

Just how resilient are they?

William J. Martin
Department of Mathematical Sciences
Department of Computer Science

Berk Sunar
Department of Electrical and Computer Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts
{martin,sunar}@wpi.edu

November 2, 2009

Abstract

Resilient functions can be viewed as string condensation methods which “remove” an opponent’s partial knowledge of the string. These are closely related to coding theory and the theory of orthogonal arrays. The formal definition of a resilient function assumes a strict upper bound on an adversary’s knowledge of the input. Our investigation here is motivated by real-world applications in which the inputs to our resilient function cannot be guaranteed to be so well-behaved. Using ideas from coding theory, we give a detailed performance analysis for both the general case and for resilient functions arising from specific families of binary linear codes. As it turns out, resilient functions constructed from linear codes perform *almost* perfectly halfway beyond their resiliency degree. Furthermore, we conduct our study in the concrete setting, i.e. we study the exact (non-asymptotic) performance for a given parameter size. Hence, our results are readily accessible to the practitioner who needs to pick specific parameter sizes in any given cryptographic application.

1 Introduction

We study the performance of resilient functions beyond their resiliency degree. Among other applications, we are motivated by the introduction of numerous physical attacks that target the implementation of cryptographic schemes. Resilient functions provide a useful tool in the hands of cryptographers, who employ these functions to handle the risk that secrets are (or may be) partially exposed to an adversary. Such exposure or leakage may result from a variety of effects: hardware/software failures, improper disposal of old equipment,

insufficient isolation of memory space from potentially malicious processes (e.g. viruses and worms), failures and bugs in security protocols, etc. Given that we are computing and communicating with imperfect protocols running on imperfect equipment that often leaks information, it becomes essential to

- soften the restrictions we place on an attacker’s capabilities, and
- analyze the performance of cryptographic schemes when the security assumptions no longer hold.

In this paper, we focus on the role played by resilient functions in this effort.

First introduced by Chor, et al. [9] and, independently, by Bennett, Brassard and Robert [3] resilient functions (along with secret sharing schemes, introduced by Shamir [26]) were among the first primitives to be used in the construction of cryptographic schemes that survive in the presence of leaked key bits. For example, in these original references, resilient functions were proposed to enable fault-tolerant distributed computing and privacy amplification. In order to motivate the detailed study of resilient functions that follows, we first survey a few recent developments in cryptography involving information leakage.

One striking instance of such an attack was presented by van Someren [25] and developed more thoroughly by Shamir and van Someren [23]. The technique is quite simple, yet effective; it works by scanning for high entropy strings in computer memory. As it turns out, cryptographic keys have high likelihood of being uncovered in such a search. Within weeks of the publication of this attack, computer viruses exploiting these ideas emerged in public (cf. [6]). Such attacks then fueled the development of numerous practical and theoretical techniques for countermeasures.

To tackle the leakage problem in a formal cryptographic setting, Dodis, et al. [6] introduced exposure resilient functions which generalize classical resilient functions as defined by Chor, et al. by allowing for an imperfect output distribution. Dodis, et al. observe that, as long as the output distribution is exponentially close to uniform, the construction may still be used in many cryptographic applications. Their construction achieves its goal by introducing an extractor function that guaranties a near-uniform output distribution as long as the input distribution has sufficient min-entropy. The randomness required by the extractor is also derived from the input (specifically, from input bits not exposed to the adversary). In [15] Ishai, et al. go one step further and devise a secret-sharing-based technique to protect against information leakage during computation. They note that exposure resilient functions provide protection for storage but not computation.

To provide a more comprehensive solution, the *physically observable cryptography* framework was introduced by Micali et al. with the hope of formally capturing information leakage through probing attacks on storage devices. Similarly, the Algorithmic Tamperproof Model was developed by Gennaro et al. [8] to determine if existing provably secure schemes can be strengthened to survive against physical attacks while making minimal assumptions on read-proof and tamper-proof memory.

In the meantime, physical attacks are being improved at an alarming pace. Skorobogatov [24] showed that key bits can be recovered from memory even if the memory was erased,

provided an adversary has direct physical access to the memory device. Despite the strength of this attack, it requires advanced equipment and technical skills. In contrast, the more recent so-called *cold-boot attacks* introduced by Halderman et al. [14] require no equipment and only common programming skills. Cold-boot attacks allow an adversary to defeat even the strongest disk encryption products (e.g. Microsoft’s BitLocker) by simply reading the encryption keys from the physical memory quickly (within a few minutes) after the power is turned off.

Cold-boot attacks have motivated the introduction of a number of theoretical constructions that provide protection when an adversary learns a fraction of a stored secret [1, 21]. In [1], Akavia, et al. introduced a more realistic model that considers security against a wide class of side-channel attacks when some function of the secret key bits is leaked. In the same work, it is shown that Regev’s lattice-based scheme [22] is resilient to key leakage. More recently, Naor, et al. [21] proposed a generic construction for a public-key encryption scheme that is resilient to key leakage.

In this paper we are motivated by the fact that resilient functions will have to be used in many such real-world applications with non-ideal settings. In our applications, we expect no guarantees — only probability estimates — on the behavior of an adversary or an imperfect environment. We show that one can still make remarkably accurate statements about the expected behavior of the resilient function¹ when the number of leaked bits exceeds the resiliency degree. Our contribution is complementary to the work in [6] on exposure resilient functions: whereas they relax the definition of a resilient function to allow for an imperfect output distribution, thereby achieving a more flexible primitive through the use of an extractor, we instead study the performance of resilient functions as originally defined when the function is subject to conditions worse than expected. Our estimates are best when one has sufficient structural information about the binary linear code employed to define the function. More specifically, we effectively bound the entropy of the output for any number of leaked input bits. Our treatment is fundamentally different from the one in [6] since we study concrete security. Thus our approach allows one to precisely determine parameter sizes for resilient functions to be used in any given application.

2 Preliminaries

Throughout, let $\mathbb{Z}_2 = \{0, 1\}$ with modular arithmetic and consider functions

$$F : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m.$$

We say F has *input size* n and *output size* m . If X is a discrete random variable taking on values in $\mathcal{X} = \mathbb{Z}_2^n$ according to some probability distribution \mathcal{D} with probability mass function p (where we write $p_x = \mathbf{Prob}[X = x]$), then X has (*Shannon*) *entropy*

$$H(X) = \sum_{x \in \mathcal{X}} -p_x \log_2 p_x.$$

¹Our treatment applies to *linear* resilient functions only. But all efficiently computable resilient functions known to us are essentially linear functions.

For any function F as above, such a probability distribution \mathcal{D} on \mathcal{X} induces a probability distribution \mathcal{E} on the codomain $\mathcal{Y} = \mathbb{Z}_2^m$ with probability mass function q given by $q_y = \text{Prob}[F(X) = y]$ where X is chosen according to distribution \mathcal{D} . So we obtain a random variable $Y = F(X)$ taking values in \mathcal{Y} and the entropy of Y is defined in a similar manner to that of X .

Definition 2.1 (Resilient Function). An (n, m, t) -resilient function is a function

$$(y_1, y_2, \dots, y_m) = F(x_1, x_2, \dots, x_n)$$

from \mathbb{Z}_2^n to \mathbb{Z}_2^m enjoying the property that, for any t coordinates i_1, \dots, i_t , for any constants z_1, \dots, z_t from \mathbb{Z}_2 , and for any element y of the codomain

$$\text{Prob}[F(x) = y | x_{i_1} = z_1, \dots, x_{i_t} = z_t] = \frac{1}{2^m}.$$

In the computation of this probability all x_i are viewed as independent random variables each of which takes on the value 0 or 1 with probability 0.5. We refer to the integer t as the resiliency degree of F .

In more informal terms, if up to t of the input bits are deterministic and the remaining bits are uniformly random and independent, the output of the resilient function will be perfectly random (or unpredictable). From a cryptographic viewpoint, knowledge of any t values of the input to the function does not allow one to make any better than a random guess at the output, even if one knows the function F in advance.

A simple technique for constructing resilient functions uses binary linear error-correcting codes. By an $[n, m, d]$ -code, we mean an m -dimensional subspace C of \mathbb{Z}_2^n in which any two distinct codewords (i.e., vectors in C) differ in at least d coordinates. Clearly $C = \text{rowsp } G$ for some $m \times n$ matrix G over the binary field; if C is equal to the row space of such a matrix, we say G is a *generator matrix* for C . The *weight enumerator* of C is the generating function

$$W_C(x) = \sum_{i=0}^n A_i x^i$$

where A_i is the number of codewords of Hamming weight i . For example, $A_0 = 1$, $A_1, \dots, A_{d-1} = 0$.

Theorem 2.2. (e.g., [9]) Let G be a generator matrix for a binary linear $[n, m, d]$ -code. Define a function $F : \mathbb{Z}_2^n \mapsto \mathbb{Z}_2^m$ by the rule $F(x) = Gx$. Then F is an $(n, m, d - 1)$ -resilient function.

The proof hinges on the simple fact that, since every non-zero codeword has Hamming weight at least d , the submatrix of G obtained by deleting any collection of up to $d - 1$ columns still has full row rank, so the corresponding linear transformation is still surjective.

For more information on resilient functions, and their connections to codes and designs see [7] and [27]. In this paper, all codes will be binary.

In [28], Stinson and the authors applied resilient functions to random number generators, an interesting situation where the choice of deterministic bits is not adversarial, but the probability that more than $d - 1$ bits are deterministic is non-negligible ($d - 1$ being the resiliency degree of a resilient function constructed from an $[n, m, d]$ -code). Immediately, we began to ask questions about the behavior of the function when the input conditions degrade beyond the resiliency degree. We wondered if all is lost or, as one intuitively expects, if the performance degrades smoothly as the number of deterministic bits exceeds the resiliency degree.

Specific questions we consider in this paper are the following:

- In an (n, m, t) -resilient function, what is the probability that the output entropy is still m if $k > t$ input bits are deterministic?
- When the number of deterministic input bits to an (n, m, t) -resilient function exceeds t , what is the expected value of the output entropy?
- How does the model handle independent but biased bits? I.e., suppose the n input bits are independent random variables each with its own bias towards one or zero; what can one say about the output entropy?
- How do familiar families of binary linear codes behave when the number of deterministic bits is equal to or larger than the minimum distance of the code?

We also hoped to gain some knowledge of the behavior of a resilient function when certain dependencies exist among various subsets of input bits. Our analysis is quite limited in this case. Since specific dependencies can lead to significant failure of the output, our results on this topic are quite crude. It may be that one may establish encouraging lower bounds on the output entropy if one stipulates only very restricted sorts of dependencies, but since we saw no practical use of such artificial assumptions, we did not pursue this further.

3 Preserving full entropy

Let C be an $[n, m, d]$ -code with generator matrix G and let $F : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$ be the corresponding $(n, m, d - 1)$ -resilient function. We have already pointed out that the deletion of any $d - 1$ or fewer columns of G results in a matrix of rank m . Clearly there are some sets of d or more columns whose deletion results in a matrix of rank less than m (i.e., if we delete a set of coordinates containing the support of any non-zero codeword). Let us call such a set of coordinates *degenerate* and let $N(t)$ denote the number of t -element sets of coordinates which are degenerate.

More generally, if S is any linear subspace of the binary space \mathbb{Z}_2^n , then $F(S)$ is also a subspace. We say S is *degenerate* if $F(S)$ has dimension less than m . We are most interested in the special case when $S = S_T$ consists of all binary n -tuples x satisfying $x_i = 0$ for $i \in T$ where T is a specified set of coordinates. Now the two notions of degeneracy coincide.

We will consider probability distributions on \mathbb{Z}_2^n which are uniform on some subspace S_T as described above and zero outside S_T . For any such distribution \mathcal{D} with associated random variable X , we obtain a transformed distribution \mathcal{E} on \mathbb{Z}_2^m with associated random variable $Y = F(X)$; clearly \mathcal{E} is uniform on $F(S_T)$. It is also obvious that the distribution \mathcal{D} has Shannon entropy $H(X) = n - |T|$ and the output distribution \mathcal{E} has Shannon entropy less than m if and only if T is degenerate. The *output entropy*, or entropy of distribution \mathcal{E} is our primary interest in this paper.

It will be useful to immediately generalize these notions to affine subspaces $S_T + z$ where z is not the zero tuple; that is, for any set T of t coordinates and any fixed values $\{z_i : i \in T\}$, the same reasoning about entropy holds when our input distribution has mass $p_x = 2^{t-n}$ on $\{x \in \mathbb{Z}_2^n : \forall i \in T (x_i = z_i)\}$ and has $p_x = 0$ for all other x . (Let us temporarily denote this distribution by $\mathcal{D}_{T,Z}$ where $Z = (z_i : i \in T)$.)

Definition 3.1. Let $F : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$ be given via generator matrix G and let degeneracy of subsets of $[n]$ be defined as above with respect to this matrix G . For a given integer t ($0 \leq t \leq n$), let a t -element subset $T \subseteq [n]$ be chosen uniformly at random (i.e., with probability $1/\binom{n}{t}$). Define $I(t)$ to be the event that T is a degenerate set of coordinates.

Observe that, in light of the previous discussion, we have

$$\text{Prob}[I(t)] = \text{Prob}[H(\mathcal{E}) < m \mid \mathcal{D} = \mathcal{D}_{T,Z} \text{ for some } T, Z, |T| = t].$$

Theorem 3.2. Let G be a the generator matrix for a binary $[n, m, d]$ -code. A set T of coordinates is degenerate with respect to G if and only if it contains the support of some non-zero codeword in $C = \text{rowsp } G$. For $t < \frac{3}{2}d$,

$$\text{Prob}[I(t)] = \frac{1}{\binom{n}{t}} \sum_{i=d}^t A_i \binom{n-i}{t-i}.$$

where $\sum_i A_i x^i$ is the weight enumerator of C . Finally, if T is any t -element subset with $t < \frac{3}{2}d$ and Z is any set of t binary values, then the probability distribution $\mathcal{D} = \mathcal{D}_{T,Z}$ satisfies $H(\mathcal{E}) \geq m - 1$ where $\mathcal{E} = F(\mathcal{D})$ is the output distribution of the resilient function $F(x) = Gx$ applied to distribution \mathcal{D} .

Proof. If G' is obtained from matrix G by deleting t columns, then $y^\top G' = 0$ forces $y^\top G = 0$ unless those t columns contain the support of some non-zero codeword. For $t < 3d/2$, any set T of t coordinate positions can contain the support of at most one non-zero codeword by the triangle inequality. So each codeword of Hamming weight i is contained in $\binom{n-i}{t-i}$ degenerate sets of coordinates. Summing over i gives the desired probability expression. For the last part, simply observe that the submatrix G' of G obtained by deleting less than $\frac{3}{2}d$ columns always has rank at least $m - 1$. \square

Later, we will generalize this result using the higher spectra of code C ; but the above expression for $\text{Prob}[I(t)]$ is easy to compute for t up to $1.5d$. In the following section, we explicitly compute this probability for some well-known codes. We go further by providing bounds on the failure probability for resilient functions constructed from several major classes of codes.

4 Analysis: specific classes of codes

In this section we refine our performance analysis by focusing on specific families of resilient functions.

4.1 Codes with Near Binomial Weight Distribution

Assume a k -dimensional binary linear code whose weight distribution is well-approximated by the binomial distribution. Note that this approximation works well for several important families of codes [18, page 283]. For example, Kasami et al. [16] prove that the weights of a binary primitive BCH code have approximate binomial distribution. So, for the following discussion, assume

$$A_i \leq \kappa \binom{n}{i} 2^{k-n} \quad , \quad \text{for } \forall i \geq d .$$

The failure probability for $t = 1.5d$ deterministic input bits satisfies

$$\begin{aligned} \text{Prob}[I(t)] &= \sum_{i=d}^t A_i \frac{\binom{n-i}{t-i}}{\binom{n}{t}} \\ &\leq \kappa 2^{k-n} \sum_{i=d}^t \binom{n}{i} \frac{\binom{n-i}{t-i}}{\binom{n}{t}} \\ &\leq \kappa 2^{k-n} \sum_{i=d}^t \binom{t}{i} . \end{aligned}$$

Note that $\sum_{i=d}^t \binom{t}{i} = \sum_{i=0}^{t-d} \binom{t}{i}$. Furthermore, assuming $t - d < t/2$ it holds [17, Thm. 1.4.5] that

$$\sum_{i=0}^{t-d} \binom{t}{i} \leq 2^{tH_2(\frac{t-d}{t})} \quad (4.1)$$

where $H_2(\cdot)$ denotes the binary (Shannon) entropy function. Using this bound together with the asymptotic Hamming bound [17, Thm. 5.2.8], $k/n + H_2(\frac{d}{2n}) \leq 1$, we obtain the following upper bound on the failure probability

$$\text{Prob}[I(t)] \leq \kappa 2^{-nH_2(\frac{d}{2n}) + tH_2(\frac{t-d}{t})} . \quad (4.2)$$

Setting $\delta = d/n$ and substituting $t = 1.5d$, the probability becomes bounded as follows:

$$\text{Prob}[I(1.5d)] \leq \kappa 2^{-n(H_2(\frac{\delta}{2}) - 1.377\delta)} .$$

where 1.377 is short for $\frac{3}{2}H_2(\frac{1}{3})$. Note that, in the exponent, we have $H_2(\frac{\delta}{2}) - \frac{3}{2}H_2(\frac{1}{3})\delta > 0$ for $\delta < 2/3$; any binary code of dimension at least two has this property. Hence, the probability of failure is decreasing exponentially with n for families of codes that have a weight distribution which is approximately binomial.

We summarize the result in the following theorem.

Theorem 4.1. Let C be an $[n, k, d]$ -code with weight distribution $A_i \leq \kappa 2^{k-n} \binom{n}{i}$ for $i > 0$. Then for $t < 2d$,

$$\text{Prob}[I(t)] \leq \kappa 2^{-nH_2(\frac{d}{2n}) + tH_2(\frac{t-d}{t})} .$$

and, in particular, with $\delta = d/n$,

$$\text{Prob}[I(1.5d)] \leq \kappa 2^{-n(H_2(\frac{\delta}{2}) - \frac{3}{2}H_2(\frac{1}{3})\delta)} .$$

Goppa Codes: The reference [13] provides empirical evidence that the weight enumerator of Goppa codes is very close to that expected of random linear codes, i.e.

$$A_i \approx \binom{n}{i} 2^{k-n} , \text{ for } \forall i \geq d .$$

The same reference provides evidence that the error in this approximation decreases exponentially with increasing code length. The existence of *good* Goppa codes that meet the Gilbert-Varshamov bound is well known [17]. Furthermore with the approximation technique given above in Equation (4.1) we can bound the failure probability of good Goppa codes as follows

$$\text{Prob}[I(t)] \leq 2^{-nH_2(\frac{d}{n}) + tH_2(\frac{t-d}{t})} .$$

For $t = 1.5d$ we obtain

$$\text{Prob}[I(1.5d), n \rightarrow \infty] \leq 2^{-n(H_2(\delta) - 1.377\delta)} .$$

Hence, we want $H_2(\delta) > 1.377\delta$ to obtain asymptotic exponential decrease in the failure probability. Solving the inequality we obtain the condition $\delta < \frac{2}{3}$. However, a binary linear code of dimension larger than one cannot have $d > \frac{2n}{3}$. Hence the failure probability is exponentially decreasing with increasing n for all well behaving Goppa codes². Also note that, the maximum (negative) constant of the exponent is obtained for $\delta = 0.278$ for which we obtain an information rate of $R = 0.147$.

The significance of this analysis is that it provides us evidence that resilient functions constructed with asymptotically well-behaving codes give close to perfect performance up to halfway beyond their resiliency degree.

5 Codes with concentrated weight distribution

In this section, we focus on the opposite end of the spectrum and survey the resilience performance of codes that have a rather concentrated weight distribution. Note that, despite the result of [16] alluded to above, such an approximation is far from accurate for certain families of codes, i.e. for codes in which the majority of the codewords have weight close to the minimum distance d .

²Due to the Gilbert Varshamov bound, $H_2(\delta) = 1 - R$, this is equivalent to requiring $R \geq 1 - H_2(2/3) = 0.081$.

Reed-Muller Codes: The weight distribution of the first order Reed Muller codes is well known, i.e. for $RM(1, u) = [2^u, u + 1, 2^{u-1}]$ we have $A_0 = A_{2^u} = 1$, $A_{2^{u-1}} = 2^{u+1} - 2$, and $A_i = 0$ otherwise. This simplifies the derivation substantially: for any $t < 2^u$,

$$I(t) = (2^{u+1} - 2) \binom{2^{u-1}}{t - 2^{u-1}} \binom{2^u}{t}^{-1}.$$

Example 5.1. Consider the resilient function constructed from the binary first order Reed-Muller code $RM(1, 4) = [16, 5, 8]$. We tabulate the performance of the resilient w.r.t. t , the number of deterministic input bits as follows.

t	8	9	10	11
$N(t)$	30	240	840	1680
$\text{Prob}[I(t)]$	0.00233	0.0209	0.104	0.384

(This code is considered further in Example 6.6.)

Going further, we bound the single bit loss probability of first order Reed-Muller codes at $t = 1.5d$ with the following theorem:

Theorem 5.2. The probability of a deterministic bit being produced at the output of a resilient function constructed from a first order Reed-Muller code for an input block with $t = 1.5d = 1.5 \cdot 2^{u-1}$ behaves asymptotically as follows

$$\text{Prob}[I(1.5d)] \sim 2^{-0.311277n + \log_2 n + 1.29}$$

Proof. The bound follows directly from the simplification of the bit-loss probability

$$\text{Prob}[I(t)] = \frac{(2^{u+1} - 2) \binom{2^{u-1}}{t - 2^{u-1}}}{\binom{2^u}{t}}$$

calculated at $t = 1.5d = 1.5 \cdot 2^{u-1}$. The expression is simplified using Stirling's factorial approximation [17, Thm. 1.4.2], i.e. $n! \sim n^n e^{-n} \sqrt{2\pi n}$ for large n .

$$\begin{aligned} \text{Prob}[I(1.5d)] &= (2^{u+1} - 2) \frac{\binom{2^{u-1}}{2^{u-2}}}{\binom{2^u}{1.5 \cdot 2^{u-1}}} \\ &= (2^{u+1} - 2) \frac{2^{u-1}! (3 \cdot 2^{u-2})!}{2^u! 2^{u-2}!} \\ &\sim (2^{u+1}) \frac{(2^{u-1})^{2^{u-1}} e^{-(2^{u-1})} \sqrt{2\pi(2^{u-1})} (3 \cdot 2^{u-2})^{3 \cdot 2^{u-2}} e^{-(3 \cdot 2^{u-2})} \sqrt{2\pi(3 \cdot 2^{u-2})}}{(2^u)^{2^u} e^{-(2^u)} \sqrt{2\pi(2^u)} (2^{u-2})^{2^{u-2}} e^{-(2^{u-2})} \sqrt{2\pi(2^{u-2})}} \\ &\sim \sqrt{6} \cdot 2^{u+(u-1)2^{u-1}+3(u-2)2^{u-2}-(u-2)2^{u-2}-u2^u} 3^{3 \cdot 2^{u-2}} \\ &\sim \sqrt{6} \cdot 2^{(3 \log_2 3 - 6)2^{u-2} + u} \\ &\sim 2^{-0.311277n + \log_2 n + 1.29}. \end{aligned}$$

Since Stirling's approximation becomes asymptotically precise, we have $\text{Prob}[I(1.5d)] = \Theta(2^{-0.311277n})$.

Simplex Codes: The weight distribution of the simplex code $[2^u - 1, u, 2^{u-1}]$ is simply given as $A_{2^u-1} = 2^u - 1$ and $A_i = 0$ for remaining values of $i > 0$. Hence,

$$\text{Prob}[I(1.5d)] = \frac{\sum_{i=d}^{1.5d} A_i \binom{n-i}{1.5d-i}}{\binom{n}{1.5d}}.$$

For $t = 1.5d = 2^{u-1} + 2^{u-2}$, we have

$$\begin{aligned} \text{Prob}[I(1.5d)] &= \frac{\sum_{i=d}^{1.5d} A_i \binom{n-i}{1.5d-i}}{\binom{n}{1.5d}} \\ &= (2^u - 1) \frac{\binom{2^u-1-2^{u-1}}{2^{u-2}}}{\binom{2^u-1}{2^{u-1}+2^{u-2}}} \\ &= (2^u - 1) \frac{(2^{u-1} - 1)!(2^{u-1} + 2^{u-2})!(2^{u-1} - 2^{u-2} - 1)!}{(2^{u-1} - 1 - 2^{u-2})!(2^{u-2})!(2^u - 1)!}. \end{aligned}$$

The expression is again simplified using Stirling's factorial approximation and by taking the first two terms in the Taylor Series approximation as follows

$$\begin{aligned} \text{Prob}[I(1.5d)] &\approx (2^u - 1) \frac{(2^{u-1} - 1)^{2^{u-1}-1} e^{-(2^{u-1}-1)} \sqrt{2\pi(2^{u-1} - 1)}}{(2^{u-2})^{2^{u-2}} e^{-(2^{u-2})} \sqrt{2\pi(2^{u-2})}} \\ &\quad \frac{(3 \cdot 2^{u-2})^{3 \cdot 2^{u-2}} e^{-(3 \cdot 2^{u-2})} \sqrt{2\pi(3 \cdot 2^{u-2})}}{(2^u - 1)^{2^{u-1}} e^{-(2^u-1)} \sqrt{2\pi(2^u - 1)}} \\ &\approx \frac{\sqrt{3}(2^u - 1)}{\sqrt{2^{u-1} + 1}} \frac{(2^{(u-1)(2^{u-1}-1)} - (2^{u-1} - 1)2^{(u-1)(2^{u-1}-2)})(3 \cdot 2^{u-2})^{3 \cdot 2^{u-2}}}{2^{(u-2)2^{u-2}}(2^{u(2^u-1)} - (2^u - 1)2^{u(2^u-2)})} \\ &\approx \frac{\sqrt{3}(2^u - 1)}{\sqrt{2^{u-1} + 1}} \frac{2^{(u-1)(2^{u-1}-2)}(2^{u-1} - (2^{u-1} - 1))3^{3 \cdot 2^{u-2}}2^{2^{u-1}}}{2^{u(2^u-2)}(2^u - (2^u - 1))} \\ &\approx \frac{\sqrt{3}(2^u - 1)}{\sqrt{2^{u-1} + 1}} \frac{2^{(u-1)(2^{u-1}-2)}3^{3 \cdot 2^{u-2}}2^{2^{u-1}}}{2^{u(2^u-2)}} \\ &\approx \frac{\sqrt{3}(2^u - 1)}{\sqrt{2^{u-1} + 1}} 2^{-u2^{u-1}+2+\log_2 3 \cdot 3 \cdot 2^{u-2}} \\ &\approx n \frac{\sqrt{3}}{\sqrt{n/2 + 1.5}} 2^{-(n+1)[\log_2(n+1)/2 - 1.188] + 2} \\ &= \Theta(\sqrt{n} 2^{-n \log_2 n}) \end{aligned}$$

Golay Codes:

Example 5.3. Consider the resilient function constructed from the perfect binary Golay code $G_{23} = [23, 12, 7]$. We tabulate the performance of the resilient w.r.t. t , the number of deterministic input bits as follows.

t	7	8	9	10
$N(t)$	253	4554	37950	194810
$\text{Prob}[I(t)]$	0.00103	0.00928	0.0464	0.170

Special Dual-BCH Codes:

Example 5.4. Consider the resilient function constructed from the dual of the double-error-correcting BCH code $[2^m - 1, 2m, 2^{m-1} - 2^{(m-1)/2}]$ [18, page 451]. For $m = 5$ we obtain a $[31, 19, 12]$ -code with the following performance table:

t	12	13	14	15	16	17
$N(t)$	310	5890	53010	300390	1201560	3604680
$\text{Prob}[I(t)]$	0.0000021	0.000028	0.00019	0.00099	0.0039	0.013

For $m = 9$, we have a $[511, 18, 224]$ -code and the performance of the resilient function has the following characteristics.

t	224	260	300	360
$\text{Prob}[I(t)]$	7.5×10^{-147}	1.3×10^{-102}	2.1×10^{-74}	1.4×10^{-44}

6 Entropy loss and higher spectra of codes

Let C be a binary $[n, m, d]$ -code. For a linear subcode C' of C define the *support* of C' to be

$$\text{supp}(C') = \{i \mid 1 \leq i \leq n, \exists c \in C' (c_i \neq 0)\}.$$

Then, for $0 \leq r \leq m$ and $0 \leq i \leq n$, define

$$A_i^{(r)} = A_i^{(r)}(C) = |\{C' \leq C : |\text{supp}(C')| = i, \dim C' = r\}|, \quad (6.1)$$

that is, $A_i^{(r)}$ is the number of r -dimensional linear subcodes of C having support of size i . The statistics $A_i^{(r)}$ record very detailed information about the structure of C . These generalize the usual coefficients of the weight enumerator A_i which count the number of codewords of weight i for each i . Some trivial values and relationships are the following

$$A_i^{(0)} = \delta_{i,0}, \quad A_0^{(r)} = \delta_{r,0}, \quad A_i^{(1)} = A_i \quad (\text{for } i = 1, \dots, n),$$

$$\sum_i A_i^{(r)} = \begin{bmatrix} m \\ r \end{bmatrix}_2$$

where $\begin{bmatrix} m \\ r \end{bmatrix}_2$ is the Gaussian coefficient denoting the number of r -dimensional subspaces of an m -dimensional vector space over the binary field $GF(2)$.

When authors speak of higher weights (or generalized Hamming weights, or Wei weights), they refer only to the integers

$$d_r = \min\{i \mid A_i^{(r)} \neq 0\}$$

for $r = 1, 2, \dots, m$. In [30], Wei introduced these ideas — higher weights and, implicitly, higher spectra — in an effort to better understand attacks on a wire-tap channel. Wei's work already had indirect implications for the theory of resilient functions.

We now establish the connection between the higher spectra and the entropy distribution of the resilient function.

Let C be a fixed binary $[n, m, d]$ -code. If c is a codeword, let $\text{supp}(c)$ denote the support of c . For $0 \leq i \leq n$ and $0 \leq r \leq m$, define

$$B_{i,r} = |\{S \subseteq [n] : |S| = i, \text{supp}(c) \subseteq S \text{ for exactly } 2^r \text{ codewords } c \in C\}|.$$

First note that, since C is binary linear, for any set S , the number of codewords having support contained in S is always a power of two. Now what is the relevance of these $B_{i,r}$ values? Indeed, if code C is employed as a resilient function as above and we know that exactly i input bits are deterministic (all others being independent and balanced), the probability that the corresponding output has entropy exactly $m - r$ is $B_{i,r} / \binom{n}{i}$. Thus we have completed the proof of

Lemma 6.1. *Let X be a random variable taking values in $\{0, 1\}^n$ according to a probability distribution $\mathcal{D}_{T,Z}$ as defined in Section 3. Then*

$$\text{Prob}[H_{\text{out}} = m - r \mid |T| = i] = B_{i,r} \binom{n}{i}^{-1}. \quad \square$$

Now the fundamental connection between these statistics and the higher spectra is given by the following

Proposition 6.2. *Let C be a binary $[n, m, d]$ -code with higher spectra $A_i^{(r)}$ and let $B_{i,r}$ be defined for C as above. Then, for each i ($0 \leq i \leq n$) and each r ($0 \leq r \leq m$), we have*

$$\sum_{k=0}^m \begin{bmatrix} k \\ r \end{bmatrix}_2 B_{i,k} = \sum_{h=0}^n \binom{n-h}{i-h} A_h^{(r)}.$$

Proof. This follows by double counting. Let

$$X = \{(C', S) : C' \leq C, \dim C' = r, |S| = i, \text{supp}(C') \subseteq S\}$$

and let us count in two ways the ordered pairs of linear subcodes of C of dimension r and sets of coordinates S of size i which contain their support. Choosing S first and then choosing a subcode of the largest subcode with this property, we obtain the quantity on the left. Choosing the subcode C' first and then locating sets S containing its support, we obtain the quantity on the right. \square

So we obtain $n + 1$ independent triangular systems, one for each $i = 0, 1, \dots, n$. The i^{th} set of equations involves only the unknowns $B_{i,0}, B_{i,1}, \dots, B_{i,m}$.

Now we employ a useful identity from the theory of special functions³.

³After proving this identity for ourselves, we came across it in [2], which addresses a closely related problem in coding theory and refers to [12].

Proposition 6.3. [See, e.g., [12]] Let q be a prime power, $n \geq 1$ and $0 \leq i, j \leq n$. Then

$$\sum_{k=0}^n (-1)^{k-j} q^{\binom{k-j}{2}} \begin{bmatrix} i \\ k \end{bmatrix}_q \begin{bmatrix} k \\ j \end{bmatrix}_q = \delta_{i,j}.$$

We will need only the case $q = 2$ here, so let us agree to suppress q from now on. Now fix i and abbreviate

$$X_{i,r} := \sum_{h=0}^n \binom{n-h}{i-h} A_h^{(r)}.$$

If we take these values as known, then for fixed i our linear system for the unknowns $B_{i,r}$ is

$$\begin{aligned} \begin{bmatrix} 0 \\ 0 \end{bmatrix} B_{i,0} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} B_{i,1} + \cdots + \begin{bmatrix} m \\ 0 \end{bmatrix} B_{i,m} &= X_{i,0} \\ \begin{bmatrix} 1 \\ 1 \end{bmatrix} B_{i,1} + \cdots + \begin{bmatrix} m \\ 1 \end{bmatrix} B_{i,m} &= X_{i,1} \\ &\vdots = \vdots \\ \begin{bmatrix} m \\ m \end{bmatrix} B_{i,m} &= X_{i,m} \end{aligned}$$

Applying Proposition 6.3, we solve to find

$$B_{i,r} = \sum_{k=0}^m (-1)^{k-r} 2^{\binom{k-r}{2}} \begin{bmatrix} k \\ r \end{bmatrix} X_{i,k}. \quad (6.2)$$

In this way, knowledge of the full range of higher spectra gives us the statistics $B_{i,r}$ and, in turn, the full probability distribution on the output entropy given any specified number of deterministic input bits.

Proposition 6.4. Let F be the resilient function constructed using binary linear $[n, m, d]$ -code C with higher spectra $A_i^{(r)}$ ($0 \leq i \leq n$, $0 \leq r \leq m$) as defined in Equation (6.1). Then the number $B_{i,r}$ of i -element subsets of the coordinates $[n]$ containing exactly 2^r codewords is given by

$$B_{i,r} = \sum_{k=0}^m \sum_{h=0}^n (-1)^{k-r} 2^{\binom{k-r}{2}} \binom{n-h}{i-h} \begin{bmatrix} k \\ r \end{bmatrix} A_h^{(k)}$$

Proof. Indeed the matrix $M = [m_{k,j}]_{k,j=0}^m$ with entries $m_{k,j} = \begin{bmatrix} j \\ k \end{bmatrix}$ has inverse $C = [c_{i,k}]_{i,k=0}^m$ given by

$$c_{i,k} = (-1)^{k-i} 2^{\binom{k-i}{2}} \begin{bmatrix} k \\ i \end{bmatrix}.$$

So the expression for $B_{i,r}$ in terms of the higher spectra follows from Proposition 6.3 and the definition of the values $X_{i,r}$. \square

Finally, we wish to show how the values $B_{i,r}$ enable us to find a good lower bound on the output entropy. Some further analysis could perhaps lead to an exact expression, but the estimate we obtain is sufficient for our purposes.

One easily checks that the function $h(x) = -x \log_2 x$ is concave: $ph(x) + (1-p)h(y) \leq h(px + (1-p)y)$ for $0 \leq p, x, y \leq 1$. More generally, if \mathcal{D} is a probability distribution on a set S with probability density function $\mathcal{D}(a) = x_a$ for $a \in S$ and if \mathcal{E} is another distribution on S with probability density function $\mathcal{E}(a) = y_a$, then

$$H(\mathcal{D}) = \sum_{a \in S} -x_a \log_2 x_a, \quad H(\mathcal{E}) = \sum_{a \in S} -y_a \log_2 y_a.$$

Now a convex combination \mathcal{F} of these two distributions has probability density function $px_a + (1-p)y_a$ for $a \in S$ and its entropy

$$H(\mathcal{F}) = \sum_{a \in S} h(px_a + (1-p)y_a) \geq \sum_{a \in S} ph(x_a) + (1-p)h(y_a)$$

is bounded below by $pH(\mathcal{D}) + (1-p)H(\mathcal{E})$ since h defined above is concave.

For our application, let us assume a fixed number i of deterministic coordinates. We assume that all $\binom{n}{i}$ combinations of coordinate positions are equally likely in our non-adversarial model. Each of these combinations yields an output distribution which is uniform on some linear subspace of \mathbb{Z}_2^m and the output entropy for such a distribution is exactly the dimension of this subspace. We have already used the higher spectra of the code to determine the values $B_{i,r}$ which give the number of i -element combinations of coordinates for which the output entropy is $m - r$. So, again with i fixed, and positions of deterministic coordinates chosen uniformly at random, the entropy of the output distribution is bounded below by

$$\sum_{r=0}^m \frac{B_{i,r}}{\binom{n}{i}} (m - r),$$

which is the expected value of the entropy for a fixed but random selection of i deterministic coordinates. This completes the proof of the following

Theorem 6.5. *Given $F : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$ of the form $x \mapsto Gx$ where G is a generator matrix for the linear code C with higher spectra $A_i^{(r)}$, the expected value of the entropy of the output distribution \mathcal{E} of F conditioned on having exactly i deterministic input bits and the remaining $n - i$ bits independent and balanced is bounded below by*

$$\mathcal{H}(i) = \sum_{r=0}^m \sum_{k=0}^m \sum_{h=0}^n (-1)^{k-r} (m - r) 2^{\binom{k-r}{2}} \frac{i(i-1) \cdots (i-h+1)}{n(n-1) \cdots (n-h+1)} \begin{bmatrix} k \\ r \end{bmatrix} A_h^{(k)} \quad (6.3)$$

where the expected value is taken over all possible choices of i deterministic coordinates, each with equal probability.

Proof. We have just seen that concavity of h implies that $\mathcal{H}(i)$ is a valid lower bound on the output entropy, and it is simply computed as an expected value $\sum_r (m - r) B_{i,r} \binom{n}{i}^{-1}$, which simplifies to the expression given using Proposition 6.4. \square

Example 6.6. The first order Reed-Muller code $\mathcal{R}_{1,4}$ is a $[16, 5, 8]$ -code with higher weight spectra given in Table 16 in [11]. Using Theorem 6.5, we obtain in Figure 1 the profile for $\mathcal{H}(i)$, giving a lower bound on the output entropy.

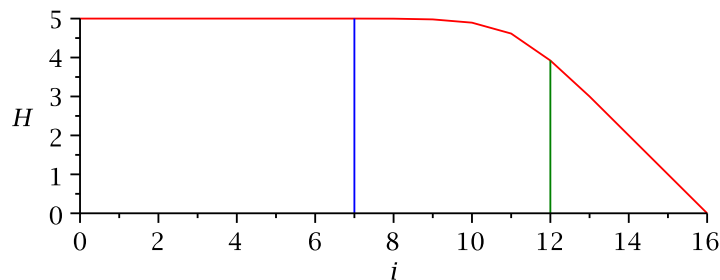


Figure 1: Expectation for output entropy for F from a $[16, 5, 8]$ -code as a function of the number of deterministic input bits. The vertical lines are at $d - 1 = 7$, up to which point previous results guarantee perfect entropy, and $1.5d = 12$, the limit addressed in Theorem 3.2.

Example 6.7. The second order Reed-Muller code $\mathcal{R}_{2,4}$ is a $[16, 11, 4]$ -code with higher weight spectra given in Table 17 in [11]. We obtain the profile for our lower bound $\mathcal{H}(i)$ in Figure 2.

Example 6.8. The extended binary Golay code is a $[24, 12, 8]$ -code with well-known weight enumerator. The higher spectra were first computed by Dougherty et al. in [10]. From this, we obtain in Figure 3 the graph of $\mathcal{H}(i)$ for this code.

7 Conclusion

We have considered, at the theoretical level, the behavior of a linear resilient function when its inputs degrade beyond acceptable levels. We find not only that the function still performs

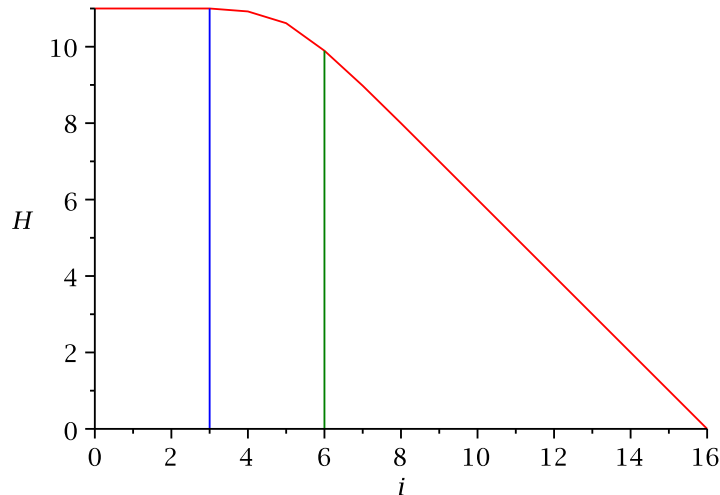


Figure 2: Expectation for output entropy for F from a $[16, 11, 4]$ -code as a function of the number of deterministic input bits. The vertical lines are at $d - 1 = 3$ and $1.5d = 6$.

well with high probability, but that one can completely characterize this behavior when the inputs are assumed to be independent.

The first part of the analysis which determines the output distribution of the resilient function up to halfway beyond the resiliency degree already accomplishes quite a bit with very little information about the underlying linear code. Given more detailed code statistics, we get better entropy estimates. A crucial tool in the latter part of the analysis is the theory of higher weights and higher spectra of linear codes, introduced by Wei in his study of the closely related wire-tap channel of Type II. At the end of his seminal paper, Wei wrote “The generalized Hamming weights also characterize a linear code’s performance as a t -resilient function, in every detail.” What is remarkable here is that we find an applied setting which demands even more detail than the higher weights can provide, thereby demonstrating an applied need for more information about the exact higher spectra of important linear codes. We hope that this paper will serve as motivation to investigate this rich area further.

Acknowledgments

This material is based upon work supported by the US National Science Foundation under Grants No. ANI-0112889, and CAREER Award ANI-0133297. B. Sunar’s work was

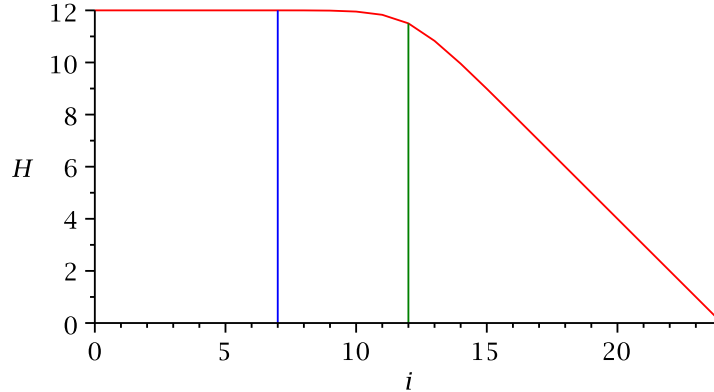


Figure 3: Expected value of output entropy for F from the extended binary Golay code as a function of the number of deterministic input bits.

supported in part by the National Science Foundation through Cybertrust grant No. CNS-0831416 and W. Martin's work was supported by the National Security Agency through grant No. H98230-07-1-0025. The authors thank Steven Dougherty for providing up-to-date information on higher spectra of codes.

References

- [1] A. Akavia, S. Goldwasser and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. pp. 474-495, in: *Theory of Cryptology Conference (TCC)* (LNCS **5444**), O. Reingold, ed., 2009.
- [2] A. Barg and A. Ashikhmin. Binomial moments of the distance distribution and the probability of undetected error. *Designs, Codes, Crypt.* **16** (1999), 103-116.
- [3] C.H. Bennett, G. Brassard, and J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, **17** (1988), no. 2, 210-229.

- [4] A. Braeken, V. Nikov, S. Nikova, B. Preneel. On boolean functions with generalized cryptographic properties. pp. 120-135, in: Progress in Cryptology - INDOCRYPT 2004 (LNCS vol. 3348), Springer Berlin, Heidelberg, 2005.
- [5] Andries E. Brouwer. Server for bounds on the minimum distance of q -ary linear codes, $q = 2, 3, 4, 5, 7, 8, 9$. URL: <http://www.win.tue.nl/~aeb/voorlincod.html>
- [6] R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz and A. Sahai. Exposure-Resilient Functions and All-or-Nothing Transforms. pp. 453-469, in: *EUROCRYPT 2000*.
- [7] C. J. Colbourn, J. H. Dinitz, and D. R. Stinson. Applications of combinatorial designs to communications, cryptography and networking, pp. 37-100, in: *Surveys in Combinatorics* (J.D. Lamb and D.A. Preece, eds.), London Mathematical Society, 1999.
- [8] R. Gennaro, A. Lysyanskaya, T. Malkin, S. Micali, and T. Rabin. Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. pp. 258-277, in: *Theory of Cryptography*, (LNCS **2951**), 2004.
- [9] B. Chor, O. Goldreich, J. Håstad, J. Friedman, S. Rudich and R. Smolensky. The bit extraction problem or t -resilient functions, pp. 396-407, in: *26th IEEE Symposium on Foundations of Computer Science*, 1985.
- [10] S. T. Dougherty, T. A. Gulliver and M. Oura. Higher weights and graded rings for binary self-dual codes. *Discr. Appl. Math.* **128** (2003), 121–143.
- [11] S. T. Dougherty and S. Han. Higher weights and generalized MDS codes. Preprint, 2009.
- [12] N. J. Fine. Hypergeometric Series and Applications. American Math. Soc., Providence, 1988.
- [13] M. Finiasz. Words of minimal weight and weight distribution of binary Goppa codes. *International Symp. Info. Th.*, Yokohama Japan, 2003.
- [14] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum and E. W. Felten. Lest We Remember: Cold Boot Attacks on Encryption Keys, pp. 331-346 in: *Proc. of 17th USENIX Security Symposium (Sec '08)*, San Jose, CA, July 2008.
- [15] Y. Ishai, A. Sahai and D. Wagner, Private Circuits: Securing Hardware against Probing Attacks, pp. 463-481, in: *Proceedings of CRYPTO 2003*, Springer-Verlag, 2003.
- [16] T. Kasami, T. Fujiwara, Lin Shu. An approximation to the weight distribution of binary linear codes. *IEEE Trans. Info. Th.*, **31** (1985), no. 6, 769–780.
- [17] J. H. Van Lint. Introduction to Coding Theory, Springer-Verlag, New York, 1998.

- [18] F. J. MacWilliams and N. J. A. Sloane. The Theory of Error-Correcting Codes, North Holland, Amsterdam, 1977.
- [19] S. Micali and L. Reyzin. Physically observable cryptography (extended abstract). pp. 278-296, in: *TCC*, 2004.
- [20] C. Munuera. On the generalized Hamming weights of geometric Goppa codes. *IEEE Trans. Info. Th.* **40** (1994), no. 6, 2092-2099.
- [21] M. Naor and G. Segev. Public-Key Cryptosystems Resilient to Key Leakage. pp. 18-35, in: *Advances in Cryptology - CRYPTO '09*, (LNCS **5677**). Springer, 2009.
- [22] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. pp. 84-93, in: *Proc. 37th Annual ACM Symp. on Theory of Comput.*, 2005.
- [23] A. Shamir, N. van Someren. Playing “hide and seek” with stored keys. pp. 118-124 in: *Proceedings of Financial Cryptography* (LNCS **1648**), 1999.
- [24] S. P. Skorobogatov. Data remanence in flash memory devices. pp. 339-353, in: *CHES* (LNCS **3659**, J. R. Rao and B. Sunar, eds.) Springer, 2005.
- [25] N. van Someren, How not to authenticate code. Crypto '98 Rump Session, Santa Barbara, 1998.
- [26] A. Shamir. How to share a secret, *Commun. ACM* **22**, no. 11 (1979), 612-613.
- [27] D. R. Stinson and K. Gopalakrishnan. Applications of Designs to Cryptography, pp. 549-557 in: CRC Handbook of Combinatorial Designs (C. D. Colbourn, and J. H. Dinitz, eds.), CRC Press, 1996.
- [28] B. Sunar, W. J. Martin, and D. R. Stinson. A Provably Secure True Random Number Generator with Built-in Tolerance to Active Attacks. *IEEE Trans. Computers* **58** (2007), no. 1, 109-119.
- [29] M. A. Tsfasman and S. G. Vladut. Geometric approach to higher weights. *IEEE Trans. Info. Th.* **41** (1995), no. 6, 1564-1588.
- [30] V. K. Wei. Generalized Hamming weights for linear codes. *IEEE Trans. Info. Th.* **37** (1991), no. 5, 1412-1418.
- [31] X.-M. Zhang and Y. Zheng. Cryptographically resilient functions. *IEEE Trans. Info. Th.* **43** (1997), no. 5, 1740-1747.