Relations between the Entropy of a Source and the Error Masking Probability for Security Oriented Codes

Osnat Keren, *Member, IEEE,* and Mark Karpovsky, *Fellow, IEEE*

***Abstract*—Security-oriented error detecting codes are used to detect fault injection attacks on cryptographic devices. These codes are usually designed for uniformly distributed codewords; i.e., for codes that have maximal entropy. In practice, the codewords are not uniformly distributed thus their entropy is smaller and their efficiency in detecting attacks degrades. This article analyzes the relation between the entropy of a code and its worst error masking probability. Based on this relation, a method for determining the rate and structure of a code that provides the required error masking probability is presented.**

I. Introduction

The security of cryptographic devices is threatened by fault injection attacks that tamper with the hardware [2]. All fault attack technologies are based on introducing faults into the circuit, which in turn, can produce an erroneous output vector. The erroneous output is usually modeled as a correct output distorted by an additive error vector. This could suggest that the problem of fault detection is equivalent to the problem of error detection in noisy communication channels. However, there is an inherent difference between the two problems: in fault injection attacks, the error vector is chosen by the attacker, and hence, it may be of any weight (multiplicity). Consequently, the efficiency of the code is not measured in terms of its error correcting capability, but in terms of its error masking probability. From this perspective, error injection attacks on hardware are similar to jamming attacks on communication channels. Both can be detected by the same type of codes.

The error masking probability of a code is the probability that an error will map a codeword into the code. The maximal error masking probability depends on the structure of the code, the encoding, and the probability of each codeword to appear.

Security oriented codes that can detect any (or almost any) error have been presented in [1], [4], [5], [6], [9], [10], [19]. These codes were designed for uniformly distributed codewords. However, the efficiency of these codes, may significantly degrade when the codewords are not equally likely to occur [13], [14]. In this article, we evaluate the efficiency of codes whose codewords are not uniformly distributed. We show that the worst error masking probability of the codes is related to the entropy of the code. The bounds presented here define the entropies for which robust codes *fail to provide* the required error masking probability. In such cases, a pre-mapping [12] or stronger codes, such as the Algebraic Manipulation Detection (AMD) code [7] whose error masking probability does not depend on the code's entropy are needed. In Section IV we show that for non-uniformly distributed codes, these stronger codes are not necessarily stronger than simple codes. We then show how the bounds can be used to determine which type of code has a smaller error masking probability.

The article is organized as follows. In the next section we define the error masking probability of the code and introduce the worst case scenario. In Section III we discuss the relation between the worst error masking probability of a code and its entropy and provide upper and lower bounds on this probability. Section IV presents a criterion for choosing a code for a given rate and entropy. In Section V we demonstrate the tightness of the bounds on sequential benchmark circuits and show that it is pointless to design a security-oriented code if its entropy is ignored. Section VI concludes the paper.

II. The hardware security problem

Consider a combinational circuit that every cycle receives an input x, and produces an output $c(x) \in \mathcal{C}$, where $\mathcal{C} \subseteq \mathbb{F}_2^n$ is the set of all possible outputs of a fault-free circuit. The set $\mathcal{C}$ is called a code, and the elements in $\mathcal{C}$ are called codewords.

We refer to the input of the circuit as a discrete random variable X from a sample space (alphabet) $\mathcal{X}$. The probability that this random variable will take the value x is denoted as $\mathtt{Pr}(X = x)$. Similarly, we refer to the output of the circuit as a random variable C from a sample space $\mathcal{C} \subseteq \mathbb{F}_2^n$. The Probability Mass Distribution (PMD) of C (in a fault-free circuit) is determined by the PMD of the inputs and the functionality of the circuit. The probability that a random variable C will take the value c_i is

$$p(c_i) \triangleq \mathtt{Pr}(C = c_i) = \sum_{x \in \mathcal{X}, c(x) = c_i} \mathtt{Pr}(X = x).$$

We write the PMD of a code as a vector, $\underline{p} = (p_1, p_2, \dots, p_{|\mathcal{C}|})$, where p_i is the probability that the discrete random variable C will take the i'th value. For convenience, we order the codewords so that

$$1 \geq p_1 \geq p_2 \cdots \geq p_{|\mathcal{C}|} > 0.$$

O.Keren is with the Faculty of Engineering, Bar-Ilan University, Israel. e-mail: osnat.keren@biu.ac.il

M. Karpovsky is with the Department of Electrical and Computer Engineering, Boston University, USA. e-mail: markkar@bu.edu

The research of the first author was supported by the ISRAEL SCIENCE FOUNDATION (grant No. 1200/12). The research of the second author was supported by NSF grant CNS1012910.

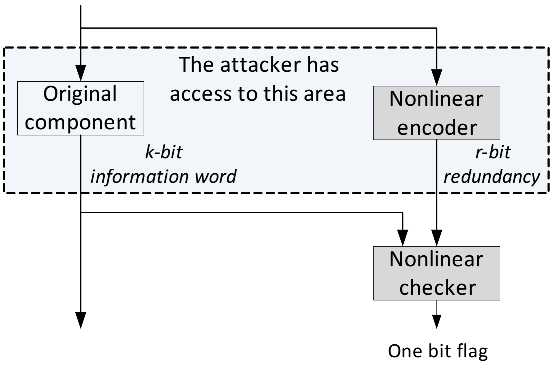

Fig. 1. A schematic architecture of a circuit component protected by a systematic security-oriented code. The white rectangle represents the original circuit and the gray rectangles represent the redundant hardware required to implement the encode and decoder. The shaded area is accessible to the attacker. The original block and the encoder must work in parallel, otherwise it is impossible to detect attacks.

The entropy of a random variable C with a PMD $\underline{p}$ is denoted by $H(C)$ and defined by

$$H(C) = H(\underline{p}) \triangleq -\sum_{i=1}^{|\mathcal{C}|} p_i \log_2(p_i) \quad \text{(bits)}.$$

The entropy of a binary random variable with PMD $\underline{p} = (1-p, p)$ is denoted by

$$h(p) \triangleq -p\log_2(p) - (1-p)\log_2(1-p).$$

The random variable C is a binary vector that represents the values on the wires at the output of the circuit at a certain clock cycle. The entropy $H(C)$ represents the minimal number of wires required to deliver the information carried by C. The wires of the unprotected circuit may be connected to different blocks, and each (or some of them) may represent a different variable; therefore, no data compression which can reduce the number of output wires of the unprotected circuit is allowed. Moreover, in order to provide reliability against random errors of small weight (i.e., errors caused by nature), and to provide security against fault injection attacks, some redundancy should be embedded into the circuit. A schematic architecture of a trustworthy circuit is shown in Fig. 1. We denote by k the number of original wires at the output of the unprotected circuit, and denote by r, $r = n-k \geq 0$, the number of redundant wires. The difference between the number of original wires k and $H(C)$ reflects overhead required for having a simple and convenient interface between the circuit blocks. The difference between k and n reflects the overhead required to have a trustworthy system[1]. Clearly, the entropy of the codewords is the entropy of the original outputs, and thus we have,

$$H(C) \leq \log_2 |\mathcal{C}| \leq k \leq n.$$

The efficiency of a code with respect to an error e is measured in terms of its error masking probability $Q(e)$. $Q(e)$ is the probability that an error e is masked by codewords in $\mathcal{C}$; i.e.,

$$Q(e) = \sum_{c, c\oplus e \in \mathcal{C}} p(c), \tag{1}$$

where the $\oplus$ sign stands for addition in $\mathbb{F}_2^n$ and the $\sum$ stands for addition in $\mathbb{R}$. The set of errors that are never detected form the *detection kernel* K_d of the code,

$$K_d = \{e : Q(e) = 1\}.$$

A code whose detection kernel is $K_d = \{0\}$ is called *robust*, and a code whose kernel is of size $1 < |K_d| < |\mathcal{C}|$, is called a *partially robust* code [6].

A code is characterized by its *maximal* error masking probability:

Definition 1: The error masking probability of a code is defined as $Q^* = \max_{e \notin K_d} Q(e)$.

Let $\delta(\tau)$ be the characteristic function of a code $\mathcal{C}$, $\delta(\tau) = 1$ if $\tau \in \mathcal{C}$, and equals zero otherwise. The error masking probability Q^* of a robust code is lower bounded by the average $Q(e)$ over all the nonzero errors. The average error masking probability is denoted by Q_{opt}; a code whose error masking probability equals Q_{opt} is called *optimum.*

Property 1:

$$Q_{opt} = \frac{|\mathcal{C}| - 1}{2^n - 1}. \tag{2}$$

Proof: The error masking probability equals

$$Q(e) = \sum_{c, c\oplus e \in \mathcal{C}} p(c) = \sum_{c \in \mathcal{C}} p(c)\delta(c \oplus e).$$

The average error masking probability over all the nonzero error vectors is

$$\begin{aligned} Q_{opt} &= \frac{\sum_{e \in \mathbb{F}_2^n \setminus \{\mathbf{0}\}} Q(e)}{2^n - 1} \\ &= \frac{\sum_{c \in \mathcal{C}} p(c) \left(\sum_{e \in \mathbb{F}_2^n} \delta(c \oplus e) \right) - \sum_{c \in \mathcal{C}} p(c)\delta(c \oplus 0)}{2^n - 1}. \end{aligned}$$

Since $\sum_{e \in \mathbb{F}_2^n} \delta(c \oplus e) = |\mathcal{C}|$ we get Eq. 2. ■

Note that Q_{opt} does not depend on the PMD of the codewords.

Another parameter that characterizes the code and does not depend on the PMD of the codewords is the probability Q_{mc}, where the subscript *'mc'* stands for *maximal correlation*:

Definition 2 ([6]): Denote by $R(e)$ the autocorrelation function $R(e) \triangleq \sum_{\tau \in F_2^n} \delta(\tau)\delta(\tau \oplus e) = |\{c | c, c \oplus e \in \mathcal{C}\}|$. The probability Q_{mc} is defined by

$$Q_{mc} \triangleq \frac{\max_{e \neq 0} R(e)}{|\mathcal{C}|}. \tag{3}$$

Note that the value of $R(e)$ equals the number of codewords that mask the error e. If there exists a nonzero error e for which $R(e) = |\mathcal{C}|$, the code is not robust, regardless of how the codewords are distributed. In this article we are interested in robust codes, i.e. code that can detect any (nonzero) error. To date, there are only two such codes, the Quadratic-Sum code

[1] We assume that a faulty network (i.e., a faulty combinational circuit) is still combinational. In other words, we assume that a fault cannot turn a memoryless circuit into an asynchronous sequential circuit. Otherwise, the circuit may become unstable and unpredictable.

[5], [8], and the Punctured-Cubic code [1], [9]. The Quadratic-Sum code is optimum for $k = 2r$; i.e., $Q_{mc} = 2^{-r}$. The Q_{mc} of the Punctured-Cubic code is smaller or equal to 2^{-r+1}, depending on the code's parameters [9].

The value of Q_{mc} (which equals Q^* for uniformly distributed codewords) characterizes the set of codewords. The maximal number of codewords that mask a nonzero error equals $Q_{mc}|\mathcal{C}|$. This defines the worst scenario.

Definition 3: The worst error masking probability of a code with a PMD $\underline{p}$, where $p_i \geq p_{i+1}$, is denoted by Q_{wc} and is defined as:

$$Q_{wc} \triangleq \sum_{i=1}^{Q_{mc}|\mathcal{C}|} p_i. \tag{4}$$

Note that $Q_{wc} \geq Q_{mc}$, and equality holds iff the codewords are uniformly distributed; i.e., iff $H(\mathcal{C}) = \log_2(|\mathcal{C}|)$. The worst error masking probability is used to characterise the code, in practice the (true) error masking probability of the code Q^* can be significantly smaller. Methods for avoiding this worst case scenario were presented in [13], [12].

Although Q_{mc} lower bounds Q_{wc}, it is not necessarily smaller than Q^*. The following example clarifies this statement.

Example 1: Fig. 2 shows a logic scheme of a circuit that generates non-uniformly distributed codewords. The circuit has an input $x = (x_4, x_3, x_2, x_1) \in \mathbb{F}_2^4$ and produces an output $c = (c_3, c_2, c_1) \in \mathbb{F}_2^3$. The legal output combinations (codewords) are

$$\mathcal{C} = \{000, 010, 011, 110, 111\}.$$

For convenience we refer to an output by its integer value; namely, $\mathcal{C} = \{0, 2, 3, 6, 7\}$. For uniformly distributed input ($H(X) = 4$), the PMD of the codewords in a fault-free circuit is:

$$p(c) = \begin{cases} 8/16 & c_1 = 0 \quad \text{(since eight inputs} \\ & \qquad\qquad \text{produce the output "0")} \\ 3/16 & c_2 = 6 \\ 2/16 & c_3 = 2 \\ 2/16 & c_4 = 7 \\ 1/16 & c_5 = 3 \end{cases} .$$

The parameters that characterize the set of codewords (ignoring the probability of each to appear) are:

- The optimal error masking probability, $Q_{opt} = 4/7$.
- The autocorrelation of the code.

$$\begin{aligned} R(e) &= \sum_{\tau \in F_2^3} \delta(\tau)\delta(\tau \oplus e) \\ &= \begin{cases} 5 & e = 0 \\ 4 & e \in \{1, 4, 5\} \\ 2 & e \in \{2, 3, 6, 7\} \end{cases} \end{aligned} \tag{5}$$

- The error masking probability $Q_{mc} = \max_{e \neq 0} R(e)/|\mathcal{C}| = 4/5$.

The parameters that characterize the code and depend on probability of each codeword to appear are:

- The entropy of the code is $H(C) = 1.95$.
- The worst error masking probability is $Q_{wc} = (8 + 3 + 2 + 2)/16 = 15/16$.
- The (true) error masking probability of the code $Q^* = Q(e = 6) = 11/16$.

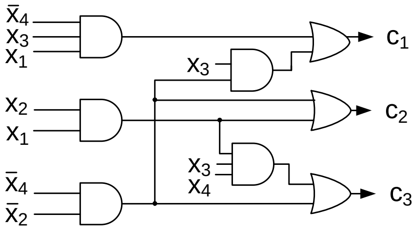

Fig. 2. Logic scheme of the circuit in Ex. 1

Note that $\mathcal{C}$ is robust, but it is not optimum as it does not meet its lower bound. In fact, for this code we have

$$Q_{opt} < Q^* < Q_{mc} < Q_{wc}.$$

In practice, it is difficult to calculate the error masking probabilities of codes with large n, especially when the codewords are not uniformly distributed. Therefore, the code to be used (equivalently, the number of redundant bits, r, to be added) needs to be chosen such that Q_{wc} is acceptable. The question we address is thus:

Let $\mathcal{C}$ be a robust code characterized by Q_{mc}. Assume that $H(C)$ is known. What can be said about Q_{wc}?

III. Bounds on the error masking probability

Consider a circuit that generates $|\mathcal{C}|$ different codewords with a PMD $\underline{p} = (p_1, p_2, \dots p_{|\mathcal{C}|})$ where $1 \geq p_1 \geq p_2 \cdots \geq p_{|\mathcal{C}|} > 0$. Assume that the code designer does not know $\underline{p}$, but he knows or can estimate the entropy, and he knows Q_{mc} of the code to be used. In this section we introduce lower and upper bounds on Q_{wc} as a function of $H(C)$ and Q_{mc}. In order to simplify the presentation and make the text more readable, instead of introducing lower and upper bounds on Q_{wc}, we introduce upper and lower bounds on $H(C)$ as a function of Q_{wc} and Q_{mc}.

Let us start by defining the (non-symmetric) distance between two PMDs:

Definition 4 ([3]): The distance between two PMDs $\underline{p}$ and $\underline{q}$ (also called the Kullback - Leibler divergence or the information divergence), is denoted by $D(\underline{p}||\underline{q})$, and defined as:

$$D(\underline{p}||\underline{q}) \triangleq \sum_i p_i \log\left(\frac{p_i}{q_i}\right) \geq 0.$$

The distance between two binary distributions $\underline{p} = (1 - p, p)$ and $\underline{q} = (1 - q, q)$ is denoted by $D_b(p||q)$, and defined as:

$$D_b(p||q) \triangleq p \log\left(\frac{p}{q}\right) + (1 - p) \log\left(\frac{1 - p}{1 - q}\right).$$

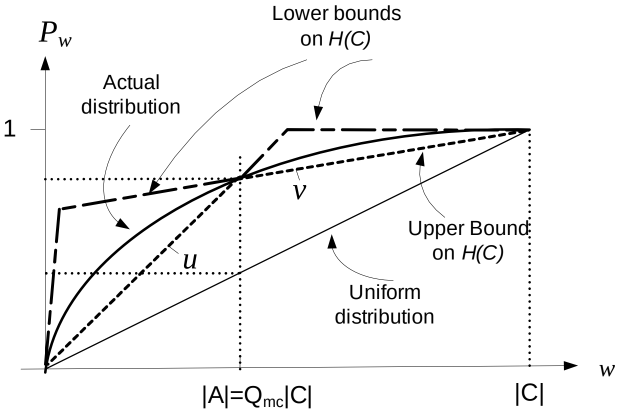

Fig. 3. The accumulated probability P_w for uniformly distributed codewords, the actual PMD of the codewords, the PMD used to derive the upper bound on $H(C)$ (Theorem 1), and the two PMDs used to derive the lower bound on the entropy (Theorem 2).

For the proofs of Theorems 1 and 2 presented below, in addition to the PMD of the code, we use another PMD, say $\underline{q}$, which like $\underline{p}$ has the following two properties:

$$q_i \geq q_{i+1} \quad \text{and} \quad \sum_{i=1}^{Q_{mc}|\mathcal{C}|} q_i = Q_{wc}.$$

Figure 3 shows the PMDs that were used to derive the upper and lower bounds on the entropy. To simplify the graph, the Y-axis represents an accumulated probability, $P_w = \sum_{i=1}^{w} p_i$, and the X-axis represents the index w. The X-axis is divided into two parts; in each part, the slope of the curve that connects the P_w's does not increase as w grows. This reflects the requirement that q_i should be greater or equal to q_{i+1}.

Theorem 1 (Upper bound): Let $\mathcal{C}$ be a code characterized by Q_{mc} and $H(C)$. Then,

$$H(C) \leq \log_2 |\mathcal{C}| - D_b(Q_{wc}||Q_{mc}). \tag{6}$$

Proof: Denote by $\underline{p}$ the PMD of the code. Let A be the set of $Q_{mc}|\mathcal{C}|$ most probable codewords, and let $B = \mathcal{C} \setminus A$. In the worst case scenario, there exists an error e that is masked by all the codewords in A; i.e., $Q_{wc} = \sum_{i=1}^{|A|} p_i$. Define $\underline{u} = (u_1, \ldots, u_{|A|})$ and $\underline{v} = (v_1, \ldots v_{|\mathcal{C}|-|A|})$ as the conditioned probabilities that a specific codeword from set A or B (respectively) is used:

$$\begin{cases} u_i = \frac{p_i}{Q_{wc}} & 1 \leq i \leq |A|, \\ \\ v_j = \frac{p_{j+|A|}}{1-Q_{wc}} & 1 \leq j \leq |\mathcal{C}| - |A|. \end{cases} \tag{7}$$

Clearly,

$$\sum_i u_i = \sum_j v_j = 1.$$

The entropy of the code in terms of $\underline{u}$ and $\underline{v}$ is

$$\begin{aligned} H(C) &= -\left[\sum_{i=1}^{|A|} u_i \cdot Q_{wc} \log(u_i Q_{wc}) \right. \\ &\quad \left. + \sum_{j=1}^{|\mathcal{C}|-|A|} v_j (1-Q_{wc}) \log(v_j(1-Q_{wc}))\right] \\ &= h(Q_{wc}) + Q_{wc} H(\underline{u}) + (1-Q_{wc}) H(\underline{v}). \end{aligned} \tag{8}$$

The entropy of a random variable is maximized if it is uniformly distributed. Therefore, we have:

$$\begin{aligned} H(C) &\leq h(Q_{wc}) + Q_{wc} \log_2(|A|) \\ &\quad + (1-Q_{wc}) \log_2(|B|) \\ &= \log_2 |\mathcal{C}| - D_b(Q_{wc}||Q_{mc}). \end{aligned}$$

■

Remark: A table with Q_{mc} values of known security oriented codes can be found in [18]. If $\mathcal{C}$ is an arbitrary set of vectors, or if its Q_{mc} is unknown, the computational complexity to compute the bound is $\mathcal{O}(n2^n)$.

We now turn to develop a lower bound on the entropy of $\mathcal{C}$. For this, we need the following three lemmas.

Lemma 1: Let Z be a discrete random variable over an alphabet of size N. Denote by $\underline{p}$, $p_1 \geq p_2 \geq \cdots \geq p_N > 0$, its PMD. Let $1/N < t < 1$. Then, the minimal entropy over all the PMDs that satisfy the restriction $p_1 \leq t$ is denoted by $\mathcal{H}_1(N, t)$, and is achieved when

$$p_i = \begin{cases} t & i \leq w = \lfloor 1/t \rfloor \\ 1 - w \cdot t & i = w + 1 \\ 0 & i > w + 1 \end{cases}.$$

Proof: Assume that the entropy is minimized by a different PMD, say $\underline{q}$, $t \geq q_1 \geq q_2 \geq \cdots \geq q_N > 0$. Since the two PMDs differ, there exist at least two indices where they have different values. Denote by i_a the minimal index for which $q_i < p_i$, and denote by i_b the maximal index where $q_i > p_i$; see to Figure 4. Clearly, $i_a < i_b$. Define as i_c the minimal index for which $q_{i_a} > q_i$ and by i_d the maximal index for which $q_{i_b} < q_i$. Let ϵ be

$$0 < \epsilon < \frac{1}{2} \min(p_{i_a} - q_{i_a},\ q_{i_a} - q_{i_c},\ q_{i_b} - p_{i_b},\ q_{i_d} - q_{i_b}).$$

The following two PMDs $\underline{b}$ and $\underline{l}$ fulfill the order requirement:

$$b_i = \begin{cases} q_i + \frac{\epsilon}{i_c - i_a} & i_a \leq i < i_c \\ q_i - \frac{\epsilon}{i_b - i_d} & i_d < i \leq i_b \\ q_i & otherwise \end{cases},$$

and

$$l_i = \begin{cases} q_i - \frac{\epsilon}{i_c - i_a} & i_a \leq i < i_c \\ q_i + \frac{\epsilon}{i_b - i_d} & i_d < i \leq i_b \\ q_i & otherwise \end{cases}.$$

Note that $\underline{q} = \frac{1}{2}(\underline{b} + \underline{l})$.

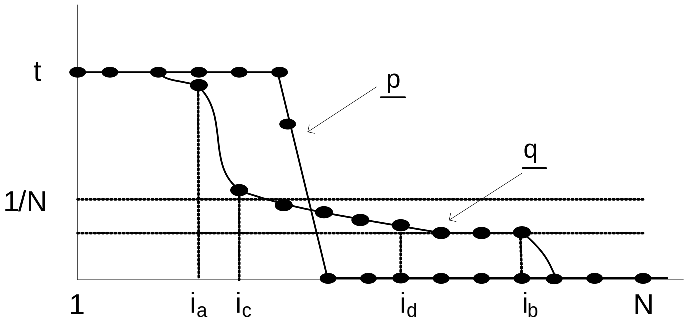

Fig. 4. Illustration for the proof of Lemma 1.

The entropy is *concave*; therefore, we have

$$\begin{aligned} H(\underline{q}) &= H(\frac{1}{2}\underline{b}+\frac{1}{2}\underline{l}) \\ &\geq \frac{1}{2}H(\underline{b})+\frac{1}{2}H(\underline{l}) \\ &\geq \min(H(\underline{b}),H(\underline{l})), \end{aligned}$$

(equality holds only if $\underline{b}=\underline{l}$). This contradicts the fact that $\underline{q}$ achieves minimal entropy under the above restriction. ■

Lemma 2: Let Z be a discrete random variable over an alphabet of size N. Denote by $\underline{p}$, $p_1 \geq p_2 \geq \cdots \geq p_N > 0$, its PMD. Let $0 < t < 1/N$. Then, the minimal entropy over all the PMDs that satisfy the restriction $p_N = t$ is denoted by $\mathcal{H}_N(N,t)$, and is achieved when

$$p_i = \begin{cases} 1-(N-1)t & i=1 \\ t & 2\leq i\leq N \end{cases}.$$

The proof of Lemma 2 is similar to the proof of Lemma 1.

Lemma 3: Let Z be a discrete random variable over an alphabet of size N. Denote by $\underline{p}$, $p_1 \geq p_2 \geq \cdots \geq p_N > 0$, its PMD. Then, the minimal entropy over all the PMDs that satisfy the restrictions $p_w = s$ and $\sum_{i=1}^{w} p_i = t$ is denoted by $\mathcal{H}_w(N,s,t)$, and is achieved when

$$p_i = \begin{cases} t-(w-1)s & i=1 \\ s & 2\leq i\leq N' \\ 1-(t+(N'-w)s) & i=N'+1 \\ 0 & otherwize \end{cases}.$$

where $N' = w + \lfloor \frac{1-t}{s} \rfloor$.

Proof: Assume that $\underline{p}$ does not minimize the entropy. Then, there exists another PMD, say $\underline{q}$, that fulfills the restrictions and minimizes the entropy. Define, $\underline{u}$ and $\underline{v}$ as

$$\begin{cases} u_i = \frac{q_i}{t} & 1\leq i\leq w, \\ v_j = \frac{q_{j+w}}{1-t} & 1\leq j\leq N-w. \end{cases} \qquad (9)$$

Since $q_w = s$ and $q_{w+1} \leq q_w$, we have $u_w = s/t$ and $v_1 \leq s/(1-t)$. Therefore, from Lemma 1 and Lemma 2 we have $H(\underline{u}) \geq \mathcal{H}_w(w,s/t)$ and $H(\underline{v}) \geq \mathcal{H}_1(N-w,s/(1-t)$. From Eq. 8,

$$\begin{aligned} H(\underline{q}) &= h(t)+tH(\underline{u})+(1-t)H(\underline{v}) \\ &\geq h(t)+t\mathcal{H}_w(w,s/t) \\ &\quad +(1-t)\mathcal{H}_1(N-w,s/(1-t) \\ &= H(\underline{p}). \end{aligned} \qquad (10)$$

This contradicts the assumption. ■

Note that in order to comply with the restriction that $p_i \geq p_{i+1}$, the value of s should be in the range

$$\frac{1-t}{N'-w} \leq s \leq \frac{t}{w}.$$

Theorem 2 (Lower bound): Let $\mathcal{C}$ be a code characterized by $H(C)$ and Q_{mc}, then

$$H(C) \geq \min_{s_l\leq s\leq s_h} \mathcal{H}_{Q_{mc}|\mathcal{C}|}(|\mathcal{C}|,s,Q_{wc}), \qquad (11)$$

where

$$s_l = \frac{1-Q_{wc}}{(1-Q_{mc})|\mathcal{C}|},$$

and

$$s_h = \frac{Q_{wc}}{Q_{mc}|\mathcal{C}|}.$$

The proof of the theorem follows directly from Lemma 3.

Note that for $Q_{wc} = Q_{mc}$ the upper and lower bounds on the entropy meet, and we have $H(C) = \log_2|\mathcal{C}|$.

Example 2: Consider the circuit of Ex. 1. From Theorem 1 the entropy of the code which equals 1.95, is upper bounded by $H(C) \leq 2.21$. Following Theorem 2, we have $w=4, t=15/16, s_l = 1/16$, and $s_h = 15/64$, and the entropy of the code is lower bounded by

$$H(C) \geq \min_{s_l\leq s\leq s_h} \mathcal{H}_4(5,s,15/16) = 1.31.$$

IV. How to use the bounds to choose a code

In the previous section we introduced upper and lower bounds on the worst error masking probability Q_{wc} as a function of the code's entropy and its Q_{mc}. In this section we show that the entropy of the code plays a major role in its design. As before, we assume that the only information available to the code designer is the entropy of the information words.

In general, there are two types of security oriented codes: codes that are assumed to detect *weak attacks* in which the attacker cannot control the codeword to be used, and codes designed to detect *strong attacks* in which the attacker chooses the information word to be transmitted. Robust codes with or without pre-mapping [9], [12], [13] are considered as a countermeasure against weak attacks, and Algebraic Manipulation Detection (AMD) codes [7] are considered to be a countermeasure against strong attacks.

AMD codes are usually considered to be stronger than robust codes since unlike robust codes their error masking probability does not depend on the PMD. In this section we show that when the entropy of the code is smaller than $\log_2|\mathcal{C}|$, strong attack detecting codes are not necessarily stronger than the simple codes. To demonstrate the role of the entropy, we assume that the code rate is fixed and investigate the best way to design the code.

A. A robust code with Pre-mapping

Pre-mapping is an encoding technique that serves to reduce the error masking probability of robust codes[2] [13], [12]. The two robust codes developed to date, the Quadratic-Sum code and the binary Punctured Cubic code, share a common property: the set of information words that mask an (arbitrary) error form a coset of a linear subspace [12]. Pre-mapping takes advantage of this property. It permutes the information words such that the largest subspace contained in the permuted subset of the most probable words is as small as possible. Specifically, let $V \subset \mathcal{C}$ be a set of *high-probability* codewords, i.e., V consists of the codewords which are most likely to occur. Let e_1, e_2, e_3 be errors with $Q_{mc}(e_i) > 0, i = 1, 2, 3$. The masking probability of these errors depends on how they distort the codewords from V. Fig. 5 illustrates the distortion of the codewords of V by e_1, e_2, e_3. Error e_1 is *detected* with high probability, since $\mathcal{C} \cap \{e_1 + V\} = \varnothing$. Error e_2 is *masked* with high probability since $\{e_2 + V\} \subseteq \mathcal{C}$. Error e_3 is detected with variable probability, depending on the cumulative probability of the codewords in $\mathcal{C} \cap \{e_3 + V\}$. The pre-mapping techniques presented in [13], [12] aim to eliminate errors of type e_2 that are masked with a high probability.

What is the worst error masking probability when pre-mapping is implemented? Denote by σ the maximal overlap, i.e. the maximal size of $\mathcal{C} \cap \{e + V\}$ over all the nonzero error vectors when pre-mapping is employed. In the worst case pre-mapping will not reduce Q_{wc}. This can happen only if

$$\sum_{i=1}^{Q_{mc}|\mathcal{C}|} p_i = \sum_{i=1}^{\sigma} p_i + \sum_{i=Q_{mc}|\mathcal{C}|+1}^{2Q_{mc}|\mathcal{C}|-\sigma} p_i.$$

In other words, pre-mapping will not improve the code if

$$p_i = s, \text{ for all } \sigma + 1 \le i \le 2Q_{mc}|\mathcal{C}| - \sigma.$$

Denote by $Q_{wc}^{(map)}$ the worst error masking probability when pre-mapping is used. Recall that we require that

$$p_\sigma \ge p_{\sigma+1} = p_{2Q_{mc}|\mathcal{C}|-\sigma} \ge p_{2Q_{mc}|\mathcal{C}|-\sigma+1} > 0.$$

This implies that s is in the range $[s_l : s_h]$ where

$$s_l = \frac{1 - Q_{wc}^{(map)}}{(1 - Q_{mc})|\mathcal{C}|},$$

and

$$s_h = \min\left(\frac{Q_{wc}^{(map)}}{Q_{mc}|\mathcal{C}|}, \frac{1 - Q_{wc}^{(map)}}{Q_{mc}|\mathcal{C}| - \sigma}\right).$$

From Theorems 1 and 2 we have the following relation between the entropy and the worst error masking probability when pre-mapping is employed.

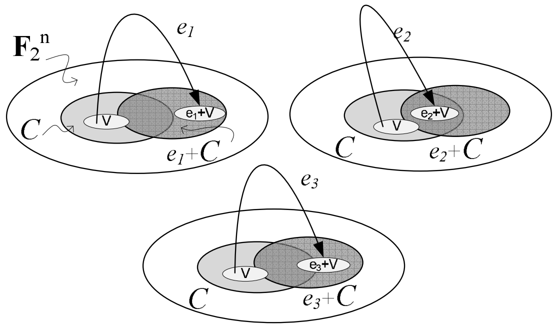

Fig. 5. Distortion of codewords by errors.

Property 2: define a PDM $\underline{q}(s)$

$$q_i(s) = \begin{cases} \frac{t}{\sigma} & 1 \le i \le \sigma \\ s & \sigma < i \le \sigma + w \\ \frac{1-(t+sw)}{|\mathcal{C}|-(w+\sigma)} & \sigma + w < i \le |\mathcal{C}| \end{cases} \qquad (12)$$

where $w = 2(Q_{mc}|\mathcal{C}| - \sigma)$ and $t = Q_{wc}^{(map)} - \frac{sw}{2}$. Then,

$$\begin{aligned} H(C) &\ge \min_{s_l \le s \le s_h} \mathcal{H}_{Q_{mc}|\mathcal{C}|}(|\mathcal{C}|, s, Q_{wc}^{(map)}) \\ H(C) &\le \max_{s_l \le s \le s_h} H(\underline{q}(s)). \end{aligned}$$

B. AMD codes

Another technique that can reduce the worst error masking probability is use of nonlinear codes with randomized embedding. In this method, each information word has multiple images. The AMD codes presented in [7], [16], [17] utilize this property to protect the system from strong attacks. In strong attacks, the attacker can choose both the information word and the injected error. Therefore, the error masking probabilities of codes that can detect strong attacks do not depend on the PMDs of the codes.

A codeword of a (k, m, l) AMD code with $r = m + l$ redundancy bits consists of three parts: a k-bit information word, an m-bit random part, and an l-bit redundancy part. The parameters of an AMD code must satisfy

$$k \le l2^m - m - l$$

otherwise, no AMD code exists [17].

In general, the error masking probability of AMD codes is lower bounded by [17]

$$Q^{(AMD)} = \lceil \frac{k+m}{r-m} \rceil 2^{-m}. \qquad (13)$$

Note that for $k + r \ge 7$, $m = r - 2$ minimizes $Q^{(AMD)}$.

Finding constructions for AMD codes with arbitrary (k, m, l) values is a challenging task. In [17] a family of AMD codes based on a Generalized Reed-Muller (GRM) codes was introduced. The codes are optimal or close to optimal for many k, m and l. In particular, for $k = 2^m l - m - 2l, m = tl$ where t is an integer $(t + 1)l = r$, the error masking probability of the code equals $1 - 2^{-m+1}$. For the special case where $t = 1$ and $k \le \frac{r}{2}(2^{\frac{r}{2}} - 3)$ and $m = l = r/2$, the error masking probability of the codes is greater or equal to $(\frac{2k}{r} + 1)2^{\frac{-r}{2}}$.

[2] Pre-mapping has a drawback in that it produces a non-systematic code. However, applications, such as circuits implementing Finite State Machines (FSMs), can employ non-systematic codes. In the next section we demonstrate the efficiency of pre-mapping on several benchmark FSMs.

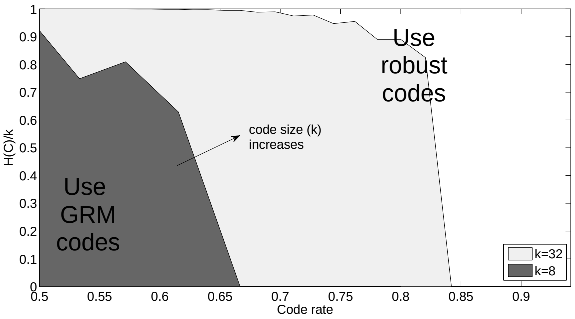

Fig. 6. The partition of the $(code\ rate, normalized\ entropy)$ plane according to the criterion in Eq. 15 for two code sizes, $k = 8$ and $k = 32$.

C. *The role of entropy in code design*

It follows from Theorem 1 that if the entropy of the code is larger than

$$\max_{1 \leq m < r} \log_2 |\mathcal{C}| - D_b(Q_{AMD}||Q_{mc}), \tag{14}$$

it is better to use robust codes such as the QS and PC code (even without pre-mapping); otherwise, it is better to try an AMD code as it *may* provide a smaller error masking probability.

Moreover, denote by $Q^{(GRM)}$ the smallest error masking probability that can be achieved by GRM based codes, and denote by Q_{mc} the error masking probability of a robust code of the same code rate under a uniform distribution of the codewords. It follows from Theorem 1 that,

> *If the entropy of the code is larger than*
>
> $$\log_2 |\mathcal{C}| - D_b(Q^{(GRM)}||Q_{mc}), \tag{15}$$
>
> *then use the robust code (even without pre-mapping), otherwise, use the GRM based code.*

The role of the entropy in code design is illustrated in Fig. 6. The x-axis represents the code rate and the y axis represents the entropy of the code normalized to the code size k. The plane is partitioned according Eq. 15 into two areas: the shaded area represents $(code\ rate, normalized\ entropy)$ pairs for which a strong MAD code may provide smaller error masking probability, and the colorless area represents cases where a simple robust code always provides better immunity.

D. *Robustness of arithmetic modules*

In this section we show how the bounds can be used to design two robust arithmetic modules: an 8-bit multiplier whose output is stored in a $k = 16$ bit register. and a 15-bit adder with a 16-bit output. The entropy of the multiplier with uniformly distributed inputs equals 13.7 and the entropy of the adder equals 15.72.

Assume that the multiplication result is protected by a $\mathcal{C}(n, k = 16)$ Quadratic Sum (QS) code [8]. Recall that a QS code is a robust code of length n, size 2^k, and has $r = n - k$ redundancy bits. For $k = 2sr$, the code is optimum, its Q_{mc} equals 2^{-r}, [8]. Figure 7 shows upper and lower bounds on $H(\mathcal{C})$ for two QS codes of dimension $k = 16$. The X-axis represents the error masking probability, and the Y-axis represents the entropy. The solid and dashed lines represent the upper and lower bounds, respectively. The dashed lines correspond to $r = 8$ and the solid lines to $r = 4$. Note that for uniformly distributed codewords; i.e., when $H(\mathcal{C}) = k = 16$, the upper and lower bound meet. This happens when $Q_{wc} = Q_{mc}$. As the entropy of the code decreases the gap between the bounds increases.

From the figure, in the worst case scenario, for $r = 8$ an attack on the multiplier can be masked with a probability of

$$0.0188 \leq Q_{wc}^{(mul)} \leq 0.39.$$

It is interesting to note that in this case, $\log_2 |\mathcal{C}| - H(C) = 2.3$ *information bits* were ”wasted” for the sake of simple representation of the product (as a number in base 2), and at least $(n - k) - \log_2(0.0188) = 2.26$ *redundancy bits* may be wasted for the same reason. In the worst case scenario, the probability that the code with $r = 8$ will not detect an attack on the adder is

$$0.0045 \leq Q_{wc}^{(add)} \leq 0.08.$$

For a given code rate, which code is better, a robust code with or without pre-mapping or an AMD code? Since a Punctured Cubic (PC) codes with $Q_{mc} = 2^{-r+1}$ exists for any k and $r \leq k$ we use it for comparison[3]. Figures 8 and 9 show the bounds on the worst error masking probability versus the number of redundancy bits. The upper bounds on the error masking probability with and without pre-mapping (Theorem 1 and Prop. 2) are shown in red. The lower bound (Theorem 2) is shown in blue. The *lower bound* on the error masking probability of an AMD code (Eq. 13) and the actual error masking probability of the GRM based codes mentioned above are shown in black.

Note that Eq. 15 suggests a simple way to decide which type of code will provide better immunity to error injection attacks (see the intersection point between the red and black lines in the figures). In fact, it follows from the criterion in Eq. 14 that for both arithmetic modules, if the number of redundancy bits must be less than six, a robust PC code (even without pre-mapping) will provide a smaller error masking probability. Otherwise, an AMD code *may* do better. Moreover, it follows from the criterion in Eq. 15 that a PC code is better than a GRM code for a multiplier with $r \leq 7$ redundancy bits and for an adder with $r \leq 13$.

V. Benchmark Circuits

In this section, we present the efficiency of Punctured-Cubic (PC) codes when used to protect the *combinational part* that generates the next-state of an FSM.

Several benchmark FSMs from the ACM/SIGDA (LGSynth91) package were examined. The FSMs' parameters and the lower and upper bounds on Q_{wc}, denoted by Q_{LB} and Q_{UB}, are summarized in Table I. The 10^{th} column in the table shows the error masking probability Q_{HB} that can be

[3] Note that if r divides k then it is better to use the QS as it has a smaller Q_{mc}.

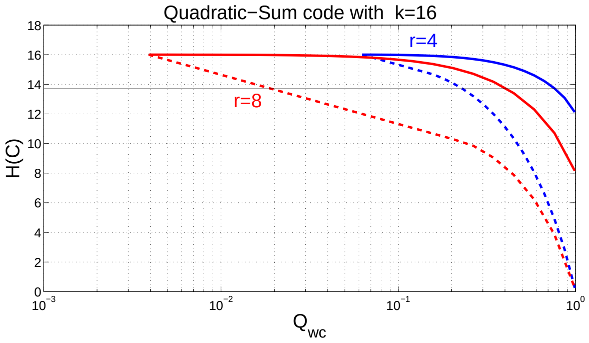

Fig. 7. Upper and lower bounds on the entropy of the Quadratic-Sum code as a function of the worst error masking probability Q_{wc}. The dashed line represents bounds for a Quadratic Sum (QS) code of size 2^{16} and $r = 8$ redundancy bits. The solid line represents bounds for a QS code of size 2^{16} and $r = 4$ redundancy bits.

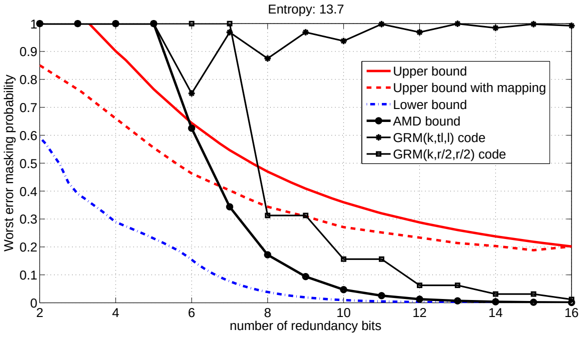

Fig. 8. The worst error masking probability for an 8-bit multiplier: Upper (red) and lower (blue) bounds on the error masking probability of a PC code. Lower bound (black dots) on the error masking probability of an AMD code and error masking probabilities of two GRM based codes (black stars and squares).

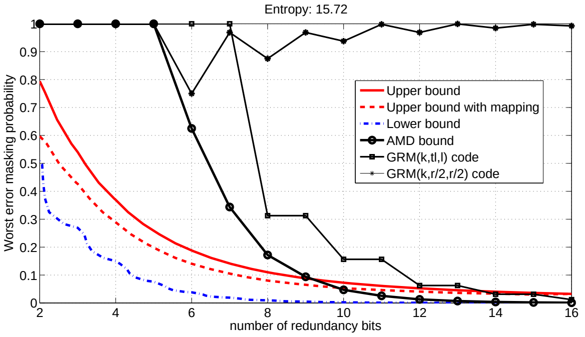

Fig. 9. The worst error masking probability for a 15-bit adder: Upper (red) and lower (blue) bounds on the error masking probability of a PC code. Lower bound (black dots) on the error masking probability of an AMD code and error masking probabilities of two GRM based codes (black stars and squares).

achieved by the Hamming ball state-assignment pre-mapping technique [13].

A graphical presentation of the experimental results is shown in Figure 10. The upper and lower bounds (Q_{LB} and Q_{UB}) define the range of Q_{wc}. The range is marked by a blue rectangle. The worst error masking probability Q_{wc} of the PC code is marked as a black dot. Note that, although the gap between the lower and upper bounds may seem large, the bounds in Theorems 1 and 2 are actually tight; in 'donfile' FSM, the Q_{wc} meets the lower bound and in 'keyb' it meets the upper bound.

The Q_{mc} that characterizes the code is marked by a red dot. Note the significant affect of the PMD on the error masking probability of the code. In almost all of the FSMs the error masking probability under uniform distribution (Q_{mc}) is significantly smaller than Q_{wc}. Hence, it is pointless to design a security-oriented code without taking into account its entropy.

Finally, the figure clearly shows that in most cases it is possible to avoid the worst error masking probability by using pre-mapping such as the Hamming ball state assignment [13]. The black triangles marks the *upper bound* on the *real* error masking probability Q^* when a Hamming ball state assignment is used. Note that the entropy based upper bound on $Q^{(map)}$ (Prop. 2) falls within the blue rectangle that marks the range between Q_{LB} and Q_{UB}. This figure demonstrates that the assumption underlying Prop. 2 is in most cases too pessimistic. In many cases, e.g. in '*tma*' FSM, pre-mapping significantly improves the *real* worst error masking probability (the black triangle is outside the blue rectangle).

VI. Conclusions

In most circuits, the words at the output of the circuit are not uniformly distributed; i.e., they do not have maximum-entropy. This degrades the efficiency of security oriented codes when they are applied to protect the circuits against fault injection attacks. Here we show that there is a relation between the entropy of the outputs and the worst error masking probability of the code; we present upper and lower bounds on this probability, and demonstrate their tightness on standard benchmark circuits. The bounds can help circuit designers to choose a code and determine the number of redundant bits required to provide an acceptable error masking probability.

References

[1] N. Admaty, S. Litsyn and O. Keren, "Punctuating, Expurgating and Expanding the q-ary BCH Based Robust Codes," *The 27-th IEEE Convention of Electrical and Electronics Engineers in Israel*, 2012.

[2] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The sorcerers apprentice guide to fault attacks," 2002.

[3] T. M. Cover and J. A. Thomas, *Elements of information theory*, 1st Edition. New York: Wiley-Interscience, 1991.

[4] S. Engelberg, O. Keren, "A Comment on the Karpovsky-Taubin Code," *IEEE Trans. Info. Theory,* Vol. 57, No. 12, pp. 8007-8010, 2011.

[5] M. G. Karpovsky, P. Nagvajara, "Optimal Codes for the Minimax Criterion on Error Detection," *IEEE Trans. on Information Theory*, Vol. 35, No. 6, pp. 1299-1305, November 1989.

[6] M.G.Karpovsky and A. Taubin, "A New Class of Nonlinear Systematic Error Detecting Codes," *IEEE Trans. Info. Theory,* Vol 50, No.8 pp.1818-1820, 2004.

TABLE I
BENCHMARK FSMS

FSM	$\lvert\mathcal{C}\rvert$	r	$\log_2 \lvert\mathcal{C}\rvert$	$H(C)$	Q_{mc}	Q_{LB}	Q_{wc}	Q_{UB}	Q_{HB}
dk16	27	3	4.75	4.56	0.30	0.34	0.49	0.55	0.34
donfile	24	2	4.58	4.58	0.67	0.67	0.67	0.71	0.50
ex1	20	3	4.32	3.13	0.40	0.68	0.94	0.98	0.51
keyb	19	3	4.25	1.08	0.42	0.92	1.00	1.00	0.89
planet	48	2	5.58	5.19	0.67	0.73	0.91	0.96	0.55
pma	24	3	4.58	3.45	0.33	0.62	0.87	0.93	0.45
s1	20	2	4.32	3.86	0.80	0.85	0.96	1.00	0.63
s1488	48	3	5.58	1.12	0.33	0.92	1.00	1.00	0.96
s1494	48	3	5.58	1.12	0.33	0.92	1.00	1.00	0.96
s1a	20	3	4.32	3.86	0.40	0.55	0.75	0.79	0.43
s208	18	4	4.17	1.08	0.22	0.89	1.00	1.00	0.99
s510	47	3	5.55	5.39	0.34	0.39	0.53	0.57	0.29
s820	25	3	4.64	1.49	0.32	0.87	1.00	1.00	0.86
s832	25	3	4.64	1.49	0.32	0.87	1.00	1.00	0.86
sand	32	2	5.00	4.48	0.50	0.62	0.84	0.90	0.52
scf	121	4	6.92	2.04	0.13	0.83	1.00	1.00	0.89
styr	30	3	4.91	1.83	0.27	0.83	0.98	1.00	0.84
tbk	32	3	5.00	2.42	0.25	0.76	0.90	1.00	0.80
tma	20	3	4.32	3.29	0.40	0.66	0.91	0.95	0.51

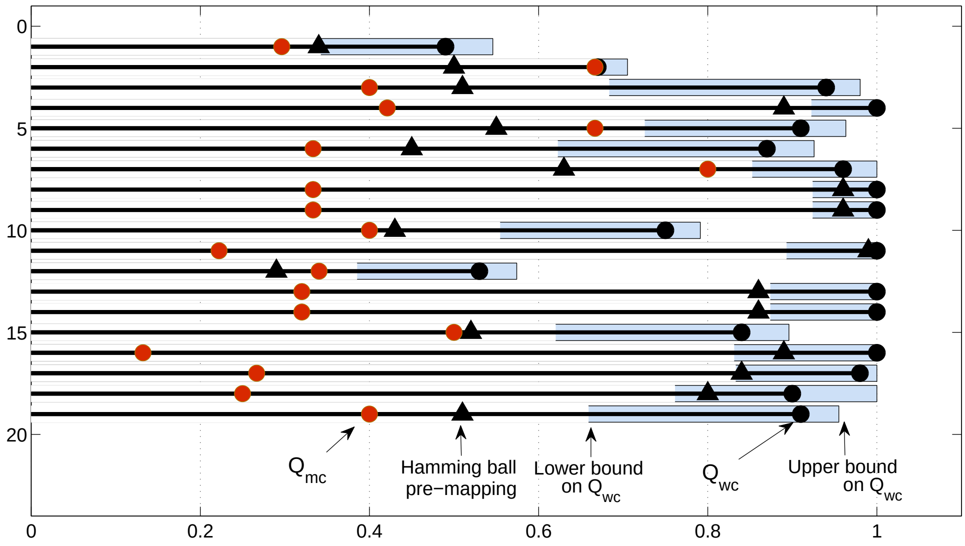

Fig. 10. The error masking probabilities of benchmark FSMs. The blue rectangles mark the range of Q_{wc}. The actual worst case error masking probability and the error masking probability of the code when Hamming ball pre-mapping is used are marked as black dots and black triangles, respectively. The value of the corresponding Q_{mc} is marked as a red dot.

[7] M.G. Karpovsky and Z. Wang, "Design of Strongly Secure Communication and Computation Channels by Nonlinear Error Detecting Codes," *IEEE Trans Computers*, 2014 (to appear).

[8] K. J. Kulikowski, M. G. Karpovsky, A.Taubin, "Robust Codes and Robust, Fault Tolerant Architectures of the Advanced Encryption Standard," *Journal of System Architecture*, vol. 53, pp. 138-149, 2007

[9] Y. Neumeier, O. Keren, "Robust Generalized Punctured Cubic Codes," *IEEE Trans. on Information theory*, Vol. 60, No. 5, pp. 1-10, May 2014.

[10] K. T. Phelps, "A combinatorial construction of perfect codes," *SIAM J. Alg. disc Meth.*, 1983

[11] S. P. Skorobogatov, "Semi-Invasive Attacks a New Approach to Hardware Security Analysis," Technical Report, University of Cambridge. Number 630.

[12] I. Shumsky, O. Keren and M. Karpovsky, "Robustness of Security-Oriented Codes Under Non-Uniform Distribution of Codewords," *Dependable Computing and Communications Symposium at the International Conference on Dependable Systems and Networks, DSN-DCCS*, 2013.

[13] I. Sumsky and O. Keren, "Enhancement of Hardware Security by Hamming Ball Based State Assignment," *Information Security Journal: A Global Perspective. Special issue on Trustworthy Manufacturing and Utilization*, Vol. 22, No. 5-6, pp. 208-215, 2013. Published on-line 2014.

[14] V. Tomashevich, S. Srinivasan, F. Foerg, and I. Polian, "Cross-level Protection of Circuits Against Faults and Malicious Attacks," *IEEE 18th Inter. On-Line Testing Symposium (IOLTS)*, Sitges, pp. 150 - 155, 2012.

[15] I. M.R. Verbauwhede(Ed.), *Secure Integrated Circuits and Systems*, Springer, 2010.

[16] Z. Wang and M.G.Karpovsky, "Algebraic Manipulation Detection Codes and Their Application for Design of Secure Cryptographic Devices," *Proc of Int. Symp. on On-Line Testing*, 2011.

[17] Z.Wang and M.G.Karpovsky, "Reliable and Secure Memories Based on Algebraic Manipulation Correction Codes," *Proc Int Symp. on On-line Testing*, June 2012.

[18] Z. Wang, Mark G. Karpovsky, Konrad J. Kulikowski, "Replacing Linear Hamming Codes by Robust Nonlinear Codes Results in a Reliability Improvement of Memories," *Proc. Int. Symp. Dependable Computing*, July 2009.

[19] Z. Wang, M. G. Karpovsky, K. Kulikowski, "Design of Memories with Concurrent Error Detection and Correction by Non-Linear SEC-DED Codes," *Journal of Electronic Testing*, vol. 26, Oct 2010.