On Generation of Error Vectors[1]

Otokar Grošek, Viliam Hromada

Institute of Computer Science and Mathematics
Slovak University of Technology

The 16th Central European Conference on Cryptology
Piešťany, Slovakia, June 2016

NATO Science for Peace and Security Programme

[1] Supported by NATO SPS project 984520.

Motivation

- Generation of n-bit vectors with Hamming weight t (constant weight codewords, error vectors) is an important part of various cryptosystems.
  - E.g. in McEliece cryptosystem, we need to generate a random error vector of a given length n and with a specific Hamming weight t.
  - In Niederreiter cryptosystem, not only we need to have a constant weight codeword, we also want to be able to convert it to a binary string and vice versa.
- There are many proposed algorithms.
- We propose a new method, interesting from an algebraic point of view.

Generation of a random n-bit vector with t ones

There are many ways, how to uniformly generate a random error vector with a good pseudorandom generator, e.g.:

- **Knuth shuffling** - start with a n-bit vector $\underbrace{11...1}_{t}\underbrace{0...0}_{n-t}$ and t-times swap one of the ones with a bit at a random position in the vector.
- $n = 5, t = 2 : w = 11000$
  1. $r = 3, r \in_R \{1, ..., 5\} : 01100$
  2. $r = 5, r \in_R \{2, ..., 5\} : 00101$
- Let $w[i]$ be the i-th bit of the vector w.
- It can be proved that the probability $Pr(w[i] = 1) = \frac{t}{n}$ for any $i \in \{1, ..., n\}$ in the resulting vector.
- The assumption is that we are able to uniformly select numbers at random modulo different integers.

Generation of a random n-bit vector with t ones

- **Permutation of n elements** - a permutation of n elements can be used to determine the positions of one bits.
- $n = 5, t = 3 : w = 00000, C = \{1, 2, 3, 4, 5\}$
  1. $r = 3, r \in_R \{1, ..., 5\} : C[r] = 3, 00100, C = \{1, 2, 5, 4, 5\}$
  2. $r = 3, r \in_R \{1, ..., 4\} : C[r] = 5, 00101, C = \{1, 2, 4, 4, 5\}$
  3. $r = 2, r \in_R \{1, ..., 3\} : C[r] = 2, 01101, C = \{1, 4, 4, 4, 5\}$
- The assumption is that we are able to uniformly select numbers at random modulo different integers.

Generation of a random n-bit vector with t ones

- **Trellis generator** (proposed as suitable for hardware implementation by Butler, Sasao in 2011)
- The resulting vector is generated in n clocks - bit by bit.
- The probabilities of generating a bit 0/1 constantly changes according to the previously generated bits, so that all possible vectors have the same probability.
- In the beginning, the probability of a bit 1 is $\frac{t}{n}$.
  - Bit 0 => the probability changes to $\frac{t}{n-1}$.
  - Bit 1 => the probability changes to $\frac{t-1}{n-1}$.
- Easy way is to generate a number from a certain set and have a threshold for bits 0/1.
- The assumption is that we are able to uniformly select numbers at random modulo different integers.

Generation of a random n-bit vector with t ones - Trellis

$n = 5, t = 2$. Green are numbers associated with 0 bit, red are numbers associated with 1 bit.

1. $\{1, 2, 3, 4, 5\}$. $r \in_R \{1, ..., 5\}, r = 2 \Rightarrow w = 0$
2. $\{1, 2, 3, 4\}$. $r \in_R \{1, ..., 4\}, r = 1 \Rightarrow w = 00$
3. $\{1, 2, 3\}$. $r \in_R \{1, ..., 3\}, r = 3 \Rightarrow w = 001$
4. $\{1, 2\}$. $r \in_R \{1, ..., 2\}, r = 2 \Rightarrow w = 0011$
5. $\{1\}$. $r \in_R \{1, ..., 1\}, r = 1 \Rightarrow w = 00110$

Index to constant weight codeword converter

- More advanced methods are able to generate a constant weight codeword from a binary sequence / integer.
- And also convert the constant weight codeword back to the original piece of information.
- One of the well-known methods is an algorithm based on Combinatorial Number System (Cover, 1973).
- A different one, based on run-length encoding, was proposed by Sendrier in 2005.

CNS

- There are $\binom{n}{t}$ n-bit words with Hamming weight t.
- Every number N from 0 to $\binom{n}{t}$ can be expressed by a Combinatorial Number System, i.e.

$$N = \binom{c_t}{t} + \binom{c_{t-1}}{t-1} + \ldots + \binom{c_1}{1},$$

  where $c_t > c_{t-1} > \ldots > c1 \geq 0$.
- The set of digits c_i is unique for each N.
- Digits c_i represent positions of 1 bits in a n-bit vector, which defines a bijective mapping between constant weight codewords of length n, weight t and integers ranging from 0 to $\binom{n}{t} - 1$.

CNS

$n = 5, t = 2$. There are $\binom{5}{2} = 10$ different vectors.

N	Representation	Codeword
0	$\binom{1}{2} + \binom{0}{1}$	00011
1	$\binom{2}{2} + \binom{0}{1}$	00101
2	$\binom{2}{2} + \binom{1}{1}$	00110
3	$\binom{3}{2} + \binom{0}{1}$	01001
4	$\binom{3}{2} + \binom{1}{1}$	01010
5	$\binom{3}{2} + \binom{2}{1}$	01100
6	$\binom{4}{2} + \binom{0}{1}$	10001
7	$\binom{4}{2} + \binom{1}{1}$	10010
8	$\binom{4}{2} + \binom{2}{1}$	10100
9	$\binom{4}{2} + \binom{3}{1}$	11000

Run-length encoding

- Sendrier proposed a way how to encode constant weight codewords into binary strings (and vice-versa).
- It is a modified Golomb's run-length encoding.
- The original encoding scheme uses a parameter d which depends on the probability that a memoryless source generates a bit 0 (or 1).
- Then, rather than encoding some binary sequence, it encodes the lengths of runs of zero bits.
- Sendrier's modification changes the parameter d after the encoding of each run and is used to encode the constant weight codewords into binary strings.
- One drawback of this method is a variable length of the resulting binary strings.

Run-length encoding

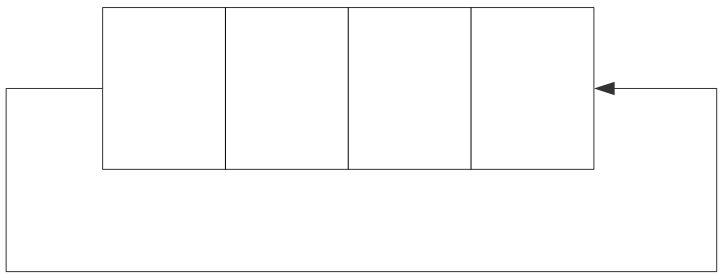
$n = 5, t = 2$.

Codeword	Run-length	Binary sequence
00011	[3 0]	11
00101	[2 1]	1001
00110	[2 0]	10000
01001	[1 2]	011
01010	[1 1]	0101
01100	[1 0]	0100
10001	[0 3]	0011
10010	[0 2]	00100
10100	[0 1]	0001
11000	[0 0]	0000

New proposed method

- We propose a new method, which is not very practical, but is interesting from an algebraic point-of-view.
- Let $V_n = \mathbb{F}_2^n$ be the n-dimensional vector space and $E_t = \{e | hwt(e) = t\} \subset V_n$ be the set of all binary vectors of length n and weight t.
- Let **A** be the associated $n \times n$ matrix to the characteristic polynomial $f(x) = x^n + 1$ over $\mathbb{F}_2$ of the LFSR.

- $f(x) = x^n + 1$ is a characteristic polynomial of a LFSR whose new feedback bit is equal to its output bit.
- Since its length is n, the period of its output is also at most n.
- E.g. a polynomial $f(x) = x^4 + 1$ corresponds to a LFSR:



- Its associated matrix:

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Let $[u] = \{u, u\mathbf{A}, u\mathbf{A}^2, ..., u\mathbf{A}^{d-1}\}$ be a class of words obtained from u by consecutive shifts, where d is the smallest period.

$$\begin{pmatrix}1 & 1 & 0 & 0\end{pmatrix}\mathbf{A} = \begin{pmatrix}1 & 1 & 0 & 0\end{pmatrix}\begin{pmatrix}0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{pmatrix} = \begin{pmatrix}1 & 0 & 0 & 1\end{pmatrix}$$

$$\begin{pmatrix}1 & 0 & 0 & 1\end{pmatrix}\mathbf{A} = \begin{pmatrix}0 & 0 & 1 & 1\end{pmatrix}$$

$$\begin{pmatrix}0 & 0 & 1 & 1\end{pmatrix}\mathbf{A} = \begin{pmatrix}0 & 1 & 1 & 0\end{pmatrix}$$

$$\begin{pmatrix}0 & 1 & 1 & 0\end{pmatrix}\mathbf{A} = \begin{pmatrix}1 & 1 & 0 & 0\end{pmatrix}$$

$$[1100] = \{1100, 1001, 0011, 0110\}$$

$$[1010] = \{1010, 0101\}$$

- We define a relation ρ_n such that $u\rho_n v$ iff u, v belong to the same class.
- This relation is an equivalence relation.
- $u\rho_n = [u]$.
- For any u the cardinality of $[u] = u\rho_n$ divides the order of $\mathbf{A}$ in the general linear group $GL(n, \mathbb{F}_2)$.

Theorem

Let ρ_n be the equivalence relation on E_t defined such that $u \rho_n v$ iff both u, v belong to the same class. Then

1. *There exists a class $u\rho_n$ with cardinality r if and only if $r \mid n$ and $\frac{n}{r} \mid t$.*
2. *All classes have the same cardinality if and only if $gcd(n, t) = 1$.*

Example

Let $n = 20, t = 10$. Then we have the following classes:

- 9225 classes with cardinality $r = 20$,
- 25 classes with cardinality $r = 10$,
- 1 class with cardinality $r = 4$,
- 1 class with cardinality $r = 2$.

Classes with a given cardinality

- Let $C(n, t, r)$ denote the number of classes with cardinality r for given n and t.
- E.g. in the previous example $C(20, 10, 2) = 1, C(20, 10, 20) = 9225$.
- According to the theorem and definition of ρ_n:

$$\binom{n}{t} = \sum_{\substack{r|n \\ n/r|t}} rC(n, t, r). \tag{1}$$

- 2 special cases:

$$C(n, n, r) = \begin{cases} 1, & \text{if } r = 1; \\ 0, & \text{if } r > 1; \end{cases} \qquad C(n, 0, r) = \begin{cases} 1, & \text{if } r = 1; \\ 0, & \text{if } r > 1. \end{cases}$$

Classes with a given cardinality

Theorem

Let for given n, t, $0 < t < n$, be an integer r, $1 < r < n$, such that $r \mid n$ and $\frac{n}{r} \mid t$. Then there exists at least one class $|u\rho_n| = r$ and word w of the length r and weight $\frac{tr}{n}$ such that

$$u = \overbrace{w||w||...||w}^{n/r} = w^{n/r}$$

Moreover, let $d = gcd(n, t)$ and $d(g)$ be the number of divisors of g. Then we have precisely $d(g)$ nonzero summands in the formula (1).

Examples

- $n = 6, t = 6$. This is one of the special cases, since $n = t$.
  $\binom{6}{6} = 1C(6, 6, 1) = 1 \times 1$.
- $n = 20, t = 10 \Rightarrow gcd(20, 10) = 10, d(10) = 4$
  $\binom{20}{10} = 20C(20, 10, 20) + 10C(20, 10, 10) + 4C(20, 10, 4) + 2C(20, 10, 2) = 20 \times 9225 + 10 \times 25 + 4 \times 1 + 2 \times 1$
- $n = 1024, t = 70 \Rightarrow gcd(1024, 70) = 2, d(2) = 2$.
  $\binom{1024}{70} = 1024C(1024, 70, 1024) + 512C(1024, 70, 512)$.
  In the case $r = 512, u = w||w$.
  $512C(1024, 70, 512) = \binom{512}{35} \Rightarrow C(1024, 70, 512) = \frac{\binom{512}{35}}{512}$
  $C(1024, 70, 1024) = \frac{\binom{1024}{70} - \binom{512}{35}}{1024}$

Generation of a random constant weight word

- We can use the fact that all constant weight words of length n and weight t are divided into equivalence classes with r elements.
- First, randomly select one of the equivalence classes.
- Second, randomly select one of the elements from the equivalence class.
- It may happen that different classes have different number of elements.
- So that the probability distribution of selecting a particular class has to be proportionate to the cardinalities.

Example

Considering $n = 20, t = 10$, one may choose:

- an equivalence class with cardinality $r = 20$ with probability $\frac{20}{\binom{20}{10}}$.
- an equivalence class with cardinality $r = 10$ with probability $\frac{10}{\binom{20}{10}}$.
- an equivalence class with cardinality $r = 4$ with probability $\frac{4}{\binom{20}{10}}$.
- an equivalence class with cardinality $r = 2$ with probability $\frac{2}{\binom{20}{10}}$.

And afterwards a random number from the set $\{0, ..., r - 1\}$ to get the corresponding shift of the representative of the class.

Example

- For a class $[00110101010101010101]\rho_n$ with $r = 20$, if a random shift 3 is generated then the resulting vector is 10101010101010101001.
- For a class $[00110011001100110011]\rho_n$ with $r = 4$, if a random shift 1 is generated then the resulting vector is 01100110011001100110.

Thank you for your attention!