Preprint of paper that was presented at the Fourth IMA Conference on Cryptography and Coding, which was held 13-15 December 1993 in Cirencester, England and that has since appeared in *Codes and Cyphers: Cryptography and Coding IV* (Ed. P. G. Farrell). Essex, England: Formara Ltd., 1995, pp. 33-47.

Some Applications of Coding Theory in Cryptography

J. L. Massey

Signal and Information Processing Laboratory
Swiss Federal Institute of Technology (ETH), Zürich

Abstract The applicability of techniques in coding theory to problems in cryptography is illustrated by examples concerning secret-sharing schemes with tailored access priveleges, the design of perfect local randomizers, the construction of t-resilient functions, and the quantization of the nonlinearity of boolean functions. Some novel coding concepts, in particular the notions of minimal codewords in linear codes and of a partition of the space of *n*-tuples based on nonlinear systematic codes akin to the coset partition for linear codes, are shown to be necessary to treat the cryptographic problems considered. The concepts of dual codes and dual distance as well as the relation between codes and orthogonal arrays are seen to play a central role in these applications of coding theory to cryptography.

1 Introduction

Coding theory, which had its inception in the late 1940's, is now generally regarded as a mature science. Cryptography on the other hand, at least in the public sector, is a science in the phase of early and rapid development. The thesis of this paper is that there are many problems in cryptography to which the well-developed techniques and extensive results of coding theory can be fruitfully applied. Our approach to demonstrating this thesis is anecdotal–we simply recount instances in our own research in cryptography where we have found coding theory to be an exceedingly useful tool.

Sometimes our application of coding theory to cryptography has forced us to introduce a new coding concept that may well be useful in its own right. One instance of this is the concept of "minimal codewords" in linear codes that we found necessary to treat secret-sharing schemes with tailored access priveleges as described below in Section 2. Our cryptographic applications of coding theory have also caused us to gain a deeper appreciation of the connection between codes, whether linear or nonlinear, and orthogonal arrays. This connection is described in Section 3 where again a new coding concept is required, in this case a partition of the space of n-tuples based on nonlinear systematic codes that is akin to the familiar coset partition for linear codes. In Sections 4 and 5 we bring the tools of Section 3 to bear on two problems of cryptographic interest: the design of "perfect local randomizers" and the construction of "t-resilient functions". In Section 6 we show how the venerable Reed-Muller codes provide a natural way to quantify the nonlinearity of a boolean function, a problem of considerable cryptographic interest. We conclude in Section 6 with a few remarks.

2 Secret Sharing with Tailored Access Priveleges

The age-old way to share a secret, such as the three-digit combination *17–14–92* to a combination lock with *100* positions on its dial would be to give part of the secret to each user, say *17* to Alice, *14* to Bob and *92* to Carol, telling each user the position of his or her share in the secret. There are two drawbacks to such secret sharing by dissection. First, the secret "leaks out" as more shares are learned. In the example, each share already reduces the uncertainty of its owner about the secret to only 10^4 out of the 10^6 possible combinations. Knowledge of two shares reduces this uncertainty further to only 10^2 combinations. Second, all shares are required to determine the secret. If Carol loses her share, Alice and Bob after combining their shares still must experiment to determine which of the remaining possible 10^2 combinations is the correct one. Such considerations led Shamir [1] and Blakely [2] independently in 1979 to formulate so-called *(S, T) threshold schemes for secret sharing* in which a secret is transformed into a list of S shares in such a manner that
(P1) knowledge of any T shares (where by "knowledge of a share" we will always mean knowledge of both its value and its position in the list of shares) reveals the secret, but
(P2) knowledge of $T - 1$ or fewer shares gives no information whatsoever about the secret.

With no loss of essential generality when considering secret-sharing schemes, we can and do suppose that the secret is an element of $GF(q)$, the finite field of q elements, as also is each of the shares. McEliece and Sarwate [3] have given an elegant formulation of (S, T) threshold schemes in terms of q-ary maximum-distance-separable (MDS) codes of block length $n = S + 1$ with q^k codewords, i. e., q-ary (n, k) codes with minimum distance $d = n - k + 1$, as follows: The secret x_1 is chosen as the first digit of the codeword. The code digits $x_2, x_3, \dots x_k$ are next chosen independently and uniformly at random in $GF(q)$ and the full codeword $\boldsymbol{x} = [x_1\ x_2\ \dots\ x_n]$ then computed. The share list is $x_2, x_3, \dots x_n$ so that $S = n - 1$. Because any k components of a codeword in an (n, k) MDS code form an information set, i. e., the values of these k components can be arbitrarily selected and then determine the entire codeword, property (P1) is satisfied for $T = k$ Because the first component x_1 can be included in an information set with any $k - 1$ other components, the values of these other components cannot constrain the value of x_1 and hence property (P2) is satisfied. When the MDS code is a Reed-Solomon code, this McEliece-Sarwate construction of an (S, T) threshold scheme is equivalent to the scheme suggested by Shamir [1] and formulated in terms of polynomial interpolation.

Sometimes one wishes to share a secret so that some users have greater privelege of access to it than do others. For instance, suppose that Alice is president of a bank in which Bob, Carol and David are tellers. One might desire that Alice together with any one of the tellers should have access to the secret, but that not even all three tellers together can gain access to the secret without Alice. For such cases, one can define the *access structure* of the secret-sharing scheme as consisting of all those sets of shares that determine the secret but that contain no proper subset that also determines the secret. For the example just mentioned, the access structure consists of the following sets of shares: {A, B}, {A, C}, and {A, D}, where A, B, C and D denote the shares for Alice, Bob, Carol and David, respectively. One can then formulate an *(S, Γ) access-structure scheme for secret sharing* in which a secret is transformed into S shares in such a manner that
(P3) knowledge of the shares in any set in Γ determines the secret, but
(P4) knowledge of the shares in a set not in Γ and having no subset in Γ reveals no information whatsoever about the secret.
The question then arises: which access structures can be realized by q-ary (n, k) codes in general and by linear codes in particular? To treat this problem for linear codes, we introduced in [4] what appears to be a new coding concept, viz. "minimal codewords", that we now review.

The q-ary n-tuple $\boldsymbol{x}$ is said to cover the q-ary n-tuple $\boldsymbol{x}'$ in case that in every coordinate where $\boldsymbol{x}'$ is non-zero, $\boldsymbol{x}$ is also non-zero. For example, the 3-ary 4-tuple $\boldsymbol{x} = [0\ 1\ 2\ 1]$ covers the 3-ary 4-tuple $\boldsymbol{x}' = [0\ 2\ 1\ 0]$. We will say that a codeword $\boldsymbol{x}$ in a q-ary linear (n, k) code is minimal if (i) $\boldsymbol{x}$ is a non-zero codeword whose leftmost non-zero component is a 1, and (ii) $\boldsymbol{x}$ covers no other codeword whose leftmost non-zero component is a 1.

Example 1 Let $GF(q)$ have characteristic 2 and consider the q-ary linear $(5, 3)$ code for which

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

is a generator matrix. One easily checks that $[1\ 1\ 1\ 0\ 0]$, $[1\ 1\ 0\ 1\ 0]$, $[1\ 1\ 0\ 0\ 1]$, $[0\ 0\ 1\ 1\ 0]$, $[0\ 0\ 1\ 0\ 1]$ and $[0\ 0\ 0\ 1\ 1]$ are all and only the minimal codewords of this code.

The most interesting property of minimal codewords is the following.

Proposition 1 Every codeword in a q-ary linear (n, k) code can be written as a linear combination of those minimal codewords that it covers.

Proof

Let $\mathbf{x}$ be a non-zero codeword. If $\mathbf{x}$ is minimal then there is nothing to prove. Otherwise, $\mathbf{x}$ covers a minimal codeword $\mathbf{x_1}$ and hence there is a constant c_1 such that the codeword $\mathbf{x} - c_1\mathbf{x_1}$ has smaller Hamming weight than does $\mathbf{x}$. Because $\mathbf{x}$ covers $\mathbf{x} - c_1\mathbf{x_1}$, it follows that if $\mathbf{x} - c_1\mathbf{x_1}$ is $\mathbf{0}$ or a minimal codeword then the proof is complete. Otherwise, $\mathbf{x} - c_1\mathbf{x_1}$ covers a minimal codeword $\mathbf{x_2}$ and hence there is a constant c_2 such that the codeword $\mathbf{x} - c_1\mathbf{x_1} - c_2\mathbf{x_2}$, which is still covered by $\mathbf{x}$, has smaller Hamming weight than does $\mathbf{x} - c_1\mathbf{x_1}$. Etc.

Every q-ary linear (n, k) code V with no idle components (i. e., with no components that are 0 in all codewords) determines an (S, Γ) access-structure secret-sharing scheme with $S = n - 1$ as follows. With no loss of essential generality, the secret is taken as the first digit x_1 of the codeword and the digits in some other specified $k - 1$ components, which together with the first component form an information set, are chosen independently at random. The codeword $\mathbf{x} = [x_1\ x_2\ ...\ x_n]$ is then computed and the share list taken as $x_2, x_3, ... x_n$ so that $S = n - 1$. The following result, taken from [4], determines the corresponding access structure.

Proposition 2 The access structure of the secret-sharing scheme correponding in the manner just specified to the q-ary linear (n, k) code V consists precisely of those share sets corresponding to those minimal codewords in the *dual* code $V^{\perp}$ having 1 as their first component in the manner that the share set specified by such a minimal codeword contains the share x_i $(2 \leq i \leq n)$ if and only if the i-th component of this codeword is non-zero.

Example 2 If we take the matrix in Example 1 to be a parity-check matrix for the code V, i. e., a generator matrix for the dual code $V^{\perp}$, then the minimal codewords with first component 1 in this dual code are $[1\ 1\ 1\ 0\ 0]$, $[1\ 1\ 0\ 1\ 0]$ and $[1\ 1\ 0\ 0\ 1]$. By Proposition 2, the access structure consists of the share sets $\{x_2, x_3\}$, $\{x_2, x_4\}$ and $\{x_2, x_5\}$. Letting x_2, x_3, x_4 and x_5 be the shares for Alice, Bob, Carol and David, respectively, we see that we have realized the desired access structure for the bank in which Alice is president.

Proof of Proposition 2

Suppose that $\{x_1, x_2, ... x_t\}$ is a share set in the access structure realized by the code V. Because V is linear, it must be the case that the secret x_1 is then determined as a linear combination of the shares in this share set, i. e., there must exist constants $a_2, a_3, ... a_t$ (all necessarily non-zero) such that

$$x_1 = a_2x_2 + a_3x_3 + \ldots + a_tx_t \qquad (1)$$

or, equivalently, such that

$$x_1 - a_2x_2 - a_3x_3 -. .. - a_tx_t = 0$$

or, again equivalently, such that $[1\ -a_1\ -a_2\ \ldots\ -a_n]$ is a codeword in the dual code $V^{\perp}$. This codeword must be minimal for otherwise, by Proposition 1, it covers a minimal codeword for which (1) can be written with at least one of $a_2, a_3, \ldots a_t$ equal to 0 and hence a proper subset of $\{x_2, x_3, \ldots x_t\}$ would reveal the secret x_1, contradicting that this share set is in the access structure. Conversely, any minimal codeword with leading component 1 in the dual code $V^{\perp}$, say $[1\ -a_1\ -a_2\ \ldots\ -a_n]$ with $a_1, a_2, \ldots a_t$ all non-zero, corresponds to a relation (1) and hence implies that $\{x_2, x_3, \ldots x_t\}$ suffice to determine x_1, but the minimality of this codeword implies that (1) cannot be written for any other choice of $a_1, a_2, \ldots a_t$ and hence no proper subset of $\{x_2, x_3, \ldots x_t\}$ suffices to determine x_1 so that this share set is indeed in the access structure.

Among the open problems suggested by Proposition 2 are the following: Give an algorithm that, for a desired access structure, either finds a linear code V that realizes this access structure in the manner of Proposition 2 or else shows that no such code exists. Show that any access structure that can be realized in some manner can also be realized by a linear code in the manner of Proposition 2 or else give a counterexample to this assertion.

3 Codes and Orthogonal Arrays

In this section we collect some definitions and results that will be needed in the two following sections. We begin by recalling a result from [5] and then proving an analogous result for dual codes.

Lemma 1 If the $k \times n$ q-ary matrix $\boldsymbol{G}$ is the generator matrix of a q-ary linear (n, k) code V, then the minimum distance d of V is the smallest number of columns that can be removed from $\boldsymbol{G}$ to yield a matrix with rank less than k.

Proof Suppose that deleting t columns of $\boldsymbol{G}$ gives a matrix $\tilde{\boldsymbol{G}}$ of rank less than k. Then there is a non-zero k-tuple $\boldsymbol{u}$ such that $\boldsymbol{u}\tilde{\boldsymbol{G}} = \boldsymbol{0}$ and hence such that the non-zero codeword

$\boldsymbol{uG}$ has Hamming weight at most t. But the minimum weight of the non-zero codewords in a linear code coincides with its minimum distance d so we must have $t \geq d$. Choosing the t deleted columns of $\boldsymbol{G}$ to correspond to the non-zero components of a codeword $\boldsymbol{uG}$ of weight d shows that $t = d$ suffices.

Lemma 2 If the $k \times n$ q-ary matrix $\boldsymbol{G}$ 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 $\boldsymbol{G}$ that form a $k \times t$ matrix with rank less than t.

Proof

We first note that t ($t \geq 1$) chosen columns of $\boldsymbol{G}$ form a submatrix of rank less than t if and only if these columns are linearly dependent, i. e., if and only if there is a non-zero n-tuple $\boldsymbol{x}$ that is non-zero only in the t corresponding components and gives $\boldsymbol{Gx^T} = \boldsymbol{0^T}$ or, equivalently, gives $\boldsymbol{xG^T} = \boldsymbol{0}$. But $\boldsymbol{G}$ is a parity-check matrix for the dual code $V^{\perp}$ and hence $\boldsymbol{xG^T} = \boldsymbol{0}$ is just the condition that $\boldsymbol{x}$ be a codeword in $V^{\perp}$. Because $\boldsymbol{x}$ is non-zero, it has Hamming weight at least $d^{\perp}$ and hence $t \geq d^{\perp}$. Choosing the $t = d^{\perp}$ columns of $\boldsymbol{G}$ to correspond to the non-zero components of a minimum-weight codeword in $V^{\perp}$ shows that these $t = d^{\perp}$ columns of $\boldsymbol{G}$ are indeed linearly dependent.

An *orthogonal array* $OA_{\lambda}(t, n, q)$ is a $\lambda q^t \times n$ array of q-ary symbols 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 said to be *simple*. An orthogonal array $OA_{\lambda}(t, n, q)$ with $t \geq 2$ is of course also an $OA_{\lambda q}(t - 1, n, q)$ so that the maximum t for which a given array is an orthogonal array is the parameter of interest.

Example 3 The 4×3 array

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

whose rows are the codewords of a binary linear $(3, 2)$ code is a simple orthogonal array $OA_1(2, 3, 2)$ since each binary 2-tuple appears in exactly 1 row in every choice of 2 columns of the array. This $t = 2$ is the maximum t for which this 4×3 array is an orthogonal array.

Proposition 3 The maximum t for which a $q^k \times n$ q-ary 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] is $t = d^\perp - 1$.

Remark For $k = n$, the dual code has only one codeword; for the lemma still to hold in this case, one must adopt the convention that $d^\perp = n + 1$. For $d^\perp = 1$, the code V has one or more idle components and hence its codewords cannot be the rows of an orthogonal array.

Proof

Let $\tilde{\boldsymbol{G}}$ be the submatrix formed by some choice of t columns of a generator matrix $\boldsymbol{G}$ for V. Then $\tilde{\boldsymbol{x}} = \boldsymbol{u}\tilde{\boldsymbol{G}}$ is a mapping from the information k-tuple $\boldsymbol{u}$ to those t components of the codeword that correspond to the chosen columns. This mapping is surjective (or "onto") if and only if $\tilde{\boldsymbol{G}}$ has rank t. Moreover, if this mapping is surjective, its linearity guarantees that there are the same number of solutions $\boldsymbol{u}$ to the equation $\tilde{\boldsymbol{x}} = \boldsymbol{u}\tilde{\boldsymbol{G}}$ for each of the q^t choices of $\tilde{\boldsymbol{x}}$, i. e., the same number of codewords that have this pattern $\tilde{\boldsymbol{x}}$ in the chosen t components. The proposition now follows directly from Lemma 2.

A q-ary nonlinear (n, k) code C of course has no dual. However, one can define a dual distance $d^\perp$ for such a code to be the smallest positive integer i such that $B_i > 0$ where $(B_0, B_1, \ldots B_n)$ is the MacWilliams' transform of the average distance distribution $(A_0, A_1, \ldots A_n)$ between the codewords of C, cf [6, pp. 135-141]. For a linear code, this dual distance is the minimum distance of the dual code. The following remarkable fact was first proved by Delsarte [7], cf. [6. p. 139] and cf. [8] for a recent and masterful treatment of this subject.

Proposition 3+ Proposition 3 also holds for a q-ary nonlinear (n, k) code C when $d^\perp$ is taken to be the dual distance of C.

Stinson [9] has defined a *large set of simple orthogonal arrays* $OA_\lambda(t, n, q)$ to be a collection of q^{n-t}/λ such arrays with the property that every q-ary n-tuple appears as a row in exactly one of the arrays in the collection.

If S is the set of rows in a simple orthogonal array $OA_\lambda(t, n, q)$ and $\boldsymbol{x}$ is a q-ary n-tuple, then it is easy to see that the array whose rows form the translated set $\boldsymbol{x} + S$ (i. e., the set formed by adding $\boldsymbol{x}$ to each n-tuple in S) is also a simple orthogonal array $OA_\lambda(t, n, q)$. If S is a linear code V, then $\boldsymbol{x} + S = \boldsymbol{x} + V$ is a coset of V. Because cosets are either identical or disjoint, we obtain the following result immediately from Proposition 3.

Proposition 4 The maximum t such that the collection of q^{n-k} arrays, whose rows are the codewords in a q-ary linear (n, k) code V with $k < n$ and $d^{\perp} > 1$ and in its $q^{n-k} - 1$ proper cosets, is a large set of simple orthogonal arrays $OA_{\lambda}(t, n, q)$ is $t = d^{\perp} - 1$, where $d^{\perp}$ is the minimum distance of the dual code $V^{\perp}$.

Because there is nothing corresponding to a coset partition for a nonlinear code, it is not obvious that Proposition 4 can be extended to nonlinear (n, k) codes. This quandry was, however, resolved in [5] for the important special case of systematic nonlinear (n, k) codes. A q-ary (n, k) code (linear or not) is said to be *systematic* if it has at least one information set, i. e., k components whose values can be arbitrarily selected and then determine the entire codeword. All linear codes are systematic. Not all nonlinear (n, k) codes are systematic, but all the interesting nonlinear q-ary (n, k) codes that have been found to date have been systematic. If C is a systematic q-ary (n, k) code and if E is the set of all q^{n-k} q-ary n-tuples that contain only zeroes in the components corresponding to an information set for C, then the sets $\boldsymbol{e} + C$ for $\boldsymbol{e} \in E$ are disjoint and exhaust the set of all q^n q-ary n-tuples. If C is an $OA_{\lambda}(t, n, q)$, then so is $\boldsymbol{e} + C$. The following result from [5] thus follows immediately from Property 3+.

Proposition 4+ Proposition 4 holds also for a *systematic* q-ary nonlinear (n, k) code C when the cosets of V are replaced by the sets $\boldsymbol{e} + C$ for $\boldsymbol{e} \in E$, where E is the set of all q^{n-k} q-ary n-tuples that contain only zeroes in the components specifying an information set for C.

4 Local Randomization

Cryptographers often wish to "stretch" randomness. For instance, a cryptographer designing the *running-key generator* for an additive stream cipher would like this device to transform the k-bit secret key (which can be selected uniformly at random by coin tossing) into a much longer sequence of n bits, the so-called *running key*, such that this longer sequence still appears to an attacker to be completely random. (This running key would then be added bit-by-bit in $GF(2)$ to the plaintext to produce the ciphertext.) Because the running key cannot have greater entropy than the secret key that determines it, a completely random running key is impossible for $n > k$. However, short segments of the running key could still be completely random even when $n >> k$. Motivated by such considerations and by a related complexity-theoretic definition of Schnorr [10], Maurer and Massey [11] defined an injective (or "one-to-one") function f from k q-ary digits to n q-ary digits ($n > k$) to be a *perfect local randomizer of order t* if choosing the k input digits $u_1, u_2, \ldots u_k$ 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.

Suppose now that we select some t components of the output $[x_1\ x_2\ ...\ x_n] = f(u_1, u_2, ...\ u_k)$. For these components to be uniformly random, each of the q^t possible values of these t components must occur for eactly $\lambda = q^k/q^t = q^{k-t}$ of the possible values of the argument $[u_1\ u_2\ ...\ u_k]$. In other words, in the $q^k \times n$ array whose rows are the values of $[x_1\ x_2\ ...\ x_n]$ (i. e., are the range of the function f), each possible value of these t components must appear in $\lambda = q^{k-t}$ rows. We summarize.

Lemma 3 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 $q^k \times n$ array having as its rows the range of f is an orthogonal array $OA_\lambda(t, n, q)$.

But any q^k distinct q-ary n-tuples can be taken to be the codewords of a q-ary (n, k) code (not necessarily linear). Moreover, an injective function from k q-ary digits to n q-ary digits is an *encoder* for such a code so that the following proposition from [11] now follows directly from Lemma 3 and Proposition 3+.

Proposition 5 An injective function from k q-ary digits to n q-ary digits is a perfect local randomizer of order t ($t \geq 1$) if and only if it is the encoder for a q-ary (n, k) code (not necessarily linear) with dual distance $d^\perp$ greater than t.

It was pointed out in [5] that, because the Kerdock codes, cf. [6, p. 456], are binary (n, k) nonlinear codes with $n = 2^{r+1}$, $k = 2r+2$ and $d^\perp = 6$ for odd r at least 3, and because for such r there is no binary (n, k) linear code with the same n and with $k = n - 2r - 2$ and minimum distance $d = 6$ (these are the parameters of the nonlinear Preparata codes that are known to have larger d than the best linear codes with the same n and k, cf. [12]), it then follows from Proposition 5 that there are infinitely many choices of n and k such that there exists a nonlinear perfect local randomizer of order t from k bits to n bits but there exists no such linear perfect local randomizer, which answers a question that had been raised in [11].

5 Resilient Functions

Cryptographers often seek to prevent divide-and-conquer attacks on their systems by ensuring that local constraints on an input result in no constraints on the output. Such considerations led Chor, Goldreich, Hastad, Friedman, Rudich and Smolensky [13] and, later but independently, Bennett, Brassard and Robert [14] to introduce the notion of resilient functions. 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 digits are fixed and the values of the other $n - t$ input digits chosen uniformly at random, all $n - k$ output digits are uniformly random.

Example 4 The linear function $f(x_1, x_2, x_3) = x_1 + x_2 + x_3$ from $n = 3$ binary digits to $n - k = 1$ binary digit has the function table:

x_1 x_2 x_3	$f(x_1,x_2,x_3)$
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1

This is a 2-resilient function since the set of arguments $[x_1\ x_2\ x_3]$ giving $f(x_1, x_2, x_3) = 0$ contains each possible value of x_1 and x_3, for instance, exactly once as does also the set of arguments giving $f(x_1, x_2, x_3) = 1$; hence, when x_1 and x_3 are fixed to any values, $f(x_1, x_2, x_3)$ will equal to 1 with probability $1/2$ when x_2 is chosen by coin tossing.

One sees readily from this example that the condition for $f(x_1, x_2, \dots x_n) = [y_1\ y_2 \dots y_{n-k}]$ to be t-resilient is that there exists a positive integer λ such that, for each choice of the output $[y_1\ y_2 \dots y_{n-k}]$, the set $f^{-1}(y_1, y_2, \dots y_{n-k})$ (i. e., the set of those arguments $[x_1\ x_2 \dots x_n]$ that give the function value $[y_1\ y_2 \dots y_{n-k}]$) has the property that, for every choice of t components of the argument, each of the q^t possible values of these t components appears exactly λ times. We summarize as follows.

Lemma 4 A function $f(x_1, x_2, \dots x_n) = [y_1\ y_2 \dots y_{n-k}]$ from n q-ary digits to $n\text{-}k$ q-ary digits is t-resilient if and only if the collection of the q^{n-k} sets $f^{-1}(y_1, y_2, \dots y_{n-k})$ determined by the possible choices of $[y_1\ y_2 \dots y_{n-k}]$ is a large set of simple orthogonal arrays $OA_\lambda(t, n, q)$.

If $\boldsymbol{H}$ is an $(n - k) \times n$ parity-check matrix for a q-ary linear (n, k) code V, then the linear function $f(x_1, x_2, \dots x_n) = [x_1\ x_2 \dots x_n]\ \boldsymbol{H}^T$ has the property that the inverse function $f^{-1}(y_1, y_2, \dots y_{n-k})$ is a bijective mapping from the q^{n-k} q-ary $(n - k)$-tuples to the collection of all cosets of V. (This linear function f is called a "syndrome former" in coding theory.) The following result is thus an immediate consequence of Lemma 4 and Proposition 4.

Proposition 6 If $\boldsymbol{H}$ is an $(n - k) \times n$ parity-check matrix for a q-ary linear (n, k) code V, then the maximum t such that the linear function $f(x_1, x_2, \dots x_n) = [x_1\ x_2 \dots x_n]\ \boldsymbol{H}^{\boldsymbol{T}}$ is t-resilient is $t = d^{\perp} - 1$, where $d^{\perp}$ is the minimum distance of the dual code $V^{\perp}$.

In fact, the above argument together with Proposition 4+ shows the truth of the following stronger result from [5].

Proposition 6+ If the function $f(x_1, x_2, \dots x_n) = [y_1\ y_2 \dots y_{n-k}]$ from n q-ary digits to $n - k$ q-ary digits has the property that the sets $f^{-1}(y_1, y_2, \dots y_{n-k})$ for the q^{n-k} choices of $[y_1\ y \dots y_{n-k}]$ are the collection of sets $\boldsymbol{e} + C$, $\boldsymbol{e} \in E$, where C is a systematic q-ary (n, k) code and E is the set of all q^{n-k} q-ary n-tuples that contain only zeroes in the components corresponding to an information set for C, then the maximum t such that f is t-resilient is $t = d^{\perp} - 1$, where $d^{\perp}$ is the dual distance of C.

This proposition was used in [5] together with the properties of the nonlinear Kerdock codes to show the existence of infinitely many nonlinear t-resilient functions with t greater than can be obtained by linear functions for the same n and $n - k$, thereby disproving an earlier conjecture [13] that not even one such nonlinear function existed.

6 Quantifying Nonlinearity of Boolean Functions

Linearity is often said to be the cryptographer's curse (or the cryptanalyst's blessing). Indeed cryptographers often speak of selecting some "highly nonlinear" transformation for use within their ciphers. How to quantify nonlinearity is a task that has often occupied cryptographers. One approach stems from the so-called algebraic normal form of a boolean function. Suppose that f is a boolean function of n binary variables. We will consider such a function as a mapping from $GF(2)^n$ to $GF(2)$. Such a function can be written uniquely as

$$f(x_1, x_2, \dots x_n) = a_0 1 + a_1 x_1 + \dots a_n x_n + a_{12} x_1 x_2 + a_{13} x_1 x_3 + \dots a_{n-1,n} x_{n-1} x_n + \dots a_{12\dots n} x_1 x_2 \dots x_n \qquad (2)$$

where the coefficients $a_0, a_1, \dots a_n, a_{12}, a_{13}, \dots a_{n-1,n}, \dots a_{12\dots n}$ and all operations are in $GF(2)$. This expression for f is called its *algebraic normal form* (ANF) and is frequently used to define the order of nonlinearity of f. The contant term 1 in (2) is the zero-order term; the variables $x_1, x_2, \ldots x_n$ are the first-order terms, the products of pairs of variables $x_1x_2, x_1x_3, \ldots x_{n-1,n}x_n$ are the second-order terms; etc. The *nonlinear order* of f is then defined as the order of the term of maximum order with a non-zero coefficient in the ANF of f (and the zero function is said to have nonlinear order $-\infty$). For example, the function $f(x_1, x_2, x_3) = 1 + x_1x_2 + x_2x_3$ has nonlinear order 2. Nonlinear order plays an important role in determining the "linear complexity" of a sequence formed by using the function to combine the outputs of n linear feedback shift-registers, cf. [15]. That nonlinear order is not an entirely satisfactory measure of nonlinearity, however, can be inferred from the example $f(x_1, x_2, x_3) = x_1x_2x_3$, which has the maximum possible nonlinear order, 3, for a function of $n = 3$ variables. But this function differs from the least nonlinear function (the zero function) in its value for only one argument, namely $[x_1\ x_2\ x_3] = [1\ 1\ 1]$ so that it is "very close" to this least nonlinear function.

More insight can be gained into the ANF of a function if we consider the function tables of the monomial functions of order 0 and 1. For $n = 3$, one obtains the following function tables:

$[x_1\ x_2\ x_3]$	$[0\ 0\ 0]$	$[0\ 0\ 1]$	$[0\ 1\ 0]$	$[0\ 1\ 1]$	$[1\ 0\ 0]$	$[1\ 0\ 1]$	$[1\ 1\ 0]$	$[1\ 1\ 1]$
1	1	1	1	1	1	1	1	1
x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1

A coding theorist looking at this 4×8 array of binary digits will immediately recognize it at the generator matrix of the $(8, 4)$ first-order Reed-Muller code in precisely the form suggested by Reed [16]. It follows from (2) that the function table of any function $f(x_1, x_2, x_3)$ of nonlinear order 1 or less is just a linear combination of the rows of this matrix, i. e., a codeword in this first-order Reed-Muller code. Rueppel [17, p. 129], who noted this fact, used the Hamming distance of the function table of a function from the nearest codeword in the first-order Reed-Muller code of length 2^n, divided by n for a convenient normalization, as a measure of how accurately the function could be approximated by a linear or affine function, i. e., by a function of nonlinear order 1 or less, and used this approach to explore the nonlinearity of the functions that appear in the S-boxes ("substitution boxes") of the Data Encryption Standard (DES).

Nothing, however, requires us to stop at the monomials of nonlinear order 1 or less in our examination of boolean functions. For the above example with $n = 3$, the mononials of order 2 have the following function tables:

x_1x_2	0	0	0	0	0	0	1	1
x_1x_3	0	0	0	0	0	1	0	1

x_2x_3	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*1*

A coding theorist will again immediately recognize these three binary rows as those that must be adjoined to the generator matrix for the *(8, 4)* first-order Reed-Muller code to obtain the generator matrix of the *(8, 7)* second-order Reed-Muller code in precisely the form suggested by Reed [16]. Again it follows from (2) that the function table of any function $f(x_1, x_2, x_3)$ of nonlinear order *2* or less is just a linear combination of the rows of this latter generator matrix, i. e., a codeword in this second-order Reed-Muller code. The obvious extension can be summarized as follows.

Proposition 7 The codewords of the *i*-th order ($0 \le i \le n$) binary Reed-Muller code of length 2^n, when the generator matrix is taken in the Reed form [16], are precisely the function tables of the boolean functions $f(x_1, x_2, ... x_n)$ of nonlinear order i or less with the argument values $[x_1\ x_2\ ...\ x_n]$ taken in natural binary order.

This result suggests that what we will call the *nonlinearity spectrum* $(\delta_0, \delta_1, ... \delta_n)$ of a non-zero boolean function $f(x_1, x_2, ... x_n)$, where δ_i is the normalized Hamming distance of the function table to the nearest codeword in the *i*-th order Reed-Muller code of length 2^n, gives a much truer picture of the function's nonlinearity than does merely its nonlinear order (which is the smallest *i* such that $\delta_i = 0$) or merely its normalized distance δ_1 to linear and affine functions. For example, $f_1(x_1, x_2, x_3) = 1 + x_1 + x_2x_3$ and $f_2(x_1, x_2, x_3) = 1 + x_2x_3$ have the respective function tables:

$f_1(\mathbf{x})$	*1*	*1*	*1*	*0*	*0*	*0*	*0*	*1*
$f_2(\mathbf{x})$	*1*	*1*	*1*	*0*	*1*	*1*	*1*	*0*

One easily checks that the nonlinearity spectra of f_1 and f_2 are $(1/2, 1/4, 0, 0)$ and $(1/4, 1/4, 0, 0)$ respectively, showing that f_1 is indeed "more nonlinear" than f_2, although both functions have nonlinear order *2* and distance $\delta_1 = 1/4$ to the linear and affine functions.

We note that computing the nonlinearity spectrum of a boolean function $f(x_1, x_2, ... x_n)$ requires one to do complete nearest-codeword decoding of the Reed-Muller codes of length 2^n. This is not a great problem for the small values of *n*, typically $n \le 6$, usually encountered in cryptographic systems, but it is already a daunting problem for *n* as small as *10*. In such cases, in place of the true (normalized) Hamming distance δ_i to a codeword in the *i*-th order Reed-Muller code, one might be forced to replace δ_i with the distance $\tilde{\delta}_i$ to the codeword found by Reed's simple majority-decoding algorithm [16], which is guaranteed to decode to the nearest codeword whenever this codeword is less than half the minimum distance of the code from the given word (which word in this case is a function table).

7 Concluding Remarks

That we have not given more examples of the application of coding theory to cryptography [or, perhaps better said, of the interplay between coding theory and cryptography] is due more to the need to hold this paper to a reasonable length than to the paucity of such examples, even when we restrict ourselves to those that have arisen in our own research. The sciences of coding and cryptography appear to us to be intrinsically intertwined. As the former science is in a higher stage of deveopment, i. e., the theory is more extensive and cohesive, its study is a natural springboard for those who want to dive into the much-less-well-charted waters of cryptography. We would be pleased if this paper encourages a few to take this plunge.

References

1. Shamir, A. (1979). How to share a secret. *Communications of the ACM*, **22**, 612-613.

2. Blakley, G. R. (1979). Safeguarding cryptographic keys. *Proceedings AFIPS 1979 National Computer Conference*, 48, 313-317.

3. McEliece, R. J. and Sarwate, D. V. (1981). On sharing secrets and Reed-Solomon codes. *Communications of the ACM*, **24**, 583-584.

4. Massey, J. L. (1993). Minimal codewords and secret sharing. *Proceedings 6th Joint Swedish-Russian International Workshop on InformationTheory*, 276-279.

5. Stinson, D. R. and Massey, J. L. (1994). An infinite class of counterexamples to a conjecture concerning non-linear resilient functions," *Journal of Cryptology,* to appear.

6. MacWilliams, F. J. and Sloane, N. J. A. (1977). *The theory of error-correcting codes.* North-Holland.

7. Delsarte, P. (1972). Bounds for unrestricted codes, by linear programming. *Philips Research Reports Supplement*, **27**, 272-289.

8. Delsarte, P. (1994). Application and generalization of the MacWilliams transform in coding theory. *Proceedings 15th Benelux Symposium on Information Theory*, 9-44.

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

10. Schnorr, C. P. (1988). On the construction of random number generators and random function generators. *Lecture Notes in Computer Science*, **434**, 423-428.

11. Maurer, U. M. and Massey, J. L. (1991). Local randomness in pseudorandom sequences," *Journal of Cryptology*, **4**, 135-149.

12. Goethals, J.-M. and Snover, S. L. (1972). Nearly perfect binary codes. *Discrete Mathematics*, **3**, 65-88.

13. Chor, B., Goldreich, O., Hastad, J., Friedman, J., Rudich, S. and Smolensky, R. (1985). The bit extraction problem or *t*-resilient functions. *Proceedings 26th IEEE Symposium on Foundations of Computer Science*, 396-407.

14. Bennett, C. H., Brassard, G. and Robert, J.-M. (1988). Privacy amplification by public discussion. *SIAM Journal of Computing*, **17**, 210-229.

15. Key, E.L. (1976). An analysis of the structure and complexity of nonlinear binary sequence generators. *IEEE Transactions on Information Theory*, **22**, 732-736.

16. Reed, I. S. (1954). A class of multiple-error-correcting codes and the decoding scheme. *IRE Transactions on Information Theory*, **4**, 38-49.

17. Rueppel, R. A. (1986). *Analysis and design of stream ciphers*. Springer.