



# Integer Codes in Coding and Computing

Ulrich Tamm

**Abstract**—A single checksum for codes consisting of  $n$  integer components is investigated. In coding theory this is mostly used for single error –correction in unconventional error models. If the errors are such that a single component  $c_i$  is distorted to  $c_i \pm e_i$ , the analysis leads to equivalent group factorizations. We shall present several code constructions for this model, give a short survey on the coding theoretical and mathematical background, and also emphasize applications in cryptography and computer science.

**Index Terms**—single – error correction, perfect codes, group factorization, steganography, distributed computing

## I. INTRODUCTION

We are considering the following checksum for a code consists of all words  $(c_1, \dots, c_n) \in Z$  fulfilling

$$\sum_{i=1}^n w_i \cdot c_i = 0 \pmod{m}, \quad (1)$$

where  $(w_1, \dots, w_n) \in Z$  is a fixed sequence of weights and  $n$  is the length of the code.

Since the checksum is reduced modulo  $m$ , of course, the components of the code words may be regarded as letters over an alphabet of size  $m$ . Usually, this is indeed the case. However, for some applications, the letters  $c_i$  are from a much smaller alphabet.

For instance, in the famous Varshamov – Tenengolts codes [33], which arise for  $(w_1, \dots, w_n) = (1, 2, \dots, n)$ , and  $m = n + 1$ , the code words are binary. Another case, where the  $c_i$ 's are chosen from a smaller alphabet will be discussed more detailed in the application in steganography.

Varshamov – Tenengolts codes are able to correct single asymmetric errors [5] and were later also applied by Levenshtein [17] in order to correct single deletions.

Later, Levenshtein and Vinck [18] and Martirosian [20] used the checksum (1) in order to analyze single – error correcting codes for further unconventional error models as peak shifts in run–length–limited codes. The effect of a single error is reflected in the behaviour of the syndrome, which should be changed to a value different from 0 by a linear combination of the weights corresponding to the codeword's coordinates involved in this error.

The proper choice of the weight sequence is crucial for the quality of the code. It strongly depends on the type of error to be corrected.

Vinck and Morita [34] later called the codes obtained via checksum (1) integer codes.

U. Tamm is with the Department of Business Informatics, Marmara University Istanbul, Turkey and with the Department of Mathematics, University of Bielefeld, Germany, (e-mail: tamm@iee.org)

The most important case is the error type of substitution of the letter  $c_i$  by  $c'_i$ . Then the resulting syndrome is

$$\begin{aligned} w_1 c_1 + \dots + w_{i-1} c_{i-1} + w_i c'_i + w_{i+1} c_{i+1} + \dots + w_n c_n \\ = d + w_i (c'_i - c_i), \text{ for } i = 1, \dots, n \end{aligned} \quad (2)$$

A similar syndrome (with  $n$  being the number of runs in a run – length limited sequence) occurs in the correction of peak shifts discussed by Levenshtein and Vinck [18].

In order to be able to correct one single error, the syndromes of an integer code have to be pairwise different. So if the possible distortions, which can be corrected by the integer code, are from an error set  $\mathcal{E} \subset Z_m$  and the linear combinations of the weights (for instance  $w_i$  for substitutions (2) or  $w_i - w_{i+1}$  for permutations) are from a set  $\mathcal{H} \subset Z_m$ , then we have to assure that

$$e \cdot h \neq e' \cdot h' \text{ for all } e, e' \in \mathcal{E} \text{ and } h, h' \in \mathcal{H}. \quad (3)$$

If, in addition, all elements from the set  $Z_m \setminus \{0\}$  occur as a product in (3), then the code is said to be perfect. For a perfect integer code in  $Z_m$  the pair  $(\mathcal{E}, \mathcal{H})$  is also known as splitting of the additive group  $Z_m$ , cf. [27] and we shall also use this notion in the following.

Usually, we choose  $m = p$  a prime number, such that we operate in finite fields. Then  $\mathcal{E} \cdot \mathcal{H}$  yields a factorization of the multiplicative group  $Z_p^*$  (here multiplication of two sets means the set of all possible products of one element in one set with an element of the other set). For the theory of group factorizations we refer to [30], [24].

Integer codes for the error sets  $\mathcal{E} = \{\pm 1, \pm 2, \dots, \pm k\}$  are denoted as  $k$  – shift codes or  $k$  – shift designs and arise in the study of peak shift correction [18] and of correction of errors in the so – called Stein sphere [7], where a single component is distorted in such a way that the received letter  $c'_i$  is of the form  $c'_i = c_i + j, j \in \{\pm 1, \pm 2, \dots, \pm k\}$ . Conditions for the existence of perfect  $k$  – shift codes have been introduced for  $k = 1, 2$  and  $k = \frac{m-1}{2}$  in [18] and for the parameters  $k = 3$  and  $k = 4$  in [22], [31].

In [21] the error set  $\mathcal{E} = \{\pm 1, \pm a\}$  is discussed. This corresponds to the error model, in which a letter  $c_i$  is changed to one of its nearest neighbours on the  $a \times a$  – grid, where a component  $(x, y)$  is represented by the number  $x + y \cdot a$ . This can be described in such a way that the received letter is contained in the set  $\{c_i \pm 1, c_i \pm a\}$ .

The error set  $\mathcal{E} = \{\pm 1, \pm a, \pm b\}$  was studied in [31] as a special case of the more general  $\{\pm 1, \pm a, \dots, \pm a^r, \pm b, \dots, \pm b^s\}$  for positive integers  $r, s$ .

In Section 2 we shall discuss further applications of the checksum (1) in cryptography and computer science.

In Section 3 we concentrate on error – correction. First, we present a very general method to obtain perfect integer codes for any error set of the form  $\mathcal{E} = \{\pm 1, \pm a_1, \dots, \pm a_{k-1}\}$  in  $Z_p$ , where  $p = 2k + 1$  is an odd prime number. Using this condition, in perfect codes for some special error sets  $\mathcal{E}$  had been derived, for instance in [32].

In this paper, we relax the model not further requiring perfectness but a close packing. This may be even a harder task, since the error spheres around a code word are rather nasty. Some constructions will be provided.

## II. APPLICATIONS IN COMPUTER SCIENCE AND CRYPTOGRAPHY

**Lattice tilings:** Mathematically, a group factorization obtained from a perfect code with error set  $F(k) = \{\pm 1, \pm 2, \dots, \pm k\}$  corresponds to a tiling of the Euclidean space by a certain star body, the  $(n, k)$ -cross. This is a collection of unit  $n$ -dimensional cubes, with one cube in the center and a number of  $k$  consecutive cubes attached to each of its faces.

More exactly, a lattice tiling of the  $n$ -dimensional Euclidean space exists, if  $F(k)$  "splits" some abelian group, which for the groups  $Z_p$  is just a factorization. We do not go into detail here and refer to [27], [29] for further reading.

**Distributed computing:** In parallel computing processors may share some resources as memory. This is usually modeled by a graph, where the vertices denote the processors, and an edge between two vertices means that the processors are connected in the network. Some resources, as memories, software modules, or I/O-connections may be expensive and hence only be placed at a subset of the processors [4], [14]. An efficient placement of these resources leads to the concepts of codes in graphs and domination in graphs, e.g., [3], [15]. Usually, a combinatorial treatment is necessary. However, if the underlying graphs have a very regular structure, as a ring or a grid, then checksum (1) can be used to construct an efficient placement. This is essentially equivalent to a good code, where the code words correspond to the processors equipped with the resources and the error spheres around them correspond to the direct neighbours (or neighbours within a certain distance).

**Packet loss in internet protocols:** Sloane [25] used Varshamov – Tenengolts codes in order to protect internet protocols against packet losses or genome sequences against a deletion of one letter in the sequence. He also analyzed, in which cases it might be better to compute the checksum (1) modulo a number  $d$  different from 0.

**SEC–DED Codes:** If we choose all weights  $w_i = 1$ , the checksum (1) can, of course, be used to detect a single error, which would result in sum different from 0. However, it can not be recognized from which component  $i$  this error results. Choosing instead the weights  $i$  indeed gives information about the location of the component if the alphabet size  $m$

is appropriately large and hence also automatically allows to correct this error. A different way to achieve this goal is a second checksum. If this checksum is carefully chosen, it may also detect a second error. Such SEC–DED codes (single–error correction, double–error detection) are implemented in computer memories, where the errors occur in a single bit. Combining  $l$  bits to an integer modulo  $m = 2^l$ , integer codes with two checksums have been constructed as SEC–DED codes.

**Steganography:** In steganography we have a situation complementary to coding theory, where the errors occur at random. Here, sender and receiver agree on a certain set of  $n$  positions in which the sender may slightly change the value of the component  $c_i$ . In order that these changes will not be detected, there have to be very few changes of very small amplitude. For instance, it may only be allowed to change one component by adding plus or minus 1 to  $c_i$ , i.e.  $c'_i = c_i \pm 1$ . The checksum

$$\sum_{i=1}^n i \cdot c'_i \pmod{2n+1}$$

will then be used to decode the corresponding message. Lisonek's idea in [19] was that by appropriate choice of the weights in checksum (1) it is possible to obtain a better performance by changing two components by  $\pm 1$  and decode the message. In order to assure a successful decoding, Lisonek chose the weights  $w_1, w_2, \dots, w_m$  from a symmetric sum cover  $S = \{0, \pm w_1, \pm w_2, \dots, \pm w_n\}$ , which means that  $S + S = Z_m$ . This has the effect that the message can be decoded, if all the sums  $w_i + w_j$  with  $i \neq j$  are different.

In order to avoid too much overlap, the task hence is to find a large enough  $m$  such that  $S + S = Z_m$ . Lisonek provides a table for small  $m$ . Note that it is really required that  $i \neq j$ . If  $i = j$  would be allowed an amplitude 2 would be possible for some  $i$ . The symmetry condition as above (including  $\pm w_i$ ) seems to be a new requirement compared to previous calculations, for instance, by Graham and Sloane [10]

Interestingly, this application in steganography, addresses rather the additive structure of the group  $Z_m$ , whereas the factorizations important for the analysis of the change of amplitude in only one component, can be analyzed via factorizations, which of course rely on the multiplicative structure of  $Z_m$ .

**Group factorizations in cryptography:** Factorizations of groups are also used to construct cryptosystems, which may replace RSA when, for instance, reliable quantum computers will once be in use [23]. However, the group  $Z_p$  important for our error–correcting codes is too simple for such applications and does not allow a one–way function for encoding.

**Double error correction:** In principle, it would also be possible to use one checksum (1) in order to correct more than one error. For instance, Lisonek's idea for steganography may in some cases be transferred to error–correcting codes (steganography, however, is more related to covering than

to packing). Usually, a second check will be carried out, as we saw in the application of SEC–DED codes. These codes, however, only correct bit flips. The analysis of 2–error correcting codes for even the simplest symbol changes is extremely difficult. In [16] one construction is provided. Even more difficult is the correction of errors of distance 2 in the Lee metric. The reason is that this may arise in two ways. Either one component  $c_i$  is distorted to  $c'_i = c_i \pm 2$  or two components are distorted by  $c'_i = c_i \pm 1$  and  $c'_j = c_j + 1$ . This is a combination of the single–error correction as studied by Martirosian [20] and the double–error correction related to the steganographic model discussed by Lisonek.

**Further applications:** Further applications of the checksum (1) arise in coding for memories with defects [2] and for tilings by certain polyominoes as studied by Golomb [8].

### III. CONSTRUCTION OF INTEGER CODES

We shall concentrate on groups  $Z_p$  where  $p$  is a prime number. In this case the multiplicative group  $Z_p^* = (Z_p \setminus \{0\}, \cdot)$  consists of all numbers  $\{1, \dots, p-1\}$  and a splitting  $(\mathcal{E}, \mathcal{H})$  – or, equivalently, the set  $\mathcal{H}$  obtained from the weights in (1) jointly with the error set  $\mathcal{E}$  – corresponds to a factorization  $\mathcal{E} \cdot \mathcal{H}$  of the group  $Z_p^*$ .

For a composite number  $m = p_1^{s_1} \dots p_r^{s_r}$  a perfect integer code in  $Z_m$  can be obtained from the perfect integer codes in  $Z_{p_i}$  for the prime factors  $p_i$ ,  $i = 1, \dots, r$ , of  $m$ . For sets  $\mathcal{E}$  of small size it has even been shown that this is the only way to obtain perfect integer codes for composite  $m$  [27], [22].

Since we are only interested in symmetric errors, the error sets  $\mathcal{E}$  under consideration are of the form  $\{\pm 1, \pm a_1, \dots, \pm a_{k-1}\}$ . We assume the 1 to be contained in  $\mathcal{E}$  for technical reasons. Condition (4) allows us to identify the elements  $x$  and  $-x$  in  $Z_p^*$  and hence to consider factorizations by the set  $\{1, a_1, \dots, a_{k-1}\} \in Z_p^*/\{1, -1\}$ . Observe that, since  $g^{\frac{p-1}{2}} = -1$ , a generator  $g$  of  $Z_p^*$  also generates the group  $Z_p^*/\{1, -1\}$ , which hence is also cyclic.

The idea here will be to arrange that the set  $\mathcal{H}$  obtained from the weights in the definition of an integer code (1) consists of a subgroup  $\mathcal{G}$  of  $Z_p^*/\{1, -1\}$  and its translates in the cosets of  $\mathcal{G}$ . Since  $Z_p^*/\{1, -1\}$  is a cyclic group, it is generated by one element  $g$ , i. e., all elements  $x = 1, \dots, \frac{p-1}{2}$  are a power  $x = g^{ix}$  of the generator  $g$ . If  $\mathcal{G}$  is a subgroup of  $Z_p^*/\{1, -1\}$ , its order must be a divisor of  $\frac{p-1}{2}$  and  $\mathcal{G}$  itself must be generated by a power of  $g$ , i. e. for some  $t$  dividing  $\frac{p-1}{2}$

$$\mathcal{G} = \{g^{jt} : j = 0, \dots, \frac{p-1}{2t}\}.$$

**THEOREM 1:** Let  $\mathcal{E} = \{\pm 1, \pm a_1, \dots, \pm a_{k-1}\}$  be the error set of an integer code, let  $g$  be a generator of  $Z_p^*$  and let  $a_i = g^{\nu_i}$  in  $Z_p^*/\{1, -1\}$  for  $i = 1, \dots, k-1$ . Then a perfect integer code with error set  $\mathcal{E}$  exists in  $Z_p$ , exactly if for some divisor  $l$  of  $\frac{p-1}{2k}$  the powers  $\nu_i$  are such that  $\nu_i = l\mu_i$  for  $i = 0, \dots, k-1$ , where the  $\mu_i$ 's fall into the different congruence classes modulo  $k$ , i. e.,

$$\{\mu_1 \bmod k, \dots, \mu_{k-1} \bmod k\} = \{1, \dots, k-1\}. \quad (4)$$

From this theorem the next algorithm is immediate:

**Algorithm IntegerCode** (set  $\mathcal{E} = \{\pm 1, \pm a_1, \dots, \pm a_{k-1}\}$ , prime number  $p$ )

- (1) Find a generator  $g$  of  $Z_p^*/\{1, -1\}$
- (2) for  $i = 1$  to  $k-1$  write  $a_i = g^{\nu_i}$  in  $Z_p^*/\{1, -1\}$
- (3) for all divisors  $l$  of  $\frac{p-1}{2k}$ 
  - (3a) write  $\nu_i = l\mu_i$
  - (3b) if  $\{\mu_0 \bmod k, \dots, \mu_{k-1} \bmod k\} = \{0, \dots, k-1\}$  output  $(\mathcal{G} = \{(g^l)^{jk}, j = 0, \dots, \frac{p-1}{2k}\})$

This theorem was derived in [32] and hence been applied to investigate the existence of perfect integer codes.

Since perfect integer codes seem to be quite sparsely distributed one might relax the conditions and no longer require perfectness but a good packing with the error spheres.

The special error set  $\mathcal{E} = \{\pm 1, \pm 2, \dots, \pm k\}$  is also denoted as  $F(k)$  in the literature. We say that  $F(k)$   $n$ -packs  $\mathcal{G}$  with packing set  $\mathcal{H}$  of size  $n$  if all products  $m \cdot h$  with  $m \in F(k)$ ,  $h \in \mathcal{H} \subset \mathcal{G}$  are different. Of course, then  $\mathcal{H}$  is a  $k$ -shift code of size  $n$ .

Let  $m(k, n)$  denote the size of the smallest group  $\mathcal{G}$  such that a  $k$ -shift code of size  $n$  exists in  $\mathcal{G}$ . For a good shift code (or the corresponding packing of the group  $\mathcal{G}$ ) one would expect that  $m(k, n)$  is not much bigger than the theoretical lower bound  $2nk + 1$ , which is obtained for a perfect  $k$ -shift code.

Such packings have been considered e.g in [6], [12], [28]. Some applications to Information Theory are discussed in [26] and [11]. The following asymptotical result is known [26].

$$\lim_{k \rightarrow \infty} \frac{m(k, n)}{k^2} = 1$$

Motivated by the geometric application of tiling the space with certain star bodies, where  $n$  is the dimension of the space, here the parameter  $n$  is fixed and  $k$  tends to infinity. The result shows that good packings in this case cannot be expected, since  $m(k, n)$  is about  $k^2$ , which is much bigger than  $2kn + 1$  for  $n$  small compared to  $k$ .

For applications in Coding Theory, however, one would rather fix  $k$  and look for code constructions suitable for any  $n$ .

In [27] several constructions for packings by the cross  $F(k)$  ( $k$ -shift codes) are presented. Especially, an almost perfect  $(p-1)$ -shift code of size  $p+1$  exists in cyclic groups of order  $2p^2$  for an odd prime number  $p$ . For instance, if  $p = 5$ , then  $Z_{50}$  is packed by  $F(4) = \{\pm 1, \pm 2, \pm 3, \pm 4\}$  with packing set  $\mathcal{S} = \{1, 5, 9, 11, 19, 21\}$ . Observe that all the products  $m \cdot h$  of elements  $m \in F(4)$  and  $h \in \mathcal{S}$  are different and that only the elements 0 and  $p^2 = 25$  in  $Z_{50}$  cannot be obtained as such products. Indeed, the general structure of the shift code is

$$\mathcal{S} = \{1, p, 2p \pm 1, 4p \pm 1, \dots, 2\frac{p-1}{2}p \pm 1\}$$

Another idea would be to follow the constructions in Theorem 1 and no longer requiring perfectness. So one has to arrange that for a generator  $g$  the powers  $\mu_1^{-1}0, \dots, \mu_k$  of  $1 = g^{\mu_1}, 2 = g^{\mu_2}, \dots, k0g^{\mu_k}$  fall into different congruence classes. In this case a close packing will be obtained as above (if some further divisibility conditions hold). This way, one can also see, that the packings become worse the fewer residue classes are occupied by the  $\mu_i$ 's.

Some further sporadic constructions in [27] usually proceed by following the orbit of special elements in the group.

For the systematic search of 3–shift codes, this gave us the idea to the following greedy algorithm:

Consider the orbit of the element 3 in the cyclic group  $Z_l$ , i.e. the set

$$\mathcal{F} = \{3^s : s = 0, \dots, \text{ord}(3)\}.$$

Starting with  $i = 0$  include the element  $3^i$  in the shift code  $\mathcal{S}$  if possible, and set  $i \leftarrow i+1$ . When the search in  $\mathcal{F}$  is finished, continue with the same procedure in the residue classes  $a \cdot \mathcal{F}$ ,  $a \in \mathcal{G}$ .

This way, we found some quite good shift codes: For instance, in  $Z_{40}$  the 3–shift code  $\{1, 4, 5, 7, 9, 17\}$  of size 6 improves the value in Table V-4 on p. 316 in [27], where only an example of a 3–shift code of size 6 in  $Z_{43}$  was given. Further,  $Z_{56}$  contains the 3–shift code  $\{1, 4, 5, 7, 9, 11, 13, 25\}$  of size 8 and in  $Z_{88}$  there is a 3–shift code of size 13, namely  $\{1, 4, 7, 9, 11, 15, 17, 23, 25, 31, 36, 39, 41\}$ .

Observe that all the group orders  $l$  here are divisible by 4. In this case the algorithm behaves very nice, since usually almost all odd elements and almost all elements  $\equiv 2 \pmod{4}$  are included in some sphere around a codeword in the shift code.

A similar construction – following the orbit of the element 2 – allowed us to find the following 4–shift code  $\{1, 5, 8, 9, 11, 13, 14, 17, 23, 35, 37, 40\}$  of size 12 in  $Z_{99}$ .

The Greedy algorithm does not always perform very well. Indeed, we have examples where it finds very bad packings. It would be interesting to find conditions under which good packings can be obtained using the Greedy algorithm.

Further, here it was only applied to the sets  $F(3)$  and  $F(4)$ . The reason is that one should follow the orbit of one element 2, say, in the powers of the other one, then 3. For  $F(5)$  one also has to consider the powers of the element 5, which becomes much more difficult. It would be interesting to find good packings in this case.

## REFERENCES

- [1] B.F. Albdaiwi and B. Bose, "Quasi-perfect resource placement for two-dimensional toroidal networks", *J. Par. Distrib. Comput.*, vol. 65, pp. 815 – 831, 2005.
- [2] E. E. Belitskaja, V. R. Sidorenko, and P. Stenström, "Testing of memory with defects of fixed configurations", *Proceedings of 2nd International Workshop on Algebraic and Combinatorial Coding Theory*, Leningrad, pp. 24 – 28, 1990
- [3] N. Biggs, "Perfect codes in graphs", *J. Combin. Theory Ser. B* 15, pp. 289 – 296, 1973.
- [4] H. Chen and N. Tzeng, "Efficient resource placement in hypercubes using multiple-adjacency code", *IEEE Trans. Comput.* 43, pp. 23 – 33, 1994.
- [5] S. D. Constantin and T. R. N. Rao, "On the theory of binary asymmetric error correcting codes", *Information and Control*, vol. 40, pp. 20 – 26, 1979.
- [6] H. Everett and D. Hickerson, "Packing and covering by translates of certain nonconvex bodies", *Proceedings of the American Mathematical Society*, vol. 75, no. 1, pp. 87 – 91, 1979.
- [7] S. Golomb, "A general formulation of error metrics", *IEEE Trans. Inform. Theory*, vol. 15, pp. 425 – 426, 1969.
- [8] S. Golomb, *Polyominoes*, Princeton University Press, 2nd ed., 1994.
- [9] S. W. Golomb and L. R. Welch, "Algebraic coding and the Lee metric", *Error Correcting Codes (H. B. Mann ed.)*, pp. 175 – 194, 1968.
- [10] R.L. Graham and N.J.A. Sloane, "On additive bases and harmonious graphs", *SIAM J. Algebr. Discr. Math.*, vol. 1, pp. 382 – 404, 1980
- [11] W. Hamaker and S. Stein, "Combinatorial packing of  $R^3$  by certain error spheres", *IEEE Trans. Inform. Theory*, vol. 30, no. 2, pp. 364 – 368, 1984.
- [12] D. Hickerson and S. Stein, "Abelian groups and packing by semi-crosses", *Pacific J. Math.*, vol. 122, no. 1, pp. 95 – 109, 1986.
- [13] P. Horak, "On perfect Lee codes", *Discrete Math.*, vol. 309, pp. 5551 – 5556, 2009
- [14] J. Jerebic, S. Klavžar, S. Špacapan, "Characterizing  $r$ -perfect codes in direct products of two and three cycles", *Inform. Process. Lett.* 94, no. 1, pp. 1 – 6.
- [15] P.K. Jha, "Perfect  $r$ -domination in the Kronecker product of three cycles", *IEEE Trans. Circuit Systems – I: Fundamental Theory Appl.* 49, pp. 89 – 92, 2002.
- [16] H. Kostadinov, N. Manev, and H. Morita, "Double  $\pm 1$  error - correctable codes and their applications to modulation schemes", *Proceedings 11th Int. Workshop Algebraic and Combinatorial Coding Theory*, Pamporovo, Bulgaria, pp. 155 – 160, June 2008.
- [17] V. I. Levenshtein, "Binary codes with correction for deletions and insertions of the symbol 1", (in Russian), *Problemy Peredachi Informacii*, vol. 1, pp. 12 – 25, 1965.
- [18] V. I. Levenshtein and A. J. H. Vinck, "Perfect  $(d,k)$ -codes capable of correcting single peak shifts", *IEEE Trans. Inform. Theory*, vol. 39, no. 2, pp. 656–662, 1993.
- [19] P. Lisonek, "Sum covers in steganography", *Proceedings 11th Int. Workshop Algebraic and Combinatorial Coding Theory*, Pamporovo, Bulgaria, pp. 186 – 191, June 2008.
- [20] S. Martirosyan, "Single – error correcting close packed and perfect codes", *Proceedings of 1st INTAS International Seminar on Coding Theory and Combinatorics*, Thahkadzor, Armenia, pp. 90 – 115, 1996.
- [21] H. Morita, A. Geysler, and A. J. van Wijngaarden, "On integer codes capable of correcting single errors in two – dimensional lattices", *Proceedings 2003 IEEE Int. Symp. Inform. Theory*, p. 16, 2003.
- [22] A. Munemasa, "On perfect  $t$ -shift codes in Abelian groups", *Designs, Codes, and Cryptography*, vol. 5, pp. 253 – 259, 1995.
- [23] M. Qu and S.A. Vanstone, "Factorizations of elementary abelian  $p$ -groups and their cryptographic significance", *J. Cryptology*, vol. 7, pp. 201–212, 1994.
- [24] A.D. Sands and S. Szabo, *Factoring Groups into Subsets*, CRC Press, 2009.
- [25] N.J.A. Sloane, "On single deletion-correcting code", *Codes and Designs (Ray-Chaudhuri Festschrift)*, K.T. Arasu and A. Seress (eds.), de Gruyter, Berlin, pp. 490 – 499, 2002.
- [26] S. Stein, "Packing of  $R^n$  by certain error spheres", *IEEE Trans. Inform. Theory*, vol. 30, no. 2, pp. 356 – 363, 1984.
- [27] S. Stein, "Tiling, packing, and covering by clusters", *Rocky Mountain J. Math.*, vol. 16, 277 – 321, 1986.
- [28] S. Stein, "Packing tripods", *Math. Intelligencer*, vol. 17, no. 2, pp. 37 – 39, 1995.
- [29] S. Stein and S. Szabó, *Algebra and Tiling*, The Carus Mathematical Monographs 25, The Mathematical Association of America, 1994.
- [30] S. Szabó, *Topics in Factorization of Abelian Groups*, Birkhäuser, 2004.
- [31] U. Tamm, "Splittings of cyclic groups and perfect shift codes", *IEEE Trans. Inform. Theory*, vol. 44, no. 5, pp. 2003 – 2009, 1998.
- [32] U. Tamm, "On perfect integer codes", *Proceedings 2005 IEEE Int. Symp. Inform. Theory*, Adelaide, Australia, 2005.
- [33] R. R. Varshamov and G. M. Tenengolts, "One asymmetric error correcting codes", (in Russian), *Avtomatika i Telemekhanika*, vol. 26, no. 2, pp. 288 – 292, 1965.
- [34] A.J.H. Vinck and H. Morita, "Codes over the ring of integers modulo  $m$ ", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E81–A, no. 10, pp. 2013 – 2018, 1998.