ISIT'2013 1569716973

Multiply Constant Weight Codes

Zouha Cherif[1,2], Jean-Luc Danger[1,3], Sylvain Guilley[1,3], Jon-Lark Kim[4], Patrick Solé[1,5]

[1]Institut MINES-TELECOM, TELECOM ParisTech, CNRS LTCI, 46 rue Barrault, 75 634 Paris, FRANCE.
[2] Université de Lyon, CNRS, UMR5516, Laboratoire Hubert Curien 42 000, Saint-Étienne, FRANCE.
[3] Secure-IC S.A.S., 80 avenue des Buttes de Coësmes, 35 700 Rennes, FRANCE.
[4] Sogang University, Department of Mathematics, Sogang University Seoul 121-742, SOUTH KOREA.
[5] King Abdulaziz University, Department of Mathematics, Jeddah, SAUDI ARABIA.

⟨{zouha.cherif,jean-luc.danger,sylvain.guilley,patrick.sole}@telecom-paristech.fr⟩
⟨jlkim@sogang.ac.kr⟩

***Abstract*—The function $M(m,n,d,w)$, the largest size of an unrestricted binary code made of m by n arrays, with constant row weight w, and minimum distance d is introduced and compared to the classical functions of combinatorial coding theory $A_q(n,d)$ and $A(n,d,w)$. The analogues for systematic codes of $A(n,d)$ and $A(n,d,w)$ are introduced apparently for the first time. An application to the security of embedded systems is given: these codes happen to be efficient challenges for physically unclonable functions.**

Keywords: constant weight code, doubly constant weight codes, multiply constant weight codes, PUFs.

I. INTRODUCTION

The function $A(n,d,w)$, *i.e.* the largest size of a binary code of minimal distance d and constant weight w, was introduced in the sixties as a tool to bound above $A(n,d)$ the largest size of a binary code of length n and minimal distance d. It has interest in its own right, with engineering and mathematical applications. One way to generalize it is to consider so-called doubly constant weight codes whose support is partitioned into two parts of size n_1 and n_2 with $n_1+n_2=n$, and where codewords must have weight w_1 (resp. w_2) in the first (resp. second) part [5], [11]. In the present paper, we consider a generalization where the support is partitioned into m parts and where codewords must have weight w in each part. We call these codes *multiply constant weight codes*. Our motivation comes from security of embedded systems and is described in the following section. We consider constructions and bounds. The construction comes from concatenation and product code techniques. The product code necessitating one of the two codes to be constant weight hence non linear has to be generalized at the level of systematic codes. This motivates us to study systematic constant weight codes, already studied in [3], [9], and to introduce the function $S(n,d,w)$, the analogue of $A(n,d,w)$ for systematic constant weight codes. A rather different type of constructions comes from design theory especially designs admitting resolution of their blocks into parallel classes. For upper bounds we mention the immediate bound obtained from the fact that our codes are constant weight codes of length nm and weight mw. Eventually we consider asymptotic versions of the preceding bounds. This study shows that concatenation is better at low rates and product codes at high rates, the transition occurring at relative distance $\delta=\delta_0\approx 0.0417$.

The rest of this article is structured as follows. Section II contains the engineering motivation, from the field of physically unclonable functions. Section III collects the necessary definitions and notations. Section IV deals with constructions and attached lower bounds. Section V contains the upper bound. Section VI studies asymptotic versions of the bounds of Section IV. Finally, conclusions and perspectives are in Section VII.

II. MOTIVATION

The need of the code $M(m,n,d,w)$ comes from the generation of some type of Physically Unclonable Functions (PUFs) in trusted electronic circuits. The PUF aims at giving a unique signature to the device without the need for the user to program an internal memory [12]. This signature allows the user to build lightweight authentication protocol or even protect a master key in cryptographic implementations. Such key can be used for standard cryptographic protocols, or for internal cryptography (*e.g.* memory encryption). The PUF takes advantage of technological process variations to differentiate between two devices. For instance imagine a PUF composed of two delay lines with the same structure; the propagation time through these two delay lines should be theoretically the same. However, the measurement gives two different times as there is a slight difference due to the imbalance between the physical elements. The delay lines are generally controllable in order to carry out comparisons between two different paths. The control word is called a challenge. For every challenge word the PUF responds with a specific

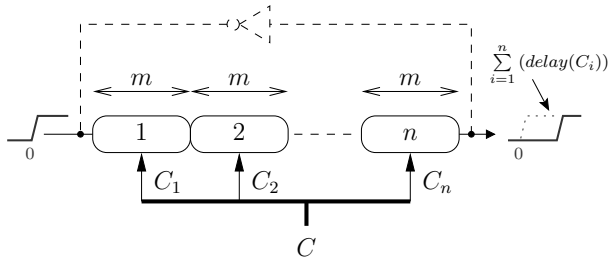

Fig. 1. Loop PUF principle

response. Hence the set of pairs (Challenge, Response) corresponds to the signature of the circuit.

The Loop PUF [4] is a set of n identical delay lines with m controlled delay elements. Each delay element can have two delays $delay(0)$ and $delay(1)$; the delay is chosen according to a control bit from the challenge word C. A challenge word C is a set of n sub-control words C_i, each sub-control word having m bits C_i^j associated to a delay element, $i \in [1, n], j \in [1, m]$. In the sequel we consider $m = 1$ to simplify the explanations. The Figure 1 illustrates the Loop PUF structure where the delay of the line controlled by the challenge C is measured.

The Loop PUF response ID is related to the comparison between the global delays (the sum of elementary delays of the n delay lines) D and D' measured when applying two different challenges C and C', respectively. If we consider n delay lines and one delay element per line ($m = 1$), then,

$$ID = Sign(D - D'), \text{ with}$$
$$D - D' = \sum_{i=1}^{n} (delay(C_i) - delay(C_i')) \quad , \qquad (1)$$

where $delay(C_i)$ is the delay (time) of the element of the delay line $i \in [1, n]$ controlled by the control bit C_i. As the delay measurement requires a great accuracy, the delay chain is looped to form a ring oscillator by means of an inverter, as shown in Figure 1. Thus the delay is obtained by inverting the measured frequency of the ring oscillator.

To increase the reliability of the PUF response and since the PUF is very sensitive to the environmental noise, it is necessary to choose challenges which offer the greater difference between their delays ($|D - D'| \gg 0$). Indeed if C_i and C_i' are not equal, the number of impacting bits is the *Hamming Distance* between C and C'.

Then,

$$D - D' = \sum_{C_i \neq C_i'} (delay(C_i) - delay(C_i')) \ . \qquad (2)$$

We can assume that the higher the $HammingDistance(C, C')$, the better is the reliability of the PUF response. Other condition for the choice of the challenges is that all applied control words must have the same Hamming weight [4].

III. Definitions and Notations

The largest size of a $q-$ary code of length n and minimum distance d is denoted by $A_q(n, d)$. When $q = 2$ it is denoted simply by $A(n, d)$. The largest size of a *linear* binary code of length n and minimum distance d is denoted by $B(n, d)$. Thus $B(n, d) \leq A(n, d)$.

The largest size of a binary code of constant weight w and minimum distance d is denoted by $A(n, d, w)$.

The analogous functions to $A(n, d)$, $A(n, d, w)$ for **systematic**, possibly nonlinear codes are denoted respectively by $S(n, d) = 2^{s(n,d)}$ and $S(n, d, w) = 2^{s(n,d,w)}$. Recall that a code C of 2^k elements is *systematic* if there is a subset I of size k of coordinate positions (the information set) such that 2^I is mapped bijectively to C. In particular every linear codes are systematic. Thus $A(n, d) \geq S(n, d) \geq B(n, d)$. The largest size of a binary code consisting of m by n arrays, with constant row weight w, and minimum distance d is denoted by $M(m, n, d, w)$. Thus $M(1, n, d, w) = A(n, d, w)$, and $M(2, n, d, w) = T(w, n, w, n, d)$, the size of an optimal so-called *doubly constant weight* code studied in [5]. Online tables of bounds on the latter function can be found at [1].

IV. Lower bounds

A. Coding constructions

We can state the following bound, based on **concatenation.**

Proposition IV.1. *Let q denote the largest integer $\leq A(n, D, w)$. We have*

$$M(m, n, \Delta D, w) \geq A_q(m, \Delta).$$

Proof. Consider a concatenation scheme [11, p.307] where the outer $q-$ary code achieves $A_q(m, \Delta)$ and the inner code is a subcode of size q of a constant weight code achieving $A(n, D, w)$. ■

This bound can sometimes be improved by using **product codes.** Recall that if C and D are two binary linear codes then their product $C \otimes D$ is the code of length nm consisting of m by n arrays whose rows belong to C and columns belong to D. If C and D have parameters $[m, k, d]$ and $[n, l, e]$ then the code $C \otimes D$ has parameters $[nm, kl, de]$ [11]. More generally the definition and preceding property remains when linear codes are replaced by systematic codes. For instance let $n = 4$ and $w = 2$. The repetition code of length $m = 6$

over $\mathbb{F}_4$ yields a lower bound of 4 on $M(6,4,12,2)$ (note that $A(4,2,2)=6$ is met by the code consisting of all the vectors of weight 2) when the product of the said constant weight code with the *binary* repetition code of length 6 yields a lower bound of 6.

Proposition IV.2. *We have*

$$M(m,n,\Delta D,w) \geq 2^{s(n,\Delta,w)s(m,D)} \geq B(m,D)^{s(n,\Delta,w)}.$$

Proof. Consider the product code where the first binary code achieves $S(n,\Delta,w)$ and the second binary code achieves $S(m,D)$. Note that, by definition, $S(m,D) \geq B(m,D)$. ■

Consider the following systematic constant weight code $\{0011, 1001, 0110, 1100\}$. Taking the product code with a $[6,2,4]$ yields a lower bound of $2^4 = 16$ on $M(6,4,12,2)$. For future use we give the following bound on $S(n,d,w)$.

We give a simple but robust construction technique for systematic constant weight codes. According to [9] this appears already in [3].

Proposition IV.3. *We have*

$$S(2n,2d,n) \geq S(n,d) \geq B(n,d).$$

Proof. Let C denote a systematic code achieving $S(n,d)$. Construct a constant weight code by the rule

$$D = \{(x,\overline{x}) |\, x \in C\},$$

where the bar denotes complementation. The code D is systematic because C is and has the required parameters. ■

Example: A well-known application of the Plotkin bound is the fact that $B(2^{m-1},2^{m-2}) = 2^m$ is achieved by the Reed Muller code $RM(1,m-1)$ [11].This yields $S(2^m,2^{m-1},2^{m-1}) \geq 2^m$.

Another construction consists in extending a version of the preceding by a suitable constant weight code.

Proposition IV.4. *If* $2^k \leq A(n,d,w)$ *we have*

$$s(n+2k,d+2,w+k) \geq k.$$

Proof. Let C denote a constant weight code achieving $A(n,d,w)$. Let ϕ denote an injection of $\mathbb{F}_2^k$ into C. Let

$$D = \{(x,\overline{x},\phi(x)) |\, x \in \mathbb{F}_2^k\},$$

where the bar denotes complementation. The code D is systematic with information set the first k coordinates and has the required parameters. ■

Example: Consider the minimum weight codewords in the $[7,4,3]$ Hamming code. They achieve $A(7,4,3) = 7$. Taking a subcode of size 4 yields $s(11,5,5) \geq 2$.

B. Designs constructions

Let X be a $2-(v,k,1)$ design. Assume a parallelism resolution into $\frac{v-1}{k-1}$ parallel classes of $\frac{v}{k}$ block each. To each class we attach an array. Thus by definition of block parameters $n = v$, $w = k$ and $m = v/k$. The weight of each array is mk. Each pair of arrays intersect in at most m places because each pair of blocks intersect in at most one place. Thus one may take $d = 2m(k-1)$. This construction thus gives a lower bound on $M(v/k, v, 2v(k-1)/k, k)$ of $\frac{v-1}{k-1}$. Possible parameters are $k = 3$ for all $v = 3 \pmod 6$ and $k = 4$ for all $v = 4 \pmod{12}$ [6].

V. Upper bounds

The following upper bound is immediate from the definitions.

Proposition V.1. *We have*

$$M(m,n,d,w) \leq A(nm,d,mw).$$

In what follows, we give a recursive upper bound for $M(m,n,d,w)$.

Proposition V.2. *We have*

(i) $M(m,n,d,w) \leq \left[\frac{n^m}{w^m} M(m,n-1,d,w-1)\right]$

(ii) $M(m,n,d,w) \leq \left[\frac{n^m}{(n-w)^m} M(m,n-1,d,w)\right]$

(iii) $M(m,n,d,w) \leq \left[\frac{u}{(m\times w^2)/n-\lambda}\right]$, *where* $d = 2u$ *and* $\lambda = w-u$.

Proof. The following are known in [8].

(i) $T(w,n,w,n,d) \leq \left[\frac{n}{w} T(w-1,n-1,w,n,d)\right]$
(ii) $T(w,n,w,n,d) \leq \left[\frac{n}{w} T(w,n,w-1,n-1,d)\right]$
(iii) $T(w,n,w,n,d) \leq \left[\frac{n}{n-w} T(w,n-1,w,n,d)\right]$
(iv) $T(w,n,w,n,d) \leq \left[\frac{n}{n-w} T(w,n,w,n-1,d)\right]$
(v) $T(w,n,w,n,d) \leq \left[\frac{u}{(2\times w^2)/n-\lambda}\right]$

Applying (i) and (ii), we get $M(2,n,d,w) \leq \left[\frac{n^2}{w^2} M(m,n-1,d,w-1)\right]$. Similarly, applying (iii), (iv), and (v), we get $M(2,n,d,w) \leq \left[\frac{n^2}{(n-w)^2} M(m,n-1,d,w)\right]$ and $M(2,n,d,w) \leq \left[\frac{u}{(2\times w^2)/n-\lambda}\right]$. Because the above inequalities can be extended to $T(w,n,w,n,w,n,\cdots,w,n,d) = M(m,n,d,w)$ in an obvious way, we obtain the proposition. ■

By applying (i) of Proposition V.2 repeatedly, we have the following.

Corollary V.3.

$$M(m,n,d,w) \leq \left[\frac{\prod_{i=0}^{w-2}(n-i)^m}{\prod_{i=0}^{w-2}(w-i)^m} M(m,n-(w-1),d,1)\right] =$$

$$\left[\frac{n^m \cdots (n-(w-2))^m}{w^m \cdots 2^m} M(m, n-(w-1), d, 1)\right],$$

where $M(m, n-(w-1), d, 1) \leq A(m \times (n-(w-1)), d, m-(w-1))$.

VI. Asymptotics

We consider the asymptotic process where n, m are large and $d \sim \delta nm$, and $w \sim \omega n$, with $\delta, \omega \in (0,1)$. The exponents for the functions $A_q(n,d)$ and $A(n,d,w)$ are denoted by $R_q()$ and $\alpha(,)$, respectively, with q dropped if $q=2$. The exponents for the functions $S(n,d)$ and $S(n,d,w)$ are denoted by $\Sigma()$ and $\sigma(,)$, respectively. The exponent for $M(m,n,d,w)$ is denoted by $\mu(\delta,\omega)$. Thus, up to subexponential factors we have

- $A_q(n,d) \sim q^{nR_q(\delta_1)}$, if $d \sim \delta_1 n$
- $A(n,d) \sim 2^{nR(\delta_1)}$, if $d \sim \delta_1 n$
- $A(n,d,w) \sim 2^{n\alpha(\delta_2,\omega_2)}$, if $d \sim \delta_2 n$ and $w \sim \omega_2 n$
- $S(n,d) \sim 2^{n\Sigma(\delta_1)}$, if $d \sim \delta_1 n$
- $S(n,d,w) \sim 2^{n\sigma(\delta_2,\omega_2)}$, if $d \sim \delta_2 n$ and $w \sim \omega_2 n$
- $M(m,n,d,w) \sim 2^{nm\mu(\delta,\omega)}$.

We will need the binary entropy function defined for $x \in (0,1)$ by the formula

$$H(x) = -x\log x - (1-x)\log(1-x),$$

where log's are to the base 2. The choice of $\omega = 1/2$ in applications is justified by the following result.

Proposition VI.1. *We have*

$$\mu(\delta,\omega) \leq \mu(\delta,1/2) = R(\delta).$$

Proof. Trivially $\mu(\delta,\omega) \leq R(\delta)$. To prove that $\mu(\delta,1/2) = R(\delta)$, we apply a version of Bassalygo Elias trick [10, (2.7)] with B the set of all vectors of length nm and weight w on slots of length m, to get

$$A(nm,d) \leq \frac{2^{nm}}{\binom{n}{w}^m} M(m,n,d,w).$$

Passing to the limit on n, m and taking logs yields

$$R(\delta) \leq 1 - H(\omega) + \mu(\delta,\omega),$$

and letting $\omega = 1/2$ yields $\mu(\delta,1/2) \geq R(\delta)$. The result follows. ■

The asymptotic version of Proposition V.1 is then

Proposition VI.2. *We have*

$$\mu(\delta,\omega) \leq \alpha(\delta,\omega).$$

The proof is immediate and omitted. The next result is the best upper bound we know.

Corollary VI.3. *We have* $\mu(\delta,\omega) \leq g(u^2)$, *with* $g(x) = H((1-\sqrt{1-x})/2)$, *and*

$$u = -\delta + \sqrt{\delta^2 - 2\delta + 4\omega(1-\omega)}.$$

In particular

$$\mu(\delta,1/2) \leq H(1/2 - \sqrt{\delta(1-\delta)}.$$

Proof. Follows from Proposition VI.2 by [10, (2.16)]. Note that the case $\omega = 1/2$ is immediate from $\mu(\delta,1/2) = R(\delta)$ by Proposition VI.1 combined with [10, (1.5)]. ■

We are now in a position to give the main constructive lower bound of this section, based on an asymptotic version of Proposition IV.1.

Theorem VI.4. *We have*

$$\frac{32\mu(\delta,1/2)}{3} + \delta \geq \frac{3}{7}.$$

Proof. Consider a family of geometric codes of rate R_q and relative distance δ_q over the field $\mathbb{F}_q$, for q a square. Assume these parameters are above the Tsfasman-Vladut-Zink (TVZ) line [7, Theorem(13.5.4)]

$$R_q + \delta_q \geq 1 - \frac{1}{\sqrt{q}-1}.$$

We now take $q = 64 \leq A(64,32,32)$ by considering a subcode of the Reed Muller code $RM(1,6)$. We apply the concatenation scheme of Proposition IV.1 yielding

$$M(m,64,32\Delta,32) \geq A_{64}(m,\Delta)$$

with $n\mu = R_q \log(q)$, and $\delta = \lim_{m\to\infty} \frac{32\Delta}{64m} = \delta_q/2$. Specializing to $q=64$ and substituting into the above bound, the result follows. ■

This bound is a straight line cutting the $\mu-$axis in $\mu = 9/224$, and the $\delta-$axis in $\delta = 3/7$. The asymptotic version of Proposition IV.2 is then

Proposition VI.5. *We have*

$$\mu(\delta,\omega) \geq \sigma(\delta_1,\omega)\Sigma(\delta_2),$$

where δ_1, δ_2 *are arbitrary reals in* $(0,1)$ *satisfying* $\delta = \delta_1\delta_2$.

We are now in a position to give the main nonconstructive lower bound of this section.

Theorem VI.6. *We have for* $\delta \leq 1/4$, *the bound*

$$\mu(\delta,1/2) \geq (1 - H(\sqrt{\delta}))^2/2.$$

Proof. By applying Varshamov-Gilbert (VG) [11] to systematic codes (in fact linear codes) we get

$$\Sigma(\delta_2) \geq 1 - H(\delta_2).$$

Combining VG for linear codes with Proposition IV.3 we get

$$\sigma(\delta_1,1/2) \geq (1 - H(\delta_1))/2.$$

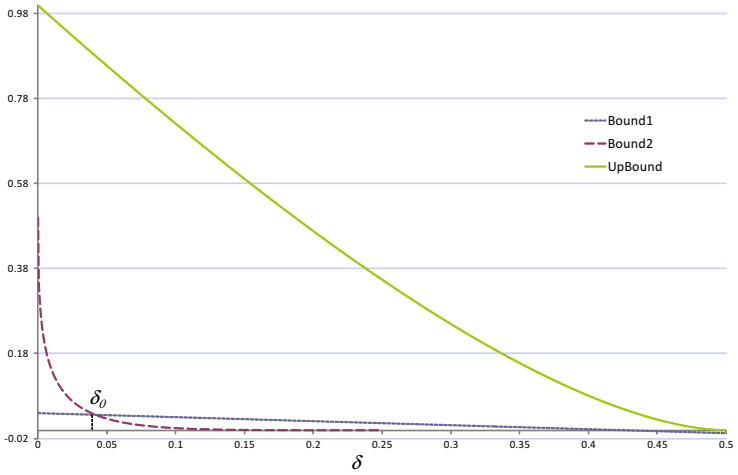

Fig. 2. Upper and lower bounds

Using Proposition VI.6 with $\delta_1 = \delta_2 = \sqrt{\delta}$, the result follows. ■

Figure 2 illustrates the best upper bound (**UpBound**) and the two lower bounds. Let us define the first (second) lower bound obtained from Theorem VI.4 (Theorem VI.6) by **Bound1**(**Bound2**). We observe that Bound2 is better than Bound1 for $\delta \leq \delta_0$ and the opposite is true for $\delta > \delta_0$, with $\delta_0 \approx 0.0417$.

VII. Conclusion and open problems

In this work, motivated by the security of embarked systems, we have introduced and studied multiply constant weight codes, a generalization of doubly constant weight codes. The main construction techniques are concatenation and product codes. They provide lower bounds on the new combinatorial function $M(m, n, d, w)$. Asymptotics show that concatenation is better at low rate and product codes better at high rate. For upper bounds we rely on the fact that our multiply constant weight codes are, in particular, constant weight codes. Non asymptotic upper bounds include an analogue of the Johnson bound. Generalizing the classical upper bounds on constant weight codes in our setting looks feasible. Eventually tabulating the various upper and lower bounds for modest values of the four parameters is a worthwhile project. In particular it would be very interesting to compare the coding constructions to the constructions arising from combinatorial designs. The new function $S(n, d, w)$ is also worth tabulating and has other applications [3], [9].

Acknowledgements: P. Solé is indebted to Professor Hoang Dau Son for pointing out the references [3], [9]. J.-L. Kim would like to mention that this work was supported by the Sogang University Research Grant of 201210058.01. Z. Chérif, S. Guilley and J.-L. Danger would like to mention that this work was supported by Orange Labs and the ENIAC European project 2010-1 "TOISE" (Trusted Computing for European Embedded Systems).

References

[1] http://webfiles.portal.chalmers.se/s2/research/kit/bounds/#1
[2] E. Agrell, A. Vardy, and K. Zeger, "Upper bounds for constant-weight codes," IEEE Transactions on Information Theory, vol. 46, no. 7, pp. 2373-2395, Nov. 2000.
[3] F. H. Binck, H-C. A. van Tilborg : Constructions and bounds for systematic tEC/AUED codes. IEEE Transactions on Information Theory (TIT) 36(6):(1990) 1381–1390.
[4] Z. Cherif, J-L. Danger, S. Guilley and L. Bossuet, "An Easy to Design Loop PUF", DSD (15th Euromicro Conference on Digital System Design), IEEE, pp. 156-162, Çeşme, Izmir, Turkey, Sept 5-8, 2012.
[5] T. Etzion, Optimal doubly constant weight codes, J. of Combin. Designs **16** (2007) 137–151.
[6] H. Hanani; D.K. Ray-Chaudhuri; R-M. Wilson. On resolvable designs. Discrete Math. 3 (1972), 343357.
[7] W.C. Huffman, V. Pless, *Fundamentals of error-correcting codes.* Cambridge University Press, Cambridge, 2003.
[8] S. Johnson, Upper bounds for constant weight error correcting codes, Discrete Math., 3 (1972) 109-124.
[9] M-C. Lin, Constant Weight Codes for Correcting Symmetric Errors and Detecting Unidirectional Errors, IEEE Transactions on Computers, Vol.42, No.11, pp.1294-1302, Nov. 1993.
[10] R. McEliece; E. Rodemich; H. Rumsey; L. Welch, New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities IEEE Transactions on Information Theory, (1977) 157 – 166.
[11] F.J. MacWilliams, N.J.A. Sloane, *The theory of Error Correcting Codes.* North Holland (1981).
[12] R. Pappu, B. Recht, J. Taylor and N. Gershenfeld, "Physical One-Way Functions", Science, 2002, September 20, vol. 297, number 5589, pages 2026–2030; DOI: DOI: 10.1126/science.1074376.