SOME APPLICATIONS OF CODE DUALITY IN CRYPTOGRAPHY

James L. Massey

1. Introduction

Our purpose in this paper is to illustrate how the algebraic notion of a dual code has found applications in the rapidly developing field of cryptography. To make this presentation self-contained, we begin by providing in Section 2 a brief review of algebraic coding theory with emphasis on dual codes and on maximum distance separable (MDS) codes. We include a full derivation of the MacWilliam identities, which relate certain properties of a code to those of its dual, because these may be of special interest to algebraists and because the technique used in the derivation might be of use within statistical group theory. For a thorough treatment of algebraic coding theory, we refer the reader to any of the excellent textbooks [1]-[4]. For an unusual recent application of coding theory to algebra, we refer the reader to [5].

In Sections 3 and 4, we describe some applications of code duality in cryptography. Section 3 treats two applications within the area of cipher design, namely the construction of "perfect local randomizers" and of "resilient functions". These applications exploit the connection betweeen orthogonal arrays and linear codes and we include a brief treatment of this connection. Section 4 treats two applications of code duality in the area of secret sharing, namely to "threshold schemes" and to "tailored-access schemes". We wish to stress that these four examples are not exhaustive of applications of code duality in cryptography, but were chosen merely to illustrate the rich field for applications of algebraic concepts and techniques that is offered by modern cryptography.

2. Algebraic Coding Theory–A Synopsis

2.1 Linear Codes

The primary subject matter of "algebraic coding theory" is the theory of "linear codes". A q-ary *linear* (n, k) *code* is a k-dimensional subspace of the vector space $\mathrm{GF}(q)^n$. In a very real sense, algebraic coding theory is the study of the basis-dependent properties of such vector spaces. As is traditionally done in coding theory, we will take the elements of $\mathrm{GF}(q)^n$ to be "n-tuples" (row vectors).

Example 1: $V = \{[0\ 0\ 0], [1\ 0\ 1], [0\ 1\ 1], [1\ 1\ 0]\}$ is a 2-dimensional subspace of $\mathrm{GF}(2)^3$ and hence is a binary linear (3, 2) code.

Linear codes are so ubiquitous in algebraic coding theory that one usually does not bother to write "linear". One simply says, for instance, that the code in Example 1 is a "binary (3, 2) code" and we will follow this abbreviated nomenclature.

2.2 Dual Codes

The vectors $\mathbf{u}$ and $\mathbf{v}$ in $\mathrm{GF}(q)^n$ are said to be *orthogonal* if their scalar product vanishes, i.e., if $\mathbf{u}\mathbf{v}^T = 0$. If V is a q-ary linear (n, k) code, then the *dual code* $V^\perp$ is the set of all n-tuples $\mathbf{u}$ that are orthogonal to every n-tuple $\mathbf{v}$ in V.

Example 1 (continued): $V = \{[0\ 0\ 0], [1\ 0\ 1], [0\ 1\ 1], [1\ 1\ 0]\}$ implies that $V^\perp = \{[0\ 0\ 0], [1\ 1\ 1]\}$. Note that $V^\perp$ is a binary (3, 1) code.

In $\mathrm{GF}(q)^n$, a non-zero vector can be orthogonal to itself. For instance, in $\mathrm{GF}(2)^2$, $\mathbf{u} = [1\ 1] \Rightarrow \mathbf{u}\mathbf{u}^T = 1 + 1 = 0$. This fact caused the mathematician J. H. van Lint, in his thoughtful textbook on coding theory [1], to warn the reader to "be careful not to think of $V^\perp$ as an orthogonal complement in the sense of vector spaces . In the case of a finite field, the subspaces V and $V^\perp$ can have an intersection larger than $\{\mathbf{0}\}$ and in fact they can even be equal".

Example 2: In $\mathrm{GF}(2)^2$, $V = \{[0\ 0], [1\ 1]\}$ implies that $V^\perp = V = \{[0\ 0], [1\ 1]\}$ so that V is a *self-dual* code.

Although the subspaces V and $V^\perp$ can have an intersection larger than $\{\mathbf{0}\}$, most of the usual relations of linear algebra still hold in $\mathrm{GF}(q)^n$. In particular, if V is a q-ary linear (n,k) code, then $(V^\perp)^\perp = V$ and $\dim(V)+\dim(V^\perp) = n$. Thus, for a self-dual code, one must have $n = 2k$.

2.3 Generator Matrices

A *generator matrix* for the q-ary (n,k) code V is any matrix G whose rows are a basis for V.

Example 3: The six matrices

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

are all of the different generator matrices for the binary (3, 2) code $V = \{[0\ 0\ 0], [1\ 0\ 1], [0\ 1\ 1], [1\ 1\ 0]\}$ of Example 1.

Engineers like to think of a generator matrix as an "encoding matrix" for the code. This interpretation results from writing $\mathbf{v} = \mathbf{u}G$ to indicate the manner in which the "information vector" $\mathbf{u}$, which is a q-ary k-tuple, is encoded into the codeword $\mathbf{v}$, which is a q-ary n-tuple.

Example 4: If the first of the six generator matrices in Example 3 is used for encoding, the encoding equation with $\mathbf{v} = [v_1\ v_2\ v_3]$ and $\mathbf{u} = [u_1\ u_2]$ becomes

$$[v_1\ v_2\ v_3] = [u_1\ u_2] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

or, equivalently, $v_1 = u_1$, $v_2 = u_2$ and $v_3 = u_1 + u_2$.

The *rate* R of a q-ary linear (n,k) code is the ratio $R = k/n$, which measures the number of "information symbols" per "code symbol". Note that a self-dual code must have rate $R = 1/2$.

2.4 Parity-Check Matrices

A *parity-check matrix* for the q-ary (n,k) code V is any matrix H whose rows span $V^\perp$. The parity-check matrix H is *reduced* if its rows are a basis for $V^\perp$, i.e., if H is a generator matrix for the code $V^{\perp}$.

Example 5: The matrix $H = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ is the unique reduced parity-check matrix for the binary (3, 2) code $V = \{[0\ 0\ 0], [1\ 0\ 1], [0\ 1\ 1], [1\ 1\ 0]\}$ of Example 1 for which $V^{\perp} = \{[0\ 0\ 0], [1\ 1\ 1]\}$.

Proposition 1. (Test for Code Membership) *If H is a parity-check matrix for the q-ary (n, k) code V, then the q-ary n-tuple $\mathbf{v}$ is a codeword if and only if*

$$\mathbf{v}H^T = \mathbf{0}. \tag{1}$$

This proposition follows from the fact that (1) is satisfied if and only if $\mathbf{v}$ is orthogonal to every row of H and hence to every linear combination of these rows, i.e., to every vector in $V^{\perp}$.

Example 6: The 3-tuple $[v_1\ v_2\ v_3]$ is a codeword in the binary (3, 2) code with parity check matrix $H = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ if and only if $[v_1\ v_2\ v_3] \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T = 0$, i.e., if and only if $v_1 + v_2 + v_3 = 0$.

Each such linear constraint is called a "parity check" of the linear code. Each of the q^{n-k} vectors in the dual code determines such a parity check, including the codeword $\mathbf{0}$ that determines the trivial parity check $0 = 0$.

2.5 Hamming Weight and Distance

The *Hamming weight* $w_H(.)$ of an n-tuple is the number of its non-zero components. The *Hamming distance* $d_H(.,.)$ between two q-ary n-tuples is the number of components in which they differ, i.e., the Hamming weight of their difference.

Example 7: $d_H([1\ 1\ 0], [0\ 1\ 1]) = w_H([1\ 1\ 0] - [0\ 1\ 1]) = w_H([1\ 0\ 1]) = 2.$

The *minimum distance*, $d_{\min}$, of a code is the smallest Hamming distance between pairs of distinct codewords in the code.

If a q-ary codeword $\mathbf{v}$ is transmitted through some channel and the q-ary n-tuple $\mathbf{r}$ is received, one says that a pattern of $d_H(\mathbf{r}, \mathbf{v})$ errors has occurred, namely the error pattern $\mathbf{e} = \mathbf{r} - \mathbf{v}$.

Proposition 2. (Properties of Minimum Distance)
i) The minimum distance, $d_{\min}$, of a q-ary (n, k) code equals the minimum weight, $w_{\min}$, of its non-zero codewords.
ii) A code can detect all patterns of t or fewer errors in every codeword if and only if $d_{\min} > t$.
iii) A code can correct all patterns of t or fewer errors in every codeword if and only if $d_{\min} > 2t$.

The first assertion is a consequence of the fact that the set of differences $V - V$ is just V itself. The second assertion is trivial, and the third follows from the fact that Hamming distance is a metric on the space $\mathrm{GF}(q)^n$ so that decoding to the nearest codeword corrects all errors of Hamming weight t or less when and only when $d_{\min} > 2t$.

The following proposition often provides the simplest and most insightful way to determine the minimum distance of a q-ary (n, k) code.

Proposition 3. (Determination of Minimum Distance) *The minimum distance of a q-ary (n, k) code with parity-check matrix H equals the smallest number of columns of H that form a linearly dependent set.*

This proposition follows from the fact that, according to (1), a codeword $\mathbf{v}$ of Hamming weight $w > 0$ specifies a set of w columns of H that are linearly dependent.

Proposition 4. (Singleton's Bound) *For every q-ary (n, k) code, $d_{\min} \leq n - k + 1$.*

This bound follows from the fact that a parity-check matrix has rank $n - k$ and thus every set of $n - k + 1$ of its columns must be linearly dependent.

2.6 Maximum Distance Separable Codes

A q-ary (n,k) code is said to be *maximum distance separable* (MDS) if $d_{\min} = n-k+1$. A subset of $\{1,2,\ldots,n\}$ containing k coordinates is an *information set* for a q-ary (n,k) code if no pair of distinct codewords coincide in these k coordinates. This terminology arises from the fact that the codeword symbols in the coordinates of an information set may be taken as the "information symbols" since they completely determine the entire codeword.

Proposition 5. (Properties of MDS Codes)
i) The q-ary (n,k) code V with parity-check matrix H is MDS if and only if the columns of H in every choice of $n-k$ columnar positions are linearly independent.
ii) Every subset of $\{1,2,\ldots,n\}$ containing k coordinates is an information set for a q-ary (n,k) code if and only if it is an MDS code.
iii) If V is an MDS code, then so is its dual code $V^{\perp}$.

The first assertion of this proposition is an immediate consequence of Proposition 3. The second assertion follows from the fact that if and only if $d_{\min} \le n-k$, then there would be two different codewords that agreed in k (or more) coordinates and hence (any choice of k of) these coordinates would not form an information set. If G is a generator matrix for a q-ary (n,k) code, then a set of k coordinates is an information set if and only if the corresponding columns of G are linearly independent, which, because G is a parity-check matrix for $V^{\perp}$, establishes the third assertion.

The celebrated Reed-Solomon codes are MDS codes. There is a q-ary (n,k) Reed-Solomon code for every $1 \le k < n < q$. The binary (3, 2) code V of Example 1 and its (3, 1) dual code $V^{\perp}$ are both MDS, but neither is a Reed-Solomon code.

2.7 MacWilliams Identities

One of the contributions of coding theory to algebra was the demonstration by MacWilliams that the Hamming weights of the codewords in V uniquely determine the Hamming weights of the codewords in the dual code $V^{\perp}$. We now give a derivation of these "MacWilliams identities", which is a slight modification of that given by Chang and Wolf [6]. For simplicity, we consider the binary case only, but the derivation generalizes easily.

Let $\mathbf{X} = [X_1\ X_2\ \ldots\ X_n]$ be a random vector in $\mathrm{GF}(2)^n$ such that X_1, X_2, ... X_n are independent random variables with $\mathrm{P}[X_i = 1] = \epsilon$, all i. If $\mathbf{v}$ is a binary n-tuple of Hamming weight i then, for every $0 \leq \epsilon \leq 1$,

$$\begin{aligned} \mathrm{P}[\mathbf{X} = \mathbf{v}] &= \epsilon^i (1-\epsilon)^{n-i}\ , \text{ and} && (2) \\ \mathrm{P}[\mathbf{X}\mathbf{v}^T = 1] &= \frac{1}{2}[1 - (1-2\epsilon)^i], && (3) \end{aligned}$$

where the right side of (3) is just the probability of an odd number of 1's in the i components of $\mathbf{X}$ selected by the non-zero components of $\mathbf{v}$. Let V be a binary (n, k) code with *weight enumerator* $(A_0,\ A_1,\ \ldots\ A_n)$, i.e., A_i is the number of codewords in V with Hamming weight i. We calculate the probability that $\mathbf{X}$ is a codeword in V in two different ways.

Let $\mathbf{v}_1$, $\mathbf{v}_2$, ... , $\mathbf{v}_{2^k}$ be the 2^k codewords in V. Then

$$\begin{aligned} \mathrm{P}[\mathbf{X} \in V] &= \mathrm{P}[\mathbf{X} = \mathbf{v}_1\ \cup\ \mathbf{X} = \mathbf{v}_2\ \cup\ \ldots\ \cup\ \mathbf{X} = \mathbf{v}_{2^k}] \\ &= \mathrm{P}[\mathbf{X} = \mathbf{v}_1] + \mathrm{P}[\mathbf{X} = \mathbf{v}_2] + \ldots + \mathrm{P}[\mathbf{X} = \mathbf{v}_{2^k}] \end{aligned}$$

because the events in the union are mutually exclusive. Thus (2) gives

$$\mathrm{P}[\mathbf{X} \in V] = \sum_{i=0}^{n} A_i \epsilon^i (1-\epsilon)^{n-i}. \tag{4}$$

Now let $V^{\perp}$ with weight enumerator $(B_0,\ B_1,\ \ldots\ B_n)$ be the binary $(n, n-k)$ dual code of V and let $\mathbf{v}_1$, $\mathbf{v}_2$, ... , $\mathbf{v}_{2^{n-k}}$ be the 2^{n-k} codewords in $V^{\perp}$. Then

$$\mathrm{P}[\mathbf{X} \in V] = 1 - P[\mathbf{X}\mathbf{v}_1^T = 1\ \cup\ \mathbf{X}\mathbf{v}_2^T\ \cup\ \ldots\ \cup\ \mathbf{X}\mathbf{v}_{2^{n-k}}^T = 1] \tag{5}$$

because $\mathbf{X}$ will be a codeword if and only if it satisfies all 2^{n-k} parity checks. To proceed further, we need the following:

Lemma 1. (Generalization of Mutually Exclusive Events [6]) *If E_1, E_2, ... E_M are events such that when any of these events occur exactly L of these events occur, then* $\mathrm{P}[E_1 \cup E_2 \cup \ldots \cup E_M] = \frac{1}{L}(\mathrm{P}[E_1] + \mathrm{P}[E_2] + \;\; + \mathrm{P}[E_M])$.

This somewhat surprising result is a simple consequence of the fact that each sample point in the union event lies in exactly L of the individual events. But if any of the events $\mathbf{X}\mathbf{v}_1^T = 1$, $\mathbf{X}\mathbf{v}_2^T = 1$, ... , $\mathbf{X}\mathbf{v}_{2^{n-k}}^T = 1$ occur, then exactly half of these 2^{n-k} events occur because the vectors in $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{2^{n-k}}\}$ giving value 1 are the only coset of those that give the value 0. Applying the lemma in (5) gives

$$\mathrm{P}[\mathbf{X} \in V] = 1 - \frac{1}{2^{n-k-1}}(\mathrm{P}[\mathbf{X}\mathbf{v}_1^T = 1] + \mathrm{P}[\mathbf{X}\mathbf{v}_2^T = 1] + \ldots + \mathrm{P}[\mathbf{X}\mathbf{v}_{2^{n-k}}^T = 1]).$$

Using (3) gives

$$\mathrm{P}[\mathbf{X} \in V] = 1 - \frac{1}{2^{n-k-1}} \sum_{i=0}^{n} B_i \frac{1}{2}[1 - (1-2\epsilon)^i].$$

But $B_0 + B_1 + \ldots + B_n = 2^{n-k}$ so we have finally

$$\mathrm{P}[\mathbf{X} \in V] = \frac{1}{2^{n-k-1}} \sum_{i=0}^{n} B_i (1-2\epsilon)^i. \tag{6}$$

Equating expressions (4) and (6) for $\mathrm{P}[\mathbf{X} \in V]$ now gives:

Proposition 6. (MacWilliams' Identities) *For every $0 \le \epsilon \le 1$, the weight enumerator $(A_0, A_1, \ldots A_n)$ of a binary (n,k) code V and the weight enumerator $(B_0, B_1, \ldots B_n)$ of its dual $V^\perp$ satisfy*

$$\frac{1}{2^{n-k-1}} \sum_{i=0}^{n} B_i (1-2\epsilon)^i = \sum_{i=0}^{n} A_i \epsilon^i (1-\epsilon)^{n-i},$$

which is equivalent to the identities

$$\frac{2^j}{2^{n-k}} \sum_{i=j}^{n} \binom{i}{j} B_i = \sum_{i=0}^{j} (-1)^i \binom{n-i}{n-j} A_i \tag{7}$$

for $n \ge j \ge 0$.

Example 8: Consider the $(n, k) = (3, 2)$ binary code of Example 1 with weight enumerator $(A_0,\ A_1,\ A_2,\ A_3) = (1,\ 0,\ 3,\ 0)$. From (7), we find

$$\begin{array}{lrcl}
(j=3) & 4B_3 = A_0 - A_1 + A_2 - A_3 = 4 & \Rightarrow & B_3 = 1. \\
(j=2) & 2(B_2 + 3B_3) = 3A_0 - 2A_1 + A_2 = 6 & \Rightarrow & B_2 = 0. \\
(j=1) & B_1 + 2B_2 + 3B_3 = 3A_0 - A_1 = 3 & \Rightarrow & B_1 = 0. \\
(j=0) & \frac{1}{2}(B_0 + B_1 + B_2 + B_3) = A_0 = 1 & \Rightarrow & B_0 = 1.
\end{array}$$

3. Applications of Code Duality to Cipher Design

3.1 Orthogonal Arrays and Dual Codes

Because both applications of code duality to cipher design that we will treat rely on the relationship between orthogonal arrays and dual codes, we first present a brief treatment of this relationship.

An *orthogonal array* $\mathrm{OA}_\lambda(t, n, q)$ of *power* t is a rectangular array of q-ary symbols with n columns such that in every choice of t ($t \geq 1$) columns, each of the q^t possible q-ary t-tuples occurs in exactly λ rows. Orthogonal arrays with distinct rows are called *simple.*

Example 9: The binary array

$$\begin{array}{ccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0
\end{array}$$

is a simple $\mathrm{OA}_2(t, n = 7, q = 2)$ of power $t = 2$. In every pair of columns, each of the four binary 2-tuples occurs in exactly 2 rows. Note that this array is *not* an $OA_1(t, n = 7, q = 2)$ of power $t = 3$, as one sees from columns 1, 3 and 4.

The following lemma will prove useful.

Lemma 2. *If the matrix G is the generator matrix of a q-ary linear (n,k) code V, then the minimum distance $d^{\perp}$ of the dual code $V^{\perp}$ is the smallest number t of columns of G that form a $k \times t$ matrix with rank less than t.*

This lemma follows immediately from Proposition 3 and the fact that G is a parity-check matrix for the dual code.

Because c columns of a matrix are linearly dependent if and only if they form a submatrix of rank less than c, Lemma 2 implies the following result that will be the key to our applications.

Proposition 7. (Orthogonal Array Power and Dual Code Distance) *The maximum t for which the rectangular array, whose rows are the codewords of a q-ary linear (n,k) code V with $k < n$ and $d^{\perp} > 1$, is an orthogonal array $OA_{\lambda}(t,n,q)$ [necessarily simple] of power t is $t = d^{\perp} - 1$.*

Example 10: The binary array of Example 9 has as its rows the codewords of the binary (7, 3) code with generator matrix

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

The dual code is a binary (7, 4) code with $d_{\min} = 3$, which is the famous Hamming single-error-correcting code of this length.

3.2 Perfect Local Randomizers

We now consider functions that can be used within ciphers to "stretch randomness", i.e., to transform k truly random bits into a larger number n of bits in the manner that all subsets of the transformed bits that are not too large remain completely random. An injective (or "one-to-one") function f from k q-ary digits to n q-ary digits ($n > k$) is a *perfect local randomizer* (PLR) of order t if choosing the k input digits uniformly at random guarantees that the output digits in every choice of t components (not necessarily consecutive) of the output n-tuple are also uniformly random [7]. Fig. 1 emphasizes that a

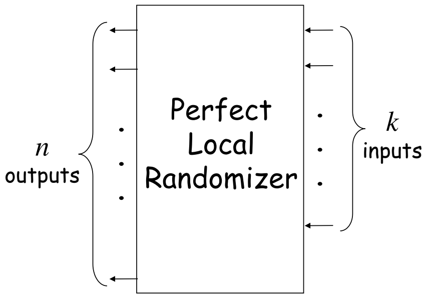


Fig. 1: Illustration of a Perfect Local Randomizer (PLR)

PLR must be a function with more output variables than input variables. One cryptographic application of such functions is in expanding a relatively short randomly-selected secret key into the many "subkeys" that are typically required within a cipher. If a PLR of order at least the length of the subkeys is used for the expansion, then each subkey will still be completely random.

We next need a characterization of PLR's in terms of orthogonal arrays. This is easily obtained upon observing that the digits in t chosen columns of a q-ary rectangular array will be uniformly random when a row of the array is chosen uniformly at random if and only if every possible t-tuple of values in these columns occurs in precisely the same number of rows.

Proposition 8. (PLR's and Orthogonal Arrays)
An injective function f from k q-ary digits to n q-ary digits ($n > k$) is a perfect local randomizer of order t if and only if the rectangular array having as its rows the range of f is an orthogonal array $OA_\lambda(t, n, q)$.

Example 11: An encoder for the binary (7, 3) code of Example 10 is a perfect local randomizer of order 2. It "stretches" a 3-bit random input to a 7-bit output, every pair of whose bits are completely random.

Propositions 7 and 8 give immediately a fundamental connection between perfect local randomizers and linear codes.

Proposition 9. (PLR's and Dual Codes)
An injective function from k q-ary digits to n q-ary digits is a perfect local randomizer of maximum order t ($t \geq 1$) if it is the encoder for a q-ary (n, k) code with dual distance $d^{\perp} = t + 1$.

Note that it is not necessary that the encoder in Proposition 9 be a linear encoder, i.e., a realization of some generator matrix for the code, as follows from the fact that orthogonal array properties do not depend on the order of the rows within the array. It is apparent from Proposition 9 that linear codes provide a rich source of perfect local randomizers and that it is the dual distance of these codes rather than their minimum distance that is of importance.

3.3 Resilient Functions

We now consider functions that can be used within ciphers to defeat "divide-and-conquer" attacks in which the attacker tries to gain information about the cipher by studying its behavior when a small number of input digits are fixed. A function f from n q-ary digits to $n - k$ q-ary digits $(1 < k < n)$ is said to be *t-resilient* if, for every choice of t of the input digits, when the values of these t digits are fixed and the values of the other $n - t$ input digits are chosen uniformly at random, all $n - k$ output digits are uniformly random [8]. Fig. 2 emphasizes that a resilient function must be a function with more input variables than output variables.

Example 12: The function $y_1 = f(x_1, x_2, x_3)$ from GF$(2)^3$ to GF(2) with function table

x_1	x_2	x_3	y_1
0	0	0	1
1	0	1	1
0	1	1	1
1	1	0	1
1	0	0	0
0	0	1	0
1	1	1	0
0	1	0	0

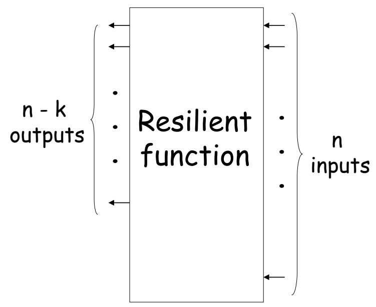


Fig. 2: Illustration of a Resilient Function

is 2-resilient. When any two inputs are fixed, say $x_1 = 0$ and $x_3 = 1$ (which we see corresponds to rows 3 and 6 of the function table), the remaining variable (in this case x_2) takes on the values 0 and 1 equally often (in this case once). Thus, when the rows are equiprobable, the output y_1 will takes on the values 0 and 1 each with probability $\frac{1}{2}$.

The argument given within this example directly implies the following characterization of resilient functions.

Lemma 3. *The function $f(x_1, x_2, \ldots x_n) = [y_1, y_2, \ldots y_{n-k}]$ from n q-ary digits to $n-k$ q-ary digits is t-resilient if and only if 1) f is balanced, i.e., the sets $f^{-1}(y_1, y_2, \ldots y_{n-k})$ have the same cardinality [which must be q^k] for all q^{n-k} choices of $y_1, y_2, \ldots y_{n-k}$, and 2) for each of the q^{n-k} choices of $y_1, y_2, \ldots y_{n-k}$, the array whose rows are the q^k n-tuples $[x_1, x_2, \ldots x_n]$ in $f^{-1}(y_1, y_2, \ldots y_{n-k})$ is an orthogonal array of power t.*

A collection of orthogonal arrays, all of the same size and all having power t, that partition the set of all q-ary n-tuples is what Stinson has called a *large set of orthogonal arrays of power t* [9]. Lemma 3 states simply that $f(x_1, x_2, \ldots x_n) = [y_1, y_2, \ldots y_{n-k}]$ is t-resilient if and only if the sets $f^{-1}(y_1, y_2, \ldots y_{n-k})$ for all q^{n-k} choices of $y_1, y_2, \ldots y_{n-k}$ form a large set of orthogonal arrays of power t.

Example 13: Note for the function in Example 12 that the arguments that give the value $y_1 = 1$, namely

$$\begin{matrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{matrix}$$

are just the codewords in the binary (3, 2) code V with $d^{\perp} = 3$ of Example 1, Thus Proposition 7 ensures that this is an orthogonal array of power $t = 2$. Note also that the arguments that give the value $y_1 = 0$, namely

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{matrix}$$

are the 3-tuples in the only proper coset of the code V. A simple check shows that this is also an orthogonal array of power $t = 2$. Thus these two arrays do indeed form a a large set of orthogonal arrays of power $t = 2$.

Adding a fixed q-ary n-tuple to every row of an orthogonal array with entries in GF(q) quite obviously gives another orthogonal array with the same power. Hence a coset of an (n, k) code V with dual distance $d^{\perp}$ is also an orthogonal array with maximum power $d^{\perp} - 1$ and of course has the same cardinality as V. It follows that V and all its proper cosets form a large set of orthogonal arrays of power $t = d^{\perp} - 1$, which establishes the following proposition.

Proposition 10. (Resilient Functions and Dual Codes)
If $f(x_1, x_2, \ldots x_n) = [y_1, y_2, \ldots y_{n-k}]$ is a function from n q-ary digits to $n - k$ q-ary digits $(1 \leq k < n)$ such that, for each of the q^{n-k} choices of $y_1, y_2, \ldots y_{n-k}$, the set $f^{-1}(y_1, y_2, \ldots y_{n-k})$ is a different coset of a linear (n, k) q-ary code with dual distance $d^{\perp} \geq 2$, then the maximum t for which $f(x_1, x_2, \ldots x_n)$ is t-resilient is $t = d^{\perp} - 1$.

We note that there is no requirement in Proposition 10 that the function f be linear. The q^{n-k} values of the function can be assigned to the different cosets of V quite arbitrarily. We see also from this proposition that (n, k) codes provide a rich source of resilient functions.

3.4 Nonlinear Codes and Orthogonal Arrays

In this subsection and only here, we briefly consider nonlinear codes. By a q-ary *nonlinear* (n, k) *code*, we mean a subset of cardinality q^k of the n-tuples in $\mathrm{GF}(q)^n$. Delsarte [10],[11] has shown that one can define a "dual distance" for such a code by applying the MacWilliams' identities to the nonlinear code as if it had a true dual, cf. also [12], and that this "dual distance" determines the power of the orthogonal array formed by the codewords in precisely the manner stated in Proposition 7. Stinson and the author [12] showed that one can also create a large family of orthogonal arrays from a q-ary nonlinear (n, k) code when the code is *systematic*, i.e., when there is a set of k coordinates such that no pair of distinct codewords coincide in these coordinates. The set of q^{n-k} subsets of $\mathrm{GF}(q)^n$ obtained by adding to the q^k codewords a fixed n-tuple that is all-zero in these k coordinates forms a large family of orthogonal arrays that is analogous to the set of all cosets of a linear (n, k) code. It was shown in [12] that resilient functions strictly superior to those obtained from linear codes can be obtained from nonlinear codes in this manner.

4. Applications of Code Duality to Secret Sharing

4.1 Perfect Secrecy

Before treating "secret sharing" itself, we must say a few words about Shannon's notion of "perfect secrecy". In what may well be called the first scientific paper on cryptography [13], Shannon offered the model of a "general secrecy system" shown in Fig. 3. As this figure makes clear, Shannon assumed that the attacker or "enemy cryptanalyst" has access only to the cryptogram E but not to the key K shared by the sender and receiver who attempt to convey the message M secretly. Shannon said that such a system provides *perfect secrecy*, the mathematical analog of an "unbreakable" cipher, if the cryptogram E and the message M are statistically independent.

Shannon proved that the cipher introduced by Vernam in 1926 and diagrammed in Fig. 4 provides perfect secrecy. [Vernam indeed asserted in

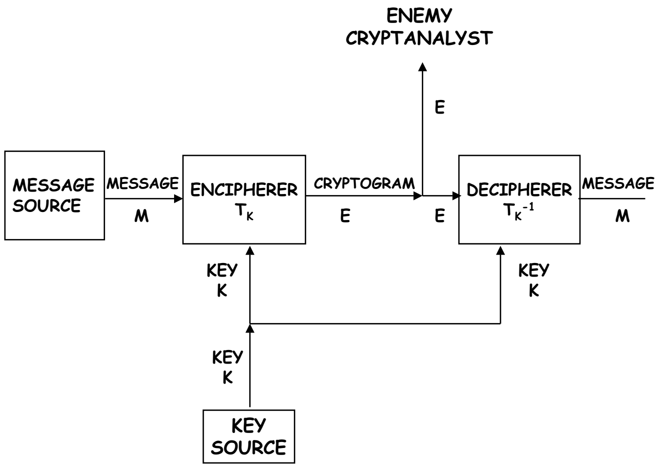


Fig. 3: Shannon's Model of a "General Secrecy System

his paper [14] that his cipher was "unbreakable", but gave no mathematical justification–many people over many centuries had mistakenly made a similar claim.] In Vernam's cipher, the key K is a binary coin-tossing sequence of the same length as the binary message M. [The *binary symmetric source* (BSS) in Fig. 4 is simply a device that produces such a *completely random* binary sequence.] The key K is added bit-by-bit modulo-two to the message M to produce the cryptogram E. Regardless of the value m of M, the cryptogram E is equally likely to take on every possible value, i.e., the cryptogram E is statistically independent of M and hence perfect secrecy is obtained. No amount of effort by the attacker can produce any information about M from E. It is of course essential that the key be used only once, for which reason Vernam's cipher is often referred to today as the "one-time pad".

4.2 Threshold Schemes for Secret Sharing

"Secret sharing" refers to any scheme in which some secret (which we will take to be a q-ary digit) is distributed among two or more parties in a manner such that only certain specified coalitions of the parties can recover the entire secret. The "classical" scheme for secret sharing is illustrated in Fig. 5, which shows

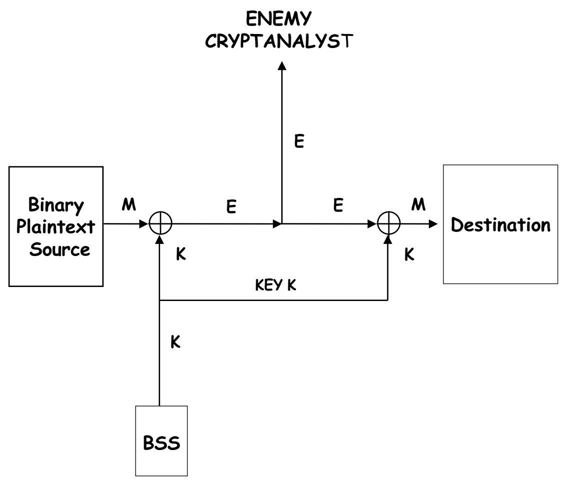


Fig. 4: Vernam's Cipher ("Perfect Secrecy")

how two criminals might share the combination of a safe in which they have placed their ill-gotten gains. There are $2^{10} = 1024$ different possibilities for the 10-bit combination, which we consider to be an element of GF(2^{10}). Because the share of each criminal is 5 bits of the combination, neither can easily open the safe alone. However, the secret has "leaked" into the shares because, given one 5-bit share, there are only $2^5 = 32$ possibilities for the remaining share needed to determine the secret. A cheating criminal need only try out these 32 combinations to find the one that opens the safe.

The "no-leakage" secret-sharing scheme illustrated in Fig. 6 ensures that neither criminal alone has any advantage in opening the safe over someone who owns no share of the secret combination. The share of the first criminal is a completely random 10-bit string that is obviously of no use by itself in opening the safe. But the share of the second criminal is the cryptogram E in Vernam's cipher when the message M is the secret and the first share is the key K—hence the second share alone likewise gives no information about the secret.

A secret-sharing scheme is said to be *perfect* if no coalition of shareholders can obtain any information about the secret except for those coalitions that are specified as authorized to obtain the secret. An (N, T) *threshold secret sharing*

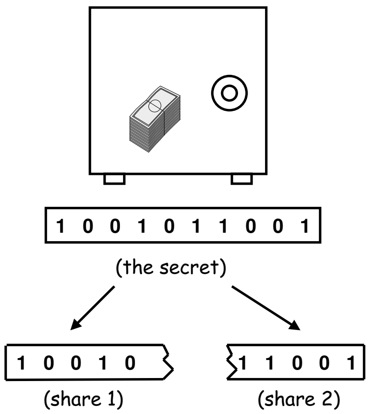


Fig. 5: Classical Secret Sharing

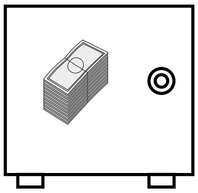
scheme is one in which the authorized coalitions are all coalitions of T or more of the N shareholders.

The secret-sharing scheme in Fig. 6 is a perfect $(N = 2, T = 1)$ threshold secret-sharing scheme. We will now see that this scheme is in fact derived from the (3, 2) linear code over $\mathrm{GF}(2^{10})$ with generator matrix and parity-check matrix

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \text{ and } H = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix},$$

respectively. Here "0" and "1" denote the neutral elements of the additive and multiplicative groups, respectively, of $\mathrm{GF}(2^{10})$ and would be represented as the 10×10 all-zero matrix and the 10×10 identity matrix, respectively, at the bit level. This code specifies the above $(N = 2, T = 1)$ threshold secret-sharing scheme in the following manner: Letting $[v_1\ v_2\ v_3]$ be the codeword in $\mathrm{GF}(2^{10})^3$, the first digit v_1 is taken to be the 10-bit secret; the second digit v_2 is chosen uniformly at random in $\mathrm{GF}(2^{10})$, and the third digit v_3 is computed so that $[v_1\ v_2\ v_3]H^{\perp} = 0$ or, equivalently, such that $v_3 = v_1 + v_2$. Finally, v_2 and v_3 are selected to be the two shares of the secret.

While the coding scheme just described is obviously identical to the perfect $(N = 2, T = 1)$ threshold secret-sharing scheme described previously, it is worthwile to seek more insight into why the coding construction succeeds. We



The Secret:	1 0 0 1 0 1 1 0 0 1
Share 1: BSS output	0 0 1 1 0 1 0 1 1 1
Share 2: Secret $\oplus$ BSS output	1 0 1 0 0 0 1 1 1 0

Fig. 6: "No-Leakage" Secret Sharing

note first that the (3, 2) code used has $d_{\min} = n - k + 1 = 2$ and hence is an MDS code. Proposition 5 assures us then that every pair of digit positions forms an information set. Because the first position together with any other position forms an information set, it follows that for each choice of the first digit [i.e., the secret], every possible value of the digit in the other position occurs in exactly one codeword. Thus specifying this other digit gives no information about the secret. This argument generalizes to every MDS code and establishes the following result.

Proposition 11. (Perfect Threshold Schemes and MDS Codes) *If the q-ary secret is taken to be the first digit of the q-ary n-tuple* $[v_1\ v_2\ \ldots\ v_n]$*, if the next* $k-1$ *digits are chosen uniformly at random, and if the final* $n-k$ *digits are computed so that* $[v_1\ v_2\ \ldots\ v_n]$ *is a codeword in an* (n, k) *q-ary MDS linear code, then* $v_2,\ v_3,\ \ldots,\ v_n$ *are the shares of a perfect* $(N = n-1, T = k)$ *threshold secret-sharing scheme.*

Secret sharing was introduced by Shamir [15] who gave an interpolation construction for perfect threshold schemes that is equivalent to the use of Reed-Solomon codes in the scheme of Proposition 11. The connection to Reed-Solomon codes was pointed out by McEliece and Sarwate [16].

4.3 Tailored-Access Schemes

We now consider, following [17], perfect secret-sharing schemes with an access structure not limited to a threshold on the number of users in an authorized coalition. We say that a set of shares in a perfect secret-sharing system is a *minimal share set* if this set of shares, but no proper subset thereof, determines the secret. It follows that the authorized coalitions of shareholders are those coalitions whose share set includes a minimal share set. In a perfect (N, T) threshold secret-sharing scheme, the minimal share sets are just the sets containing exactly T shares. We give now an example of a less regular access structure.

Example 14: Consider the 2^m-ary (5, 3) code with parity-check matrix

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix},$$

which implies that the necessary and sufficient condition for $[v_1\ v_2\ v_3\ v_4\ v_5]$ to be a codeword is that $v_1 + v_2 + v_3 = 0$ and $v_2 + v_4 + v_5 = 0$, or equivalently that $v_1 + v_2 + v_3 = 0$ and $v_1 + v_3 + v_4 + v_5 = 0$. It is easily checked that if v_1 is the secret, if v_2, v_3, v_4 and v_5 are the shares, and if v_2 and v_4 are chosen uniformly at random [where we note that $\{1,\ 2,\ 4\}$ is an information set], then $\{v_2, v_3\}$ and $\{v_3, v_4, v_5\}$ are the only minimal share sets.

We now interpret minimal share sets in terms of code properties. A codeword $[v_1\ v_2\ \ldots\ v_n]$ in a q-ary (n, k) code is *minimal* if it is non-zero, if its leftmost non-zero component is a 1, and if the coordinates where its components are non-zero do not include all the coordinates containing the non-zero components of any other codeword whose leftmost non-zero component is a 1. In an (n, k) MDS code, the minimal codewords are just the codewords with Hamming weight $d_{\min} = n - k + 1$ whose leftmost non-zero component is a 1. In Example 14,

$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 & 0 & 1 & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 & 1 & 1 & 1 \end{bmatrix}$ are all the minimal codewords in the dual code. The minimal codewords in the dual code with first component equal to 1 correspond to the minimal share sets in the manner that their last $n-1$ components [i.e., those components corresponding to shares of the secret] are the incidence vectors for the minimal share sets. In Example 14, the minimal codewords $\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 & 1 & 1 & 1 \end{bmatrix}$ in the dual code correspond to the minimal share sets $\{v_2, v_3\}$ and $\{v_3, v_4, v_5\}$, respectively.

Proposition 12. (Access Structures and Minimal Codewords)
If the q-ary secret is taken to be the first digit of the q-ary n-tuple $[v_1\ v_2\ \ldots\ v_n]$, *if the next* $k-1$ *digits are chosen uniformly at random, and if the final* $n-k$ *digits are computed so that* $[v_1\ v_2\ \ldots\ v_n]$ *is a codeword in a q-ary* (n,k) *linear code with dual distance* $d^{\perp}$ *whose first* k *components form an information set, then the minimal share sets in the resulting secret-sharing scheme are those share sets whose shares correspond to the remaining non-zero positions of a minimal codeword in the dual code whose first component is 1.*

This proposition follows from the fact that in a linear code the only constraints among code digits are those given by the codewords of the dual code. Thus, the secret v_1 is determined by some share set if and only if that share set includes all the shares corresponding to the non-zero components beyond the first of a codeword in the dual code whose first digit is 1. It follows that the minimal share sets contain precisely the shares corresponding to the non-zero components beyond the first of a minimal codeword in the dual code whose first digit is 1.

We remark in closing that Proposition 12 suggests that it is more fundamentally the fact that the dual code is also an MDS code, rather than the fact that the code itself is MDS, that is the foundation for the perfect $(N = n-1, T = k)$ threshold secret-sharing scheme described in Proposition 11.

References

[**1**] J. H. van Lint, *Introduction to Coding Theory* (1982). New York: Springer.

[**2**] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes* (1977). Amsterdam: North-Holland.

[**3**] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd Ed. (1972). Cambridge, Mass.: M.I.T. Press.

[**4**] R. E. Blahut, *Theory and Practice of Error Control Codes* (1983). Reading, Mass.: Addison-Wesley.

[**5**] S. Shokranian and M. A. Shokrollahi, *Coding Theory and Bilinear Complexity* (1993). Forschungszentrum Jülich.

[**6**] S.-C. Chang and J. K. Wolf, "A simple derivation of the MacWilliams' identity for linear codes," *IEEE Trans. Inform. Theory* (1980), vol. IT-26, 476 - 477.

[**7**] U. M. Maurer and J. L. Massey, "Local randomness in pseudorandom sequences," *J. Cryptology* (1991), vol. 4, 135-149.

[**8**] B. Chor, O. Goldreich, J. Hastad, J. Friedman, S. Rudich and R. Smolensky, "The bit extraction problem or t-resilient functions," *Proc. 26th IEEE Symp. Foundations Computer Sci.* (1985), 396-407.

[**9**] D. R. Stinson, "Resilient functions and large sets of orthogonal arrays," *Congressus Numericus* (1993), vol. 92, 105-110.

[**10**] P. Delsarte, "Bounds for unrestricted codes, by linear programming," *Philips Res. Repts. Suppl.* (1972), vol. 27, 272-289.

[**11**] P. Delsarte, "Application and generalization of the MacWilliams transform in coding theory, *Proc. 15th Benelux Symp. Inform. Th.* (1994), 9-44.

[12] D. R. Stinson and J. L. Massey, "An infinite class of counterexamples to a conjecture concerning non-linear resilient functions," *J. Cryptology* (1995), vol. 8, no. 3, 167-173.

[13] C.E. Shannon, "Communication theory of secrecy systems", *Bell System Tech. J.* (1949), vol. 28, 656-715.

[14] G.S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *J. Amer. Inst. Electrical Eng.* (1926), vol. 55, 109-115.

[15] A. Shamir, "How to share a secret," *Communications of the ACM* (1979), vol. 22, 612-613.

[16] R. J. McEliece and D. V. Sarwate, "On sharing secrets and Reed-Solomon codes," Commun. ACM, (1981), vol. 24, 583-584.

[17] J. L. Massey, "Minimal codewords and secret sharing," *Proc. 6th Joint Swedish-Russian Intl. Wkshp. Inform. Th.* (1993), 276-279.

ETH-Zürich, Switzerland, and Lund University, Sweden
Trondhjemsgade 3, 2.th
DK-2100 Copenhagen East, Denmark
JamesMassey@compuserve.com