aligned} g(D, M) &= \frac{f(M)f(M-1)\cdots f(M-D)}{f(D)f(D-1)\cdots f(0)} \\ &= \frac{(q^{M+1}-1)(q^M-1)\cdots(q^{M+1-D}-1)}{(q^{D+1}-1)(q^D-1)\cdots(q-1)}. \end{aligned} \qquad (24)$$

Proceeding as in Sections IV and VI, we again select a particular subspace S of dimension $M - 1$ to serve as the "equator." In Fig. 5, S is a projective plane. We again identify messages x with subspaces of S. But now S has subspaces of dimension $0, 1, \cdots, M - 2$ and so we can specify the dimension s of the messages as another parameter of the design. In Fig. 5, $s = 0$; another code might use $s = 1$. Given M, s, the number of distinct messages is

$$N = g(s, M - 1). \qquad (25)$$

Again, the points not in S will be keys. There are

$$K = f(M) - f(M - 1) = q^M \tag{26}$$

keys.

The key m (a point) and message x (of dimension s) determine a unique $(s + 1)$-dimensional space which will represent y. Since y has $f(s + 1)$ points and $f(s)$ of them belong to S, y contains $f(s + 1) - f(s) = q^{s+1}$ keys. Now (*ii*), (*iii*) of Section IV need not hold. Instead, for each x, the q^M keys fall into

$$A = q^{M-s-1} \tag{27}$$

bundles of

$$K/A = q^{s+1}$$

keys each. In Fig. 5, $A = q^2$, $K/A = q$.

To find p_o^* consider the matrix, Fig. 4, corresponding to a particular pair x, x'. The q^{s+1} keys in a given column y need not be distributed one to a row [as in (*i*) of Section IV]. Each cell in the matrix contains all the keys belonging to an intersection between $(s + 1)$-dimensional spaces through x and x'. If x and x' themselves intersect in an r-dimensional space $x \cap x'$ then the cell contains the q^{r+1} keys of an $(r + 1)$-dimensional space through $x \cap x'$. B must choose one of $q^{s+1}/q^{r+1} = q^{s-r}$ equally likely rows; his probability of correctly guessing y' is

$$p_o(x, y, x') = q^{r-s}. \tag{28}$$

Now (6) and (28) provide

$$p_o^* = \sum_r h(r)q^{r-s}, \tag{29}$$

where $h(r)$ is the probability that a randomly chosen x' intersects a specific x in a space of dimension r. In (29), the range of summation is $2s + 1 - M \leqq r \leqq s - 1$ provided $2s + 1 \geqq M$. But if $2s + 1 < M$, as in Fig. 5, then $x \cap x'$ can be empty. In that case the summation (29) extends over $-1 \leqq r \leqq s - 1$.

We now show

$$h(r) = q^{(s-r)^2}g(s - r - 1, M - s - 2)g(r, s)/ \\ \{g(s, M - 1) - 1\}, \tag{30}$$

which together with (24) and (29) gives p_o^*. The factor $g(r, s)$ in (30) is the number of different r-dimensional subspaces of x; it suffices to show that the remaining terms of (30) give the probability that a randomly chosen x' intersects x in a particular subspace H of dimension r. Given x, and a subspace H, we can find M basis vectors e_0, e_1,

$\cdots, e_M$ for S such that $e_0, e_1, \cdots, e_r$ span H, and $e_0, e_1, \cdots, e_s$ span x. Each x' contains H and so has a basis containing $e_0, \cdots, e_r$. The remaining $s - r$ basis vectors of x' can have the form

$$v_j = \sum_{j=r+1}^{M} \xi_{i,j} e_j, \qquad i = r + 1, \cdots, s,$$

in which $e_0, e_1, \cdots, e_r$ do not appear. In determining $\xi_{i,j}$ one must not allow x' to intersect x in a space of dimension larger than r. This requirement is equivalent to a condition that the partial sums

$$v_i^o = \sum_{j=s+1}^{M} \xi_{i,j} e_j, \qquad i = r + 1, \cdots, s,$$

of v_i be linearly independent. Then the v_i^o span an $(s - r - 1)$-dimensional subspace x^o of the $(M - s - 2)$-dimensional subspace S^o spanned by $e_{s+1}, \cdots, e_M$. The factor $g(r - s - 1,\ M - s - 2)$ in (30) is the number of ways of choosing x^o. Having chosen H and x^o (and hence ξ_{ij} for $j = s + 1, \cdots, M$), the $(s - r)^2$ numbers

$$\xi_{ij}; \qquad i = r + 1, \cdots, s; \qquad j = r + 1, \cdots, s$$

can be chosen in $q^{(s-r)^2}$ ways to specify x' completely. Now the numerator in (30) is the number of ways of picking an x' to have an r-dimensional intersection with x and the denominator is the number $N - 1$ of messages (different from x) from which B chooses x'.

Now q, M, and s determine N, K, A, p_o^*. Table II gives some of the better designs obtained by taking $q = 2$. These all have $M = 2s + 2$, so that $K/A^2 = 1$ follows from (26) and (27). For given K, the least

Table II — Designs with q = 2

Dimensions		Keys	Inputs	Prob (B wins)
M	s	K	N	p_o^*
2	0	4	3	0.6666
4	1	16	35	0.400
6	2	64	1,395	0.2222
8	3	256	200,787	0.1176
10	4	1,024	1.09×10^8	0.0606
12	5	4,096	2.3×10^{11}	0.0308
14	6	16,384	2×10^{15}	0.0155
16	7	65,536	6×10^{19}	0.0078
18	8	262,144	8×10^{24}	0.0039
20	9	1,048,576	4×10^{30}	0.00195

Table III — Design with q = 2, M = 12, s = 5

$r = \dim(x \cap x')$	$h(r)$	$p_o(x, y, x')$
−1	0.3979	0.015625
0	0.5773	0.03125
1	0.1204	0.0625
2	0.00432	0.125
3	2.9×10^{-5}	0.25
4	3.4×10^{-8}	0.5

p_o^* was always obtained when $K/A^2 = 1$; a similar phenomenon was encountered with random designs having $\lambda = 1$ [cf. eqs. (20)]. The table contains codes having N much larger than K. At the same time, p_o^* is approximately $2/K^{\frac{1}{2}}$, which compares well with the projective plane code.

IX. CHOICE OF x′

Until now B had no control over the choice of x'. We treated x' as a random variable which B accepts as given. But suppose that B has no particular x' in mind; he merely wants to mislead G by substituting any convenient wrong message x'. An optimal strategy for B must again achieve (7) but B will select x' to maximize $p_o(x, y, x')$ for each given x, y.

A code with small p_o^*, for randomly chosen x', may now be a poor one. Table III shows more detail about the code with $q = 2$, $M = 12$, $s = 5$ in Table II. This code had $p_o^* = 0.0308$, as computed from (29). But some false messages x' intersect x in spaces of dimension $r = 4$; if B substitutes one of these, his chance of success is 0.5 [eq. (28)].

Table IV — Effect of changing field, keeping key size approximately fixed

Field	Dimensions		Key bits	K/A^2	Msg bits	Prob (B wins)	
q	M	s	$\log_2 K$		$\log_2 N$	if $r = s - 1$	averaged
256	2	0	16	1	8.01	0.0039	0.0039
41	3	1	16.08	41	10.7	0.0244	0.0250
16	4	1	16	1	16.1	0.0625	0.0078
9	5	2	15.9	9	19.2	0.1111	0.0137
7	6	2	16.86	1	25.5	0.1429	0.0058
5	7	3	16.24	5	28.3	0.2000	0.0096
4	8	3	16	1	32.5	0.2500	0.0078
3	10	4	15.9	1	40.5	0.3333	0.0082
2	16	7	16	1	65.9	0.5000	0.0078

The code is good for randomly chosen x' only because B usually has a message x' with $r = -1$ or 0.

A good code for G must now have $p_o(x, y, x')$ small uniformly, not just on the average. The code of Section VIII achieves this if q is large. For (28) shows $p_o(x, y, x') \leqq 1/q$. Unfortunately for G, increasing q has the effect of decreasing N. Then G must compromise, picking q small enough to obtain large N but large enough so that B's chance of success, $1/q$, is tolerably small. Table IV shows a typical tradeoff between N and $1/q$. The designs in Table IV all have approximately the same key size $K = 2^{16}$. Table IV shows both probabilities of success for B, $1/q$ if B makes $r = s - 1$ and the averaged value (29) if B picks x' at random. If one ignores the designs with $K/A^2 \not\doteq 1$, the averaged probability doesn't change much. To reduce $1/q$ from 0.5 to 0.1 reduces the message size, $\log N$, by a factor of 3.

REFERENCES

1. W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1, 2nd edition, New York: Wiley, 1957.
2. L. E. Dubins and L. J. Savage, *How to Gamble if You Must—Inequalities for Stochastic Processes*, New York: McGraw-Hill, 1965.
3. T. M. Cover and M. E. Hellman, "The Two-Armed-Bandit Problem with Time-Invariant Finite Memory," IEEE Trans. Info. Theory, *IT-16*, No. 2 (March 1970), pp. 185–195.
4. J. L. Kelly, Jr., "A New Interpretation of Information Rate," B.S.T.J., *35*, No. 4(July 1956), pp. 917–926.
5. S. W. Golomb, "Run-Length Encodings," IEEE Trans. Info. Theory, *IT-12*, No. 3 (July, 1966), pp. 399–401.
6. H. J. Ryser, *Combinatorial Mathematics*, Carus Math. Monograph 14, Math. Assoc. America, distributed by Wiley, N. Y., 1963.
7. M. Hall, Jr., *Combinatorial Theory*, Waltham, Mass.: Blaisdell, 1967.
8. E. J. Watson, "Primitive Polynomials (Mod 2)," Math. Comp., *41*, No. 79 (July 1962), pp. 368–370.
9. J. D. Alanen and D. E. Knuth, "Tables of Finite Fields," Sankhya, *26*, 1964, pp. 305–328.
10. W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd edition, Cambridge, Mass.: M.I.T. Press, 1972.
11. T. C. Bartee and D. I. Schneider, "Computation with Finite Fields," Information and Control, *6*, No. 1 (January 1963), pp. 79–98.
12. E. R. Berlekamp, *Algebraic Coding Theory*, New York: McGraw-Hill, 1968.
13. S. Vajda, *Patterns and Configurations in Finite Spaces*, Griffin's Statistical Monograph 22, New York: Hafner, 1967.
14. S. Vajda, *The Mathematics of Experimental Design*, Griffin's Statistical Monograph 23, New York: Hafner, 1967.
15. R. D. Carmichael, *Introduction to the Theory of Groups of Finite Order*, reprinted by Dover, N. Y., 1956.