A New Upper Bound on the Block Error Probability After Decoding Over the Erasure Channel

Frédéric Didier

Abstract—**Motivated by cryptographic applications, we derive a new upper bound on the block error probability after decoding over the erasure channel. The bound works for all linear codes and is in terms of the generalized Hamming weights. It turns out to be quite useful for Reed–Muller codes for which all the generalized Hamming weights are known whereas the full weight distribution is only partially known. For these codes, the error probability is related to the cryptographic notion of algebraic immunity. We use our bound to show that the algebraic immunity of a random balanced m-variable Boolean function is of order $\frac{m}{2}(1-o(1))$ with probability tending to 1 as m goes to infinity.**

Index Terms—**Algebraic immunity, Boolean functions, erasure channel, generalized Hamming weights, Reed–Muller codes.**

I. Introduction

In this paper, we exploit the information given by the generalized Hamming weights of a linear code to derive the following upper bound on the block error probability after decoding over the erasure channel.

Theorem 1: Let $\mathcal{C}$ be a linear binary code of length n, dimension k and generalized Hamming weights $(d_i)_{i\in[\![1,k]\!]}$. We assume that transmission takes place over the erasure channel and that exactly e erasures have occurred. Then, the probability that not all e erasures are recoverable is upper bounded by

$$P_{\text{err}} \leq \prod_{i=e+1}^{n}\left(\frac{i-d_1}{i}\right)\prod_{i=2}^{k}\left(\frac{d_i}{d_i-d_1}\right). \tag{1}$$

The attractive feature of this result is that it can be applied to a rather large family of codes with unknown distance distribution, but for which a simple lower bound on the generalized weights is known. We call such codes "rate consistent" in what follows. Applying the previous theorem to such codes yields

Theorem 2: Using the same notation as in Theorem 1, if the code $\mathcal{C}$ is rate-consistent then

$$\ln(P_{\text{err}}) \leq d\left[\frac{k}{n}\left(\ln\frac{n}{d}+3\right)-\ln\frac{n}{e}\right] \tag{2}$$

where $d\ (= d_1)$ is the minimal distance of $\mathcal{C}$.

The rate-consistency property (see Section V) appears to be quite general. We have proved that all cyclic codes, self-dual codes, Reed–Muller codes, generalized Reed–Muller codes and geometric Goppa codes are rate-consistent.

Theorems 1 and 2 are particularly useful for Reed–Muller codes since they give the sharpest bounds known so far on their erasure recovering capacity. In particular, Theorem 2 shows that Reed–Muller codes with logarithmically vanishing rate are able to correct a constant fraction of erasures.

This result has an important cryptographic application because the decoding of Reed–Muller codes over the erasure channel is related to the modern algebraic attacks on a cryptosystem. These attacks introduced in [3] are quite powerful and have raised a lot of interest recently. The complexity of these attacks mainly depends on what is called the algebraic immunity of the Boolean function involved in the ciphering process ([14], [4], [1], [5], [6]). For a m-variable Boolean function, the algebraic immunity is always smaller than $(m+1)/2$. It was proven in [14] that for a balanced Boolean function it is greater than 0.22 m with probability tending to 1 as m goes to infinity. Using Theorem 2 for the Reed–Muller codes, we will be able to refine substantially this lower bound with the following theorem.

Theorem 3: The algebraic immunity r of a random balanced Boolean function in m variables satisfies for all $a<1$

$$\frac{m}{2}-\sqrt{\frac{m}{2}\ln\left(\frac{m}{2a\ln 2}\right)} \leq r \leq \frac{m+1}{2} \tag{3}$$

with probability tending to 1 as m goes to infinity.

In other words, we have proved that the algebraic immunity of almost all balanced Boolean functions in m variables is of order $\frac{m}{2}(1-o(1))$ as m tends to infinity. This result was conjectured in [2] but this is the first theoretical proof. In addition, Theorem 2 can be used to analyze rigorously various algorithms for computing the algebraic immunity, see [7].

The paper is organized as follows. First, we recall some definitions and explain the relation between Reed–Muller codes and cryptography. Then we recall known results obtained by a union bound argument and expose the proof of our new bounds given in Theorems 1 and 2. In both cases, we will apply these results to Reed–Muller codes and examine what we obtain in terms of algebraic immunity. Since we are interested in cryptographic applications, we will not only give the asymptotic form of the bounds but also their numerical values for Reed–Muller codes of small length.

II. Basic Definitions

This section is basically an enumeration of the notions and notation that we will use throughout this article. Let $\mathcal{C}$ be a linear code of length n, dimension k and minimal distance d.

Manuscript received July 11, 2005; revised April 24, 2006.

The author is with the Institut National de Recherche en Informatique et Automatique (INRIA), Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay CEDEX, France (e-mail: frederic.didier@inria.fr).

Communicated by R. J. McEliece, Associated Editor for Coding Theory.
Digital Object Identifier 10.1109/TIT.2006.881719

0018-9448/$20.00 © 2006 IEEE

A. The Issue

Given an erasure pattern, that is a binary word of length n and Hamming weight e, we are interested in the subspace $\mathcal{I}$ of the codewords included in the pattern. By included, we mean the inclusion of the support

$$\operatorname{supp}(c) := \{i, c_i \neq 0\} \quad c \in \{0,1\}^n. \tag{4}$$

Assuming that a 1 bit in the erasure pattern corresponds to an erased position, we can only retrieve an erased codeword modulo $\mathcal{I}$. In other words, unambiguous decoding is only possible if the dimension of $\mathcal{I}$ is 0.

We want to find, for a given e, an upper bound on the proportion of the erasure patterns that give a subspace of dimension different from 0. We will call this proportion the error probability and denote it by P_{err}.

Our channel model is slightly different from the classic erasure channel because we assume a fixed number of erasures. However, our results can be used to derive bounds for a given erasure probability p using the formula

$$P_{\text{err}}(p) = \sum_{e=0}^{n} \binom{n}{e} p^e (1-p)^{n-e} P_{\text{err}}(e). \tag{5}$$

B. Generalized Hamming Weights

The generalized Hamming weights (d_i) of a code are by definition, for $i \in [\![1,k]\!]$

$$d_i := \min_{v \in V_i}(|\operatorname{supp}(v)|) \tag{6}$$

where V_i is the set of all i-dimensional subspaces of $\mathcal{C}$. The support of a subspace V corresponds to the set of not-always-zero bit positions in V, that is

$$\operatorname{supp}(V) := \{i, \exists c \in V\ c_i \neq 0\} = \bigcup_{c \in V} \operatorname{supp}(c) \tag{7}$$

d_1 corresponds to the minimal distance d of the code. d_k is the cardinality of the code support and we will assume that it is equal to n.

C. Boolean Functions

An m-variable Boolean function is a mapping from $\{0,1\}^m$ to $\{0,1\}$. It is well known that such a function f can be written in an unique way as an m-variable polynomial over $\mathbf{F}_2$ where the degree in each variable is at most one using the Algebraic Normal Form (ANF). The degree of f is by definition the degree of this polynomial.

By listing the images of f over all possible values of the variables, we can also view it as a binary word of length $n = 2^m$. Using this representation, we define the Hamming weight of f as the Hamming weight of the associated binary word.

A balanced Boolean function is a function with Hamming weight equal to half its length, that is $n/2$. In cryptography, the most commonly used Boolean functions are balanced, which is why we focus on such functions in our experiments.

D. Reed–Muller Codes

The Reed–Muller code RM(r,m) of order r is the linear subspace composed by m-variable Boolean functions of degree smaller than or equal to r viewed as binary words of length n. The minimal distance of the code is (see, e.g., [13])

$$d = 2^{m-r}. \tag{8}$$

The monomials of degree smaller than or equal to r form a basis of RM(r,m). It follows that the code dimension is given by

$$k = \sum_{i=0}^{r} \binom{m}{i}. \tag{9}$$

We do not know the weight distribution of Reed–Muller codes in general (see Section III-B for more details), but the generalized Hamming weights are known and are detailed in Appendix.

E. Algebraic Immunity of Boolean Functions

We give here an interpretation of the decoding problem of a Reed–Muller code RM(r,m) over the erasure channel. The interpretation is in terms of the algebraic immunity of a certain Boolean function associated with the erasure pattern.

More precisely, we view an erasure pattern of length $n = 2^m$ as the image list of a Boolean function f in m variables. If we have another m-variable Boolean function g such that $\operatorname{supp}(g) \subset \operatorname{supp}(f)$ then we have

$$fg = g \text{ where } fg(x) = f(x)g(x) \quad \forall\, x \in \{0,1\}^m. \tag{10}$$

In other words, the subspace $\mathcal{I}$ of RM(r,m) codewords included in the erasure pattern may be viewed as the subspace formed by the functions of degree less than or equal to r invariant by f.

This means that a bound on the error probability becomes a bound on the probability that a random function of weight e admits a nontrivial invariant. The smallest r for which f or its complement $1+f$ admit a nontrivial invariant is by definition the algebraic immunity of f.

As mentioned in the introduction, this notion quantifies the immunity of a cryptosystem to some recent algebraic attacks. These attacks try to break a cryptosystem by solving an algebraic system involving the key bits. The equations involved often depend on a Boolean function f used in the ciphering process. The idea is that f or $1+f$ can be replaced by their invariants to obtain a system of lower degree. The complexity of the attack depends on the degree of the system which is nothing but the algebraic immunity of f.

III. A Simple Union Bound

We address here the following issues:

- How do the Reed–Muller codes behave experimentally on the erasure channel with constant number of erasures?
- What do the standard bounds in coding theory give in this case?

A. Experimental Results

We recall that given an erasure pattern there are $n-e$ non-erased positions, so when $k > n-e$ unambiguous decoding is not possible. For the limit case $k = n - e$ the proportion of decodable patterns corresponds to the proportion of information sets in the code.

It is well known that the error probability P_{err} of a random linear code is given by

$$P_{\text{err}} = 1 - \prod_{i=n-e-k+1}^{n-e} \left(1 - \frac{1}{2^i}\right). \tag{11}$$

From this expression we see that for $n \to \infty$, the proportion of noninformation sets becomes roughly 0.711 (see, e.g., [9]) and the error probability decreases exponentially fast in $n - e - k$.

Performing simulations for some small self-dual Reed–Muller codes with a balanced erasure pattern $(e = n/2)$ we obtain for $k = n/2$ $(m = 2r + 1)$, as shown in the first table at the bottom of the page. The sample number is 10^6 for r in $\{2, 3, 4\}$, 10^5 for r in $\{5, 6\}$, 10^4 for $r = 7$ and 10^3 for $r = 8$. For the first Reed–Muller codes with $k < n/2$ $(m = 2r + 2)$ we get with the same number of samples, as shown in the second table at the bottom of the page. These results seem to indicate, as pointed out in [2], that Reed–Muller codes behave as random codes. In other words, concerning their erasures recovering capacity, the Reed–Muller codes seem to be good codes. However, giving a tight theoretical bound is still an open problem.

B. Union Bound

For an erasure pattern of Hamming weight e we can get an error probability upper bound using the classical union bound on the erasure channel (cf. [8]):

$$P_{\text{err}} \leq \sum_{w \leq e} A_w \frac{\binom{n-w}{e-w}}{\binom{n}{e}}. \tag{12}$$

In this expression A_w is the number of codewords of Hamming weight w in the code.

This bound is good for random codes (see [12]), but we do not always know the value of the A_w's. For the Reed–Muller codes for example, we only know the weight distribution for the code of order 1 or 2, the codes $\mathrm{RM}(r, m)$ for $m \leq 8$, $\mathrm{RM}(3, 9)$, $\mathrm{RM}(3, 10)$ and their duals. So, when the A_w are unknown, we have no other choice than going for the worst and suppose that all the codewords are of minimal weight d. We obtain what we denote by the minimal distance union bound:

$$P_{\text{err}} \leq (2^k - 1) \frac{\binom{n-d}{e-d}}{\binom{n}{e}}. \tag{13}$$

Here are the first Reed–Muller codes for which this bound is smaller than 1 on a balanced erasure pattern as shown in the third table at the bottom of the page. It is possible for the Reed–Muller codes to get a slightly tighter bound because the weight distribution is known for weights smaller than $2.5d$ (cf. [10], [11]). Using the weight distribution up to $2d$, we obtain the fourth table at the bottom of the page. However, performing the same computation as in the next subsection, we can see that these bounds are equivalent from the asymptotic point of view.

C. Application to Reed–Muller Codes

We will perform the computation in the case of a Reed–Muller code with a balanced function $(e = n/2)$. The calculations are of course similar if we take $e = \alpha n$ for an α in $[0, 1]$. Let $r = \lambda m$ with λ in $]0, 1[$.

We are looking for a λ such that the error probability bound (13) tends to 0 when m goes to infinity. For this upper bound to

r, m	$2,5$	$3,7$	$4,9$	$5,11$	$6,13$	$7,15$	$8,17$
P_{err}	0.67	0.84	0.71	0.71	0.71	0.72	0.72

r, m	$2,6$	$3,8$	$4,10$	$5,12$	$6,14$
rate	0.34	0.36	0.37	0.39	0.40
P_{err}	3.10^{-3}	2.10^{-5}	0	0	0

r, m	$2,7$	$3,11$	$4,15$	$5,19$	$6,24$
$P_{\text{err}} \leq$	7.10^{-4}	10^{-15}	10^{-62}	10^{-31}	10^{-22606}

r, m	$2,6$	$3,9$	$4,13$	$5,18$	$6,22$
$P_{\text{err}} \leq$	3.10^{-3}	10^{-9}	10^{-11}	10^{-1372}	10^{-7245}

tend to 0, it is necessary that $\lambda < 1/2$. For such λ, we can write for the code rate

$$k \sim \binom{m}{\lambda m} \sim e^{m\mathrm{h}(\lambda)} \tag{14}$$

where h is the entropy function

$$h(\lambda) := \lambda \ln(\lambda) + (1-\lambda)\ln(1-\lambda). \tag{15}$$

Moreover, we have for the binomial part of (13)

$$\frac{\binom{n-d}{e-d}}{\binom{n}{e}} = \left(\frac{e}{n}\right)^{d+o(d)} = \left(\frac{1}{2}\right)^{d+o(d)} \tag{16}$$

and since

$$d = 2^{m-r} = e^{m(1-\lambda)\ln 2} \tag{17}$$

a necessary and sufficient condition for the minimal distance upper bound to vanish is given by the following inequation:

$$\mathrm{h}(\lambda) < (1-\lambda)\ln 2. \tag{18}$$

Solving this inequation for λ we get

$$\lambda < 0.227\ldots \tag{19}$$

and we retrieve the result of [14] on the algebraic immunity of a random balanced function. Note that this bound does not seem very tight because we get a rate (k/n) that goes exponentially fast to 0 whereas for random code we get a vanishing error probability for a constant rate. We will see in the following that we can obtain a much better result.

IV. New Bound

This section is devoted to the proof of our first main theorem presented in the introduction. Analyzing the behavior of a decoding algorithm in the first subsection, we will obtain an upper bound on the block error probability over the erasure channel. It is a simplified version of this bound, presented in Section IV-B, that corresponds to Theorem 1.

A. Basic Idea

Given a linear code $\mathcal{C}$ of dimension k and an erasure pattern of weight e, we recall that $\mathcal{I}$ is the subspace of the codewords included in the erasure pattern. A simple algorithm to compute $\mathcal{I}$ can be described as follows.

We pick up the $n-e$ nonerased positions in a random order. We compute incrementally $\mathcal{I}_j$, the subspace of codewords of $\mathcal{C}$ which are zero on the first j positions which have been picked up. We say that we are in state (i,j) if after considering j nonerased positions, $\mathcal{I}_j$ is of dimension i. By definition we have $\mathcal{I} = \mathcal{I}_{n-e}$ and an erasure pattern is decodable if and only if at the end of the algorithm we are in state $(0, n-e)$.

Let $p_{i,j}$ be the probability to be in state (i,j) over all erasure patterns of weight e and positions orders. Notice that the probability of unambiguous decoding is nothing but $p_{0,n-e}$. The idea behind our bound is to get information on the $p_{i,j}$'s by investigating the transition probabilities between the states (i,j).

For that, assume that we are in state (i,j) and that we pick up a new nonerased position. There are only two possibilities:

- $\dim(\mathcal{I}_{j+1}) = \dim(\mathcal{I}_j) - 1 = i - 1$;
- $\dim(\mathcal{I}_{j+1}) = \dim(\mathcal{I}_j) = i$.

Let t_{ij} be the probability of the first event. The other event will then happen with probability $1 - t_{ij}$.

Now, let us investigate these transition probabilities. Actually, given j positions and an associated space $\mathcal{I}_j$, the dimension of the next subspace depends only on whether or not the next position is in $\mathrm{supp}(\mathcal{I}_j)$. If it is in the support then the new constraint will exclude some codewords and the dimension will decrease. If it is outside the support then the subspace will be the same. Hence, using the generalized Hamming weights, it is easy to obtain a lower bound on t_{ij}

$$t_{ij} \geq t'_{ij} := \frac{d_i}{n-j}. \tag{20}$$

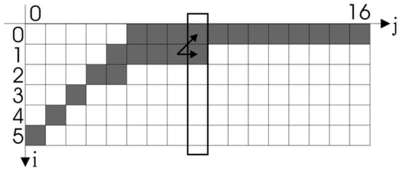
Here $n-j$ is the number of positions we can choose from and the i-th generalized Hamming weight d_i is a lower bound on an i-dimensional subcode support size. Remark that all the first j positions are outside the support by definition of $\mathcal{I}_j$, hence the bound on t_{ij}. Notice also that for the same reason, it is impossible to be in a state (i,j) where $d_i > n-j$.

The idea is now to compute new probabilities p'_{ij} just using the transition probabilities t'_{ij}. For that, we will fill an array $k \times n$ in a dynamic programming manner. We start with all the probabilities equal to 0 except $p'_{k0} = 1$, and we fill the cells of the array column by column using the relations:

$$p'_{i(j+1)} = p'_{ij}(1 - t'_{ij}) + p'_{(i+1)j} t'_{(i+1)j}. \tag{21}$$

Notice that the t'_{ij} associated with impossible states are irrelevant since these states are unreachable and their probability will remain to 0.

The process is illustrated in the next figure for the code RM$(1,4)$ of dimension 5 and generalized Hamming weights $0, 8, 12, 14, 15$, and 16.



Starting from the bottom-left corner, each cell gives his mass to its neighbors according to the black arrows in the figure. When we are in state of the form $(i, n-d_i)$ then the next picked up position will always fall in the support of $\mathcal{I}_j$. In this case, we are sure that the space dimension will decrease. So, all the states in row i with a column index strictly greater than $n-d_i$ are unreachable. The unreachable states correspond to white cells on the figure.

Now, the issue is: how are the p'_{ij}'s related to the p_{ij}'s? In fact, if we consider the partial sums column by column

$$S_{ij} := \sum_{h=0}^{i} p_{hj} \quad \text{and} \quad S'_{ij} := \sum_{h=0}^{i} p'_{hj} \tag{22}$$

we can show that

Lemma 1: for all j in $[\![0, n]\!]$ and for all i in $[\![0, k]\!]$ we have

$$S'_{ij} \leq S_{ij}. \tag{23}$$

Proof: We will prove the lemma by considering what happens to the state probabilities if we decrease the t_{ij} one by one. When we replace one t_{ij} by t'_{ij}, the probability $p_{(i-1)(j+1)}$ decreases whereas the probability $p_{i(j+1)}$ increases. However, all the partial sums of the column $j+1$ will be smaller than or equal to the previous ones. Using the same probabilities as before to compute the other columns, since we can write

$$S_{i(j+1)} = S_{ij} + t_{(i+1)j}\left[S_{(i+1)j} - S_{ij}\right] \tag{24}$$

we get

$$S_{i(j+1)} = (1 - t_{(i+1)j})S_{ij} + t_{(i+1)j}S_{(i+1)j} \tag{25}$$

and all the other partial sums are smaller than or equal to the previous ones. Finally, we finish the proof by changing one by one all the transition probabilities. □

In the end, we get an upper bound on the error probability for an erasure pattern of weight e since

$$P_{\text{err}} = 1 - p_{0(n-e)} = 1 - S_{0(n-e)} \leq 1 - S'_{0(n-e)}. \tag{26}$$

We can compute this bound by filling the array given that the generalized Hamming weights are computable. This is the case for Reed–Muller codes as explained in Appendix. Here are the first Reed–Muller codes for which this bound gives small values on balanced functions: see the table at the bottom of the page. We can see with this table that this bound is much better than the bound (13). However, more analysis is needed to apply it for large codes or to get an asymptotic behavior. This is the topic of Section IV-B.

B. Further Analysis

It appears difficult to derive a simple closed form expression of the p'_{ij}'s. However, using some linear algebra we will see here that we are able to derive enough information on them for our purpose.

We can extend the relation (21) to a linear relation between two columns of the dynamic programming array. Let C_j be column j, that is

$$C_j := \begin{pmatrix} p'_{0j} \\ \vdots \\ p'_{kj} \end{pmatrix}. \tag{27}$$

We can write

$$C_{j+1} = A_j C_j \tag{28}$$

where A_j is the following bidiagonal matrix:

$$A_j := \begin{pmatrix} 1 - t'_{0j} & t'_{1j} & & \\ & 1 - t'_{1j} & \ddots & \\ & & \ddots & t'_{kj} \\ & & & 1 - t'_{kj} \end{pmatrix}. \tag{29}$$

Let us look at a left eigenvector Σ_{ij} of A_j associated with the eigenvalue $1 - t'_{ij}$. Solving the system of equations that the coefficients of such eigenvector have to satisfy, we can take for Σ_{ij}

$$\Sigma_{ij} = \left(\underbrace{0 \ldots 0}_{i-1}, 1, \frac{t'_{(i+1)j}}{t'_{(i+1)j} - t'_{ij}}, \ldots, \prod_{h=i+1}^{k} \frac{t'_{hj}}{t'_{hj} - t'_{ij}}\right) \tag{30}$$

where the first 1 is in position i. Notice that by using the formula (20) for the t'_{ij}'s these eigenvalues do not depend on j! Hence, we will simply write in the future Σ_i where

$$\Sigma_i = \left(\underbrace{0 \ldots 0}_{i-1}, \frac{d_{(i+1)}}{d_{(i+1)} - d_i}, \ldots, \prod_{h=i+1}^{k} \frac{d_h}{d_h - d_i}\right). \tag{31}$$

The idea is now to track down the evolution of $\Sigma_i C_j$ from one column to another. We have

$$\Sigma_i C_j = \Sigma_i \left(A_{j-1} \cdots A_0 C_0\right) = \left(\prod_{h=0}^{j-1}(1 - t'_{ih})\right)\Sigma_i C_0 \tag{32}$$

that is, written with the generalized Hamming weights

$$\Sigma_i C_j = \left(\prod_{h=0}^{j-1} \frac{n - h - d_i}{n - h}\right)\Sigma_i C_0. \tag{33}$$

Moreover, since the only nonzero coefficient of C_0 is the last one and is equal to 1, using (31) we get

$$\Sigma_i C_0 = \prod_{h=i+1}^{k}\left(\frac{d_h}{d_h - d_i}\right). \tag{34}$$

This gives us for all i and j the value of a weighted sum of the last $n - i$ coefficients of the column C_j. The point is that all the weights in this summation are greater than 1, hence we have

$$1 - S_{ij} \leq \Sigma_i^t C_j. \tag{35}$$

r, m	$2,7$	$3,9$	$4,11$	$5,13$	$6,15$
P_{err}	3.10^{-7}	1.10^{-8}	4.10^{-10}	5.10^{-11}	3.10^{-6}

This yields an upper bound on the probability that the dimension of $\mathcal{I}_j$ is greater than or equal to a given value i. Applying it for $i = 1$ we get our first theorem:

$$P_{\text{err}} \leq \prod_{h=e+1}^{n} \left(\frac{h-d_1}{h}\right) \prod_{h=2}^{k} \left(\frac{d_h}{d_h - d_1}\right). \tag{36}$$

Considering the way we obtained this bound, one may think that this bound is very loose because the last coefficients of Σ_1 are very large. But actually, in the columns C_j we consider, almost all the probability mass is concentrated in the first coefficients and these are precisely the ones for which the summation weights are the closest to one.

To confirm that our simplified bound is close to $1 - p'_{0,n-e}$, let us compare the numerical results of the previous subsection (method A) and the ones obtained with formula (36) (method B), as shown in the table at the bottom of the page. The values appear to be almost the same up to degree 7 (the first degrees are not displayed). After that, computing the $p'_{i,j}$'s using dynamic programming becomes too long, but the new bound (36) can still be computed.

V. Error Bound for Rate-Consistent Codes

We apply here bound (36) to investigate the behavior of a class of linear codes that we call "rate-consistent." We says that a linear code of rate ν is rate-consistent if it does not admit any subcode of rate greater than ν. This class is fairly general and we will see below that self-dual codes, cyclic codes, Reed–Muller codes, generalized Reed–Muller codes and geometric Goppa codes are all rate-consistent.

A. Rate-Consistent Codes

The rate of a linear code is defined by its dimension divided by its length. Following this statement, we can define the rate of a subcode by its dimension over its support size. Now, since the ith Hamming weight is the minimal support size of a i-dimensional subcode, the fraction i/d_i corresponds to the maximum rate of an i-dimensional subcode.

We say that a linear code of rate ν is rate-consistent if it does not admit any subcode of rate greater than ν. That is, if the code satisfy

$$\forall\, i \in \llbracket 1, k \rrbracket, \quad \frac{i}{d_i} \leq \frac{k}{n} \tag{37}$$

where the $(d_i)_{i \in [1,k]}$ are its general Hamming weights.

This property is actually a lower bound on the generalized Hamming weights and as we will see with the following lemmas it is verified by a lot of codes.

Lemma 2: All linear self-dual codes and all linear cyclic codes are rate-consistent.

Proof: Let $\mathcal{C}$ be a linear code of dimension k and length n. Let $\mathcal{B}$ be an information set of $\mathcal{C}$. Let V be a linear subcode of dimension i. The key point here is that we have

$$|\mathcal{B} \cap \mathrm{supp}(V)| \geq i. \tag{38}$$

In order to prove the lemma for self-dual codes, remark that k positions form an information set of $\mathcal{C}$ iff there is no nonzero word in the dual of C with a support included in these positions. Therefore, the complementary of $\mathcal{B}$ is always an information set for the dual of $\mathcal{C}$. When the code is self-dual, we get two disjoint information sets and by (38) we have $|\mathrm{supp}(V) \geq 2i|$. Finally, since a self-dual code is always of rate $1/2$, we obtain Property (37).

In the case of a cyclic code, looking at the generator matrix spawned by the minimal polynomial (see [13]) one can show that any k consecutive positions (even those that wrap around the code block) form an information set. For each of these n information sets, we have Property (38). Moreover, each point in $\mathrm{supp}(V)$ can contribute to only k such information sets. Counting this contribution, we get

$$k\, |\mathrm{supp}(V)| \geq n\, i. \tag{39}$$

This concludes the proof. □

Lemma 3: The dual of a linear rate-consistent code is also rate-consistent.

Proof: We use the same notation as in the previous lemma and we denote $(d'_i)_{i \in \llbracket 1, n-k \rrbracket}$ the generalized Hamming weights of the dual of $\mathcal{C}$. The result comes from a relation between the Hamming weights of a code and those of its dual discovered by Wei [16]:

$$\{(d_i)_{i \in \llbracket 1,k \rrbracket}\} \cup \{(n+1-d'_i)_{i \in \llbracket 1,n-k \rrbracket}\} = \{1, \ldots, n\}. \tag{40}$$

We can rewrite the rate-consistency property (37) by

$$\forall a \in \llbracket 1, n \rrbracket \quad |\{d_i, d_i \in \llbracket 1, a \rrbracket\}| \leq a\frac{k}{n}. \tag{41}$$

This is clearly true when a is equal to one of the d_i's and so it is true for all a. Now, using the dual relation (40) we get

$$\forall a \in \llbracket 1, n \rrbracket \quad |\{d'_i, d'_i \in \llbracket n+1-a, n \rrbracket\}| \geq a - a\frac{k}{n} \tag{42}$$

and for a of the form $n - d'_i$, we retrieve the rate consistency property for the dual. □

Lemma 4: Reed–Muller codes are rate-consistent.

Proof: See Appendix. □

We also know or have sharp bound on the generalized Hamming weights for other code families. For instance, using the results in [15], we have checked the rate-consistency of generalized Reed–Muller codes and geometric Goppa codes. The proof

r, m	$6, 15$	$7, 18$	$8, 20$	$9, 22$
$P_{\text{err}}\ (A)$	3.10^{-6}	3.10^{-147}	$-$	$-$
$P_{\text{err}}\ (B)$	4.10^{-6}	3.10^{-147}	8.10^{-198}	5.10^{-210}

for each code is straightforward but require a precise description for the generalized Hamming weights. Describing these results is too long and beyond the scope of this paper.

B. Error Bound for Rate-Consistent Codes

We will now derive the behavior of bound (1) for rate-consistent codes. Taking the logarithm of this bound, we obtain

$$\ln(P_{\text{err}}) \le \sum_{h=e+1}^{n} \ln\left(1 - \frac{d_1}{h}\right) - \sum_{i=2}^{k} \ln\left(1 - \frac{d_1}{d_i}\right). \quad (43)$$

The first sum is upper-bounded by

$$\sum_{h=e+1}^{n} \ln\left(1 - \frac{d_1}{h}\right) \le - \sum_{h=e+1}^{n} \frac{d_1}{h} \le d_1 \ln\left(\frac{e}{n}\right). \quad (44)$$

For the second sum, notice that for all i the fraction d_1/d_i is smaller than d_1/d_2. Applying the Griesmer bound (see, e.g., [16]), a lower bound on the generalized Hamming weights for all linear codes of minimal distance d_1, we always have $d_1/d_2 \le 2/3$. So, there exists a positive constant $a < 1$ such that

$$-\sum_{i=2}^{k} \ln\left(1 - \frac{d_1}{d_i}\right) \le \sum_{i=2}^{k} \left(\frac{d_1}{d_i} + a\frac{d_1^2}{d_i^2}\right) \quad (45)$$

and the asymptotic behavior of the second logarithm sum depends on the sum of the inverse generalized Hamming weights. Lower bounding the first $i_0 = \left\lfloor \frac{d_1 k}{n} \right\rfloor$ generalized Hamming weights by d_1 and using the rate-consistency property (37) for the other, we obtain

$$\sum_{i=1}^{k} \frac{1}{d_i} \le \frac{i_0}{d_1} + \frac{k}{n} \sum_{i=i_0+1}^{k} \frac{1}{i}. \quad (46)$$

The right hand term can be upper bounded by

$$\frac{i_0}{d_1} + \left(\frac{d_1 k}{n} - i_0\right)\frac{1}{d_1} + \frac{k}{n}\int_{\frac{d_1 k}{n}}^{k} \frac{1}{x} \le \frac{k}{n} + \frac{k}{n}\int_{\frac{d_1 k}{n}}^{k} \frac{1}{x}. \quad (47)$$

In the same way, we have for the sum of squares

$$\sum_{i=1}^{k} \frac{1}{d_i^2} \le \frac{i_0}{d_1^2} + \frac{k^2}{n^2} \sum_{i=i_0+1}^{k} \frac{1}{i^2} \le \frac{k}{n}\left(\frac{1}{d_1} + \frac{k}{n}\int_{\frac{d_1 k}{n}}^{k} \frac{1}{x^2}\right). \quad (48)$$

Together with inequality (45) we get

$$-\sum_{i=2}^{k} \ln\left(1 - \frac{d_1}{d_i}\right) \le \frac{d_1 k}{n}\left[\ln\frac{n}{d_1} + 3\right]. \quad (49)$$

Finally, we obtain the following upper bound on the error probability

$$\ln(P_{\text{err}}) \le d_1\left[\frac{k}{n}\left(\ln\frac{n}{d_1} + 3\right) - \ln\frac{n}{e}\right] \quad (50)$$

which is our Theorem 2.

C. Application to Reed–Muller Codes

Since Reed–Muller codes are rate-consistent, we will just apply here the results of the previous subsection. One might wonder why we just use the rate-consistency lower bound where we could have used the exact generalized Hamming weights distribution. Actually, it appears that this lower bound is quite sharp for Reed–Muller codes of rate one half.

For Reed–Muller codes and balanced erasure pattern, we can assume that $k < 1/2$, so that n/d_1 is greater than $\sqrt{n}$. In this case, using formula (50), we get for all constant $a < 1$ a vanishing error probability when

$$\frac{k}{n} \le \frac{2a\ln 2}{\ln n}. \quad (51)$$

What is the corresponding r for such a k? Considering a binomial law X of parameter $p = 1/2$ on m trials, we have

$$k = nP(X \le r). \quad (52)$$

Using the Chernoff bound we get for a variable $\lambda > 0$

$$P\left(X - \frac{m}{2} \le -\lambda\frac{\sqrt{m}}{2}\right) \le \exp\left(-\frac{\lambda^2}{2}\right). \quad (53)$$

So, with a r of the form $m/2 - \lambda\sqrt{m}/2$, a sufficient condition on λ for k to satisfy (51) is given by

$$\lambda^2 \ge 2\ln(\ln n) - 2\ln(2a\ln 2). \quad (54)$$

In the end, we have a vanishing error probability for a r satisfying

$$r \le \frac{m}{2} - \sqrt{\frac{m}{2}\ln\left(\frac{m}{2a\ln 2}\right)} \quad (55)$$

for all a in $]0,1[$. With this result, we also get an asymptotic lower bound on the algebraic immunity of a balanced random Boolean function. This is our Theorem 3.

VI. Conclusion

We have exploited here the information contained in the generalized Hamming weights to derive an error probability upper bound over the erasure channel.

Moreover, we have seen that for a fairly general class of linear codes, we can derive an interesting error bound just knowing the dimension and the minimum distance of the code. This applies to the class of rate-consistent codes for which we have a lower bound on the generalized Hamming weights. Notice that we have proved the rate consistency for several code families and it is safe to conjecture that it is satisfied by many more families.

Finally, remark that some codes may have generalized Hamming weights far away from the lower bound given by the rate-consistency property. For those codes, one can derive a better block error probability bound after decoding over the erasure channel. This will probably not be the case for Reed–Muller codes but may apply to generalized Reed–Muller codes or geometric Goppa codes.

Appendix

We detail here the Reed–Muller generalized Hamming weights and prove the rate-consistency of this code family.

The generalized Hamming weights of $\mathrm{RM}(r,m)$ were first calculated in [16] and can be expressed quite easily. Let us consider the numbers from 0 to 2^m-1 written in base 2. For such a number x, we will denote the number of 1 in its binary decomposition by $|\overline{x}^2|$. Arrange the numbers of the set

$$X(r,m) := \left\{ x \in [\![0, 2^m-1]\!], \quad |\overline{x}^2| \leq r \right\} \tag{56}$$

in increasing order, and let x_i be the ith number of this ordered set. There are k such numbers because a trivial bijection with the monomials of $\mathrm{RM}(r,m)$ exists:

$$\left(\overline{x}^2 = \alpha_m \ldots \alpha_1\right) \Leftrightarrow \left(X_m^{\alpha_m} \ldots X_1^{\alpha_1}\right). \tag{57}$$

The generalized Hamming weights of $\mathrm{RM}(r,m)$ are then given by (see [16])

$$d_i = n - x_{k-i}. \tag{58}$$

One can verify that we have $d_1 = 2^{m-r}$ and $d_k = n$.

Now, let us prove the rate-consistency of Reed–Muller codes. We have to show that the generalized Hamming weights d_i of the Reed–Muller code of length n and dimension k satisfy

$$\forall\, i \in [\![1,k]\!], \quad d_i \geq i\,\frac{n}{k}. \tag{59}$$

We will actually prove the following equivalent formula on the x_i:

$$\forall\, i \in [\![1,k]\!], \quad x_i \leq i\,\frac{n}{k}. \tag{60}$$

The proof is done by induction on m. Property (60) is true for $m = r$. We assume it to be true for $m-1$ and want to prove it for m. Using the bijection (57) and writing $k(r,m)$ for the dimension of $\mathrm{RM}(r,m)$ it can be checked that the relation

$$k(r,m) = k(r,m-1) + k(r-1,m-1) \tag{61}$$

corresponds to the distribution of the x_i's before and after $n/2$. In the following, we will respectively write k^+ for $k(r,m-1)$ and k^- for $k(r-1,m-1)$. This notation reflects the fact that k^+ is always greater than or equal to k^-.

Using the induction hypothesis when i is smaller than k^+, we get the sought property almost directly

$$x_i \leq i\frac{2^{m-1}}{k^+} \leq i\frac{n}{2k^+} \leq i\frac{n}{k}. \tag{62}$$

For i greater than k^+, a little more work is needed. By induction, $(x_i - n/2)$ being the $(i-k^+)$th number of the set $X(r-1,m-1)$, we have

$$x_i \leq \frac{n}{2} + (i-k^+)\frac{n}{2k^-}. \tag{63}$$

Separating the term we want to obtain, we get

$$x_i \leq i\frac{n}{k} + n\left[\frac{1}{2} - \frac{k^+}{2k^-} + i\left(\frac{1}{2k^-} - \frac{1}{k}\right)\right]. \tag{64}$$

Remark that the factor multiplying the rightmost i is positive. So, by upper bounding this i by k, we can write

$$x_i \leq i\frac{n}{k} + n\left[\frac{1}{2} - \frac{k^+}{2k^-} + \frac{k}{2k^-} - 1\right]. \tag{65}$$

Finally, since $k = k^+ + k^-$, the term in bracket is equal to zero and we obtain once again the property (60).

Acknowledgment

The author is indebted to Jean-Pierre Tillich for his helpful comments and discussions.

References

[1] A. Braeken and B. Preneel, “On the algebraic immunity of symmetric boolean functions,” in *Proc. INDOCRYPT*, Jul. 26, 2005, vol. 3797, pp. 35–48 [Online]. Available: http://eprint.iacr.org/,/245,26,2005/245

[2] C. Carlet and P. Gaborit, “On the construction of balanced Boolean functions with a good algebraic immunity,” in *Proc. BFCA (First Workshop on Boolean Functions: Cryptogr. Appl.)*, Rouen, France, Mar. 2005, pp. 1–14.

[3] N. Courtois and W. Meier, “Algebraic attacks on stream ciphers with linear feedback,” *Proc. Adv. Crypt.—EUROCRYPT 2003*, vol. 2656, pp. 346–359, 2003.

[4] D. K. Dalai, K. C. Gupta, and S. Maitra, “Results on algebraic immunity for cryptographically significant Boolean functions,” in *Proc. INDOCRYPT*, ser. Lecture tes in Computer Science, A. Canteaut and K. Viswanathan, Eds. New York: Springer, 2004, vol. 3348, pp. 92–106.

[5] ——, “Cryptographically significant boolean functions: construction and analysis in term of algebraic immunity,” in *Fast Software Encryption*, ser. Lecture tes in Computer Science. New York: Springer, 2005, vol. 3557, pp. 98–101.

[6] D. K. Dalai, S. Maitra, and S. Sarkar, “Basic theory in construction of Boolean functions with maximum possible annihilator immunity,” in *Designs, Codes , Cryptogr.*, Jul. 15, 2005, vol. 40, pp. 41–58 [Online]. Available: http://eprint.iacr.org/,/229,15,2005/229

[7] F. Didier and J.-P. Tillich, “Computing the algebraic immunity efficiently,” *Fast Software Encryption, FSE*, 2006.

[8] P. Elias, “Coding for noisy channels,” *IRE Conv. Rec.*, pp. 37–46, Mar. 1955.

[9] T. Helleseth, T. Kløve, and V. I. Levenshtein, “On the information function of an error-correcting code,” *IEEE Trans. Inf. Theory*, vol. 43, pp. 549–557, 1997.

[10] T. Kasami and N. Tokura, “On the weight structure of Reed-Muller codes,” *IEEE Trans. Inf. Theory*, vol. 16, pp. 752–759, 1970.

[11] T. Kasami, N. Tokura, and S. Asumi, “On the weight enumeration of weights less than 2.5d of Reed-Muller codes,” *Inf. Contr.*, vol. 30, no. 4, pp. 380–395, 1974.

[12] S. J. MacMullan and O. M. Collins, “A comparison of known codes, random codes, and the best codes,” *IEEE Trans. Inf. Theory*, vol. 44, pp. 3009–3022, 1998.

[13] F. MacWilliams and N. Sloane*, The Theory of Error-Correcting Codes* . Amsterdam, The Netherlands: North-Holland, 1977.

[14] W. Meier, E. Pasalic, and C. Carlet, “Algebraic attacks and decomposition of Boolean functions,” *Lecture Notes in Comput. Sci.*, vol. 3027, pp. 474–491, Apr. 2004.

[15] M. A. Tsfasman and S. G. Vlăduţ, “Geometric approach to higher weights,” *IEEE Trans. Inf. Theory*, vol. 41, pp. 1564–1588, 1995.

[16] V. K. Wei, “Generalized Hamming weights for linear codes,” *IEEE Trans. Inf. Theory*, vol. 37, pp. 1412–1418, Sep. 1991.