Linear Sequences and Punctured Linear Codes

Michele Elia and Emanuele Viterbo
Dipartimento di Elettronica
Politecnico di Torino
Corso Duca degli Abruzzi 24
I-10129 Torino, Italy
e-mail: `name@polito.it`

ABSTRACT

We examine the interplay between the generation of linear sequences and the encoding of linear codes; in particular we analyze the effect of "code puncturing", having principally in mind cryptographic applications.

Keywords: Linear codes, Linear recurring sequences, Binary Goppa codes, Matrix representation.

1. INTRODUCTION

Binary *pseudorandom* sequences are used in many fields including cryptography, testing of digital devices, spread spectrum techniques and pattern recognition. The literature concerning binary sequences is vast and wide-ranging with many profound and beautiful results, and we will not even try to give a list, which would inevitably be incomplete.

In this paper, we consider the limited interplay between the generation of linear sequences and linear codes, with particular attention to the effect of "code puncturing", having in mind as a concrete application cryptography stream ciphering [2]. In this context, a binary sequence must have a long *period*, a good *linear complexity profile* (LCP), good *randomness* and an *inversion resistence*.

Let us briefly recall that the linear complexity profile is defined as the length of the shortest Linear Feedback Shift Registers (LFSR) that generates a sequence up to length n for every n. Randomness is defined in Lehmer's sense [10] and is commonly evaluated using a battery of randomness tests based on a Montecarlo principle. Inversion resistence means that it should be easy to produce the sequence from an initial state but hard to recover such an initial state from a known piece of sequence.

It is well known that a block $[x(0), x(1), \ldots, x(T-1)]$ of $T = 2^m - 1$ consecutive symbols in a binary sequence $\mathfrak{X}$ produced by a LFSR with primitive generator polynomial $g(x)$ of degree m is a codeword of a dual Hamming code $(2^m - 1, m, 2^{m-1})$ [8]. The codeword is generated, symbol by symbol, during the evolution of the LFSR, where $x(n) = \mathbf{x}^T \cdot \mathbf{u}(n)$ is computed as a linear function of the LFSR state $\mathbf{u}(n)$ and $\mathbf{x}$ is a fixed vector. In this situation, the linear complexity of the sequence is m, and it is quite easy to compute the state of LFSR from $2m$ consecutive symbols of $\mathfrak{X}$, [3].

An interesting generalization of this scheme is obtained by considering other codes, for instance dual BCH codes $(2^m - 1, mt, d)$ or dual Goppa codes $(2^m, mt, d)$, instead of dual Hamming codes. Therefore, the evolution of a LFSR is used to drive the choice of the element α of $\mathbb{F}_{2^m}$ which identifies a column $\mathbf{g}(\alpha)$ of the generator matrix $G = (\mathbf{g}(\alpha_1), \mathbf{g}(\alpha_2), \ldots, \mathbf{g}(\alpha_{2^m-1}))$ used to compute the output bit as the element $\mathbf{x}^T \cdot \mathbf{g}(\alpha)$ of a codeword that corresponds to the information vector $\mathbf{x}$. The "cost" of generating a bit in a code word of a BCH or a Goppa code is normally greater than the cost of generating a bit in a word of a Hamming code. On the other hand, an undoubted advantage is the increased difficulty of "de-engineering" the generated sequence with the aim of finding the initial content $\mathbf{u}(0)$ of the LFSR from a piece of $\mathfrak{X}$.

The choice of code does not increase the linear complexity of the generated sequence so much, a goal that can be achieved using clock-controlled LFSRs. The visible result of this expedient is a cancellation of elements from $\mathfrak{X}$, an operation that yields a sequence with shorter period but, in general, with a good linear complexity profile. This operation is equivalent to puncturing the codewords.

The effects of using linear codes other than Hamming codes and clock-controlled LFSR are investigated and some general conclusions presented.

2. FINITE STATE MACHINES

Linear feedback shift registers are particular Finite State Machines (FSM) whose evolution is governed by linear functions. A FSM is an object $\mathcal{M} = (\mathcal{U}, \mathcal{A}, \mathcal{B}, \mathfrak{s}, \mathfrak{o}, \mathbf{u}(0))$ which consists of

1. a set $\mathcal{U}$ of m-dimensional binary vectors $\mathbf{u}$ called *states*;
2. an input sequence $\mathfrak{c} = c(0), c(1), \ldots, c(n), \ldots,$ of symbols from an alphabet $\mathcal{A}$;
3. a state transition function $\mathfrak{s}$ defined from $\mathcal{U} \times \mathcal{A}$ into $\mathcal{U}$;
4. an output function $\mathfrak{o}$ defined from $\mathcal{U} \times \mathcal{A}$ into an output alphabet $\mathcal{B}$;
5. an initial state $\mathbf{u}(0)$.

Throughout the paper it will be assumed that both $\mathcal{A}$ and $\mathcal{B}$ coincide with $\mathbb{Z}_2$.

The evolution of a FSM is said to be autonomous when the input, or driving, sequence is missing, otherwise it is said to be non-autonomous. This evolution is described by a sequence of states

$$\mathfrak{S} = \mathbf{u}(0), \mathbf{u}(1), \ldots \mathbf{u}(n), \ldots,$$

where $\mathbf{u}(n) = \mathfrak{s}[\mathbf{u}(n-1), c(n)]$ and the output binary sequence $\mathfrak{X} = x(0), x(1), \ldots, x(n), \ldots$ is generated as $x(n) = \mathfrak{o}(\mathbf{u}(n), c(n))$.

When the evolution is autonomous then $\mathfrak{S}$ is periodic with period the minimum integer T such that $\mathbf{u}(n+T) = \mathbf{u}(n)$, and the output sequence $\mathfrak{X}$ has a period N which is possibly a submultiple of T. If $N = T$ the function $\mathfrak{o}$ is said to be non-degenerate.

LFSR

The state transition function $\mathfrak{s}$ of a binary LFSR of length m is specified by a binary $m \times m$ matrix A and a fixed vector $\mathbf{w}$ which are used to compute the next state as

$$\mathbf{u}(n) = A\mathbf{u}(n-1) + c(n)\mathbf{w} \qquad \forall n \geq 1$$

If the evolution of a LFSR is autonomous, then $\mathbf{w} = \mathbf{0}$ is the all-zero vector, and the period T depends on both initial state and characteristic polynomial $m(x)$ of A. If $m(x)$ is primitive and $\mathbf{u}(0)$ is not the all-zero vector, then $T = 2^m - 1$. In this case the states of the LFSR represent all non-zero elements of $GF(2^m)$. A LFSR with primitive polynomial can be forced to generate every element of $GF(2^m)$, zero included, by a convenient driving sequence.

Proposition 1 *The all-zero state can be inserted between states* $\mathbf{a}$ *and* $\mathbf{b} = A\mathbf{a}$ *by taking* $\mathbf{w} = \mathbf{b}$ *and driving sequence* $c(n)$ *with all zeros except a pair of* 1*s in correspondence to states* $\mathbf{a}$ *and* $\mathbf{b}$.

PROOF. The evolution is essentially autonomous except at the occurrence of state $\mathbf{a}$, when we have the sequence of states

$$\ldots\ \mathbf{a}\ ,\ \ \mathbf{0} = A\mathbf{a}+\mathbf{w}\ \ ,\ \ \mathbf{b} = A\mathbf{0}+\mathbf{w} = A\mathbf{a}\ \ ,\ \ A\mathbf{b} = A^2\mathbf{a}\ \ldots$$

$\square$

We note that the driving sequence is an easily generated periodic sequence.

3. CLOCK-CONTROLLED LFSR

Let A and B be $m \times m$ binary matrices. The simplest, although fairly general, form of the state transition function $\mathfrak{s}$ of a clock controlled binary LFSR is

$$\mathbf{u}(n) = [(1 - c(n))A + c(n)B]\mathbf{u}(n-1)$$

The binary control sequence $\mathfrak{c} = c(0), c(1), \ldots, c(n), \ldots$ is possibly generated by a LFSR. If $c(n) = f(\mathbf{u}(n-1))$ the machine evolution is autonomous and is called self-clock-controlled, then the feedback shift register can be viewed as a nonlinear device.
The state vector of a clock-controlled LFSR can be expressed at step n as a function of the initial state

$$\mathbf{u}(n) = \prod_{i=1}^{n}[(1 - c(n))A + c(n)B]\mathbf{u}(0)$$

From now on, let us assume that A and B are commuting binary matrices, therefore we have

$$\mathbf{u}(n) = A^{C_0(n)}B^{C_1(n)}\mathbf{u}(0) \tag{1}$$

where $C_0(n)$ and $C_1(n)$ denote the number of 0s and 1s in the sequence $\mathfrak{c}$ up to step n. The period N of the sequence $\mathfrak{S}$ is the minimum integer such that $\mathbf{u}(N + n) = \mathbf{u}(n)$. Therefore equation (1) implies

$$A^{C_0(N+n)}B^{C_1(N+n)} = I$$

and

$$C_0(N + n) + C_1(N + n) = N$$

for any n. In the following we will assume that the initial state $\mathbf{u}(0)$ belongs to a periodic cycle, so that we can write $C_0 + C_1 = N = C_0(N) + C_1(N)$.

Primitive LFSR

Let $\mathfrak{c}$ be generated by a primitive LFSR of length M. Let A be the companion matrix of a primitive polynomial $m(x)$, therefore, by Theorem 4 of the appendix, B is a nonsingular polynomial matrix in A of degree at most $m-1$, thus B is a suitable power A^{ℓ_B} of A. In a classical case $B = A^2$, which is equivalent to a two step advance of the LFSR when $c(n) = 1$.

Under these hypotheses, the period N is defined by the following equations

- $N = C_0 + C_1$
- $N = K_1(2^M - 1)$
- $C_0 = K_1\frac{2^M-2}{2}$ and $C_1 = K_1\frac{2^M}{2}$
- $C_0 + \ell_B C_1 = K(2^m - 1)$

where, K and K_1 are the smallest integers that satisfy the condition

$$K_1[(\ell_B + 1)2^{M-1} - 1] = K(2^m - 1) \qquad .$$

If $(\ell_B+1)2^{M-1}-1$ and (2^m-1) are relatively prime, then $K_1 = (2^m - 1)$, and the period $N = (2^M - 1)(2^m - 1)$. The situation is completely different for the self-clock-controlled LFSR.

Self-Clock-controlled LFSR

Let $\mathfrak{c}$ be produced by the relation $c(n) = f_1(\mathbf{u}(n-1))$ with $f_1(\cdot)$ being a function of the state vector $\mathbf{u}(n-1)$. Letting $B = A^{\ell_B}$, then the equation

$$\mathbf{u}(n) = [(1 - f_1(\mathbf{u}(n-1)))A + f_1(\mathbf{u}(n-1))B]\mathbf{u}(n-1)$$

shows that a self-clock-controlled LFSR is equivalent to a non-linear feedback shift register, with possibly a shorter period.
Assuming again that the initial state belongs to a periodic cycle, under the above hypotheses, the period N is defined by the following conditions

- $N = C_0 + C_1$
- $C_0 + \ell_B C_1 = 2^m - 1$

where C_0 and C_1 depend on $f_1(\cdot)$. For instance, in the case $B = A^2$ and $f_1(\mathbf{u}(n)) = u_i(n)$, the period is obtained as follows: the computation of C_1 is based on the distribution of the *runs* of 1s[1] in a sequence generated by a primitive LFSR of length m [1]

[1] A proper run of length k consists in k consecutive 1s between two 0s

- 1 run of length m
- 0 run of length $m-1$
- 1 run of length $m-2$
- 2^{m-k-2} runs of length k for $1 \leq k \leq m-3$

We obtain

$$C_1 = \frac{2^m - 1 + 2\delta}{3}$$

where δ is the remainder of $2^m - 1$ divided by 3, namely $\delta = 0$ if m is even, and $\delta = 1$ if m is odd,
$C_0 = 2^m - 1 - 2C_1 = \frac{2^m-1-4\delta}{3}$ and finally

$$N = \frac{2}{3}(2^m - 1) - \frac{2}{3}\delta$$

It is interesting to observe that, when m is even, the control sequence $\mathfrak{c}$ is perfectly balanced.

4. SEQUENCES AND CODES

The $T = 2^m - 1$ consecutive symbols of a binary sequence $\mathfrak{X}$ generated by a LFSR with primitive generator polynomial $g(x)$ of degree m form a codeword $[x(0), x(1), \ldots, x(T-1)]$ of a dual Hamming code. The codeword is generated, symbol by symbol, during the evolution of the LFSR, where the symbol $x(n) = \mathbf{x}^T \cdot \mathbf{u}(n)$ is computed as a linear function of the LFSR state $\mathbf{u}(n)$ and a fixed vector $\mathbf{x}$. Note that the vector $\mathbf{x}$ allows us to single out a specified codeword of the code, and n specifies the position of the bit within the codeword. In this situation, the linear complexity profile of the sequence is essentially the constant m, and it is quite easy to compute the state of LFSR from $2m$ consecutive symbols of $\mathfrak{X}$.
An interesting generalization of this scheme is obtained by considering other classes of codes, for instance dual BCH codes $(2^m-1, mt, d)$ or dual Goppa codes $(2^m, mt, d)$, instead of dual Hamming codes $(2^m-1, m, 2^{m-1})$. Therefore the evolution of a LFSR is used to drive the choice of the element α of $\mathbb{F}_{2^m}$, which identifies a column $\mathbf{g}(\alpha)$ of the generator matrix G.

BCH codes

The dual of a BCH $(2^m-1, mt, d)$ code is closely connected with a generalized Reed-Solomon code, and can be viewed as a trace code [8, p.349], namely the code that has words

$$\big(\mathrm{Tr}(f(1)), \mathrm{Tr}(f(\alpha)\alpha^{-r}), \mathrm{Tr}(f(\alpha^2)\alpha^{-2r}), \ldots \\ \ldots \mathrm{Tr}(f(\alpha^{2^m-2})\alpha^{-(2^m-2)r}))$$

where $f(z) \in \mathbb{F}_{2^m}[z]$ is a polynomial of degree less than r and α is a root of a primitive polynomial of degree m. A codeword of $\mathcal{C}$ can be viewed as a block of bits with length equal to the period of a binary sequence. The linear complexity, that is the length of the shortest LFSR that generates a codeword, is given by the following theorem.

Lemma 1 *Let $\mathfrak{s} = \{z_n = \mathrm{Tr}(\zeta\alpha^n)\}_{n=0}^{\infty}$ be a sequence generated by the trace function, where $\zeta, \alpha \in \mathbb{F}_{2^m}$, then the linear complexity $L(\mathfrak{s})$ is m, that is $\mathfrak{s}$ is produced by a linear recurrence of degree m.*

PROOF. Let $m(x) = a_m + a_{m-1}x + \ldots + a_1 x^{m-1} + x^m$ be a primitive minimal polynomial of α. We have

$$\begin{aligned} z_n &= \mathrm{Tr}(\zeta\alpha^{n-m}\alpha^m) \\ &= \mathrm{Tr}(\zeta\alpha^{n-m}(a_m + a_{m-1}\alpha + \ldots + a_1\alpha^{m-1})) \\ &= a_m\mathrm{Tr}(\zeta\alpha^{n-m}) + a_{m-1}\mathrm{Tr}(\zeta\alpha^{n-m+1}) + \cdots \\ &\quad \cdots + a_1\mathrm{Tr}(\zeta\alpha^{n-1}) \end{aligned}$$

which implies

$$z_n = a_m z_{n-m} + a_{m-1}z_{n-m+1} + \ldots + a_1 z_{n-1}$$

□

Lemma 2 *Let $\mathfrak{s} = \{z_n = z_n^{(1)} + z_n^{(2)} + \ldots + z_n^{(r)}\}_{n=0}^{\infty}$ be a sum of r linear recurring sequences whose generating polynomials $m^{(i)}(x)$ $i = 1, \ldots, r$ have the same degree m and are relatively primes. Then $\mathfrak{s}$ is a linear recurring sequence of order mr.*

PROOF. The generating polynomial of $\mathfrak{s}$ is the minimum common multiple of the generating polynomials $m^{(i)}(x)$ $i = 1, \ldots, r$, thus it is equal to their product by hypothesis and has degree mr.

□

Theorem 1 *Let the sequence*

$$\mathfrak{s} = \{z_n = \mathrm{Tr}(f(\alpha^n)\alpha^{-n2r})\}_{n=0}^{\infty}$$

be generated by the trace function, where $f(z^2)$ is a polynomial of degree $2r-2$ with every coefficient $\zeta_j \in \mathbb{F}_{2^m}$ different from zero, and $\alpha \in \mathbb{F}_{2^m}$ is a primitive element. If α^{2j-1} for $j = 1, \ldots, r$ are roots of distinct irreducible polynomials of degree m, then $\mathfrak{s}$ has linear complexity $L(\mathfrak{s}) = mr$, that is $\mathfrak{s}$ is produced by a linear recurrence of degree mr.

PROOF. By hypotheses regarding ζ_j and α^{2j-1}, which characterize BCH codes with designed minimum distance $2r+1$, we can write

$$z_n = \mathrm{Tr}(\sum_{j=1}^{r} \zeta_j \alpha^{n(2j-1-2r)}) = \sum_{j=1}^{r} \mathrm{Tr}(\zeta_j \alpha^{n(2j-1-2r)})$$

therefore z_n is a sum of r linear recurring sequences of order m with relatively prime generating polynomials, thus the conclusion follows by Lemmas 1 and 2.

□

Goppa codes

Goppa codes $(2^m-1, mt, d)$ can be generated similarly to BCH codes [8, Theorem 5, p.341] and are closely connected with generalized Reed-Solomon codes.
Let $G(z)$ be an irreducible Goppa polynomial with coefficients in the Galois field $\mathbb{F}_{2^m}$, and let $L = \{\gamma_1, \gamma_2, \ldots\} =\subseteq \mathbb{F}_{2^m}$ be the ordered location set, with $G(\gamma) \neq 0 \ \ \forall \gamma \in L$. Let α be a primitive element of $\mathbb{F}_{2^m}$, then a dual of a Goppa code has words

$$(\mathrm{Tr}(f(1)/G(1)), \mathrm{Tr}(f(\alpha)/G(\alpha)), \mathrm{Tr}(f(\alpha^2)/G(\alpha^2)), \ldots \\ \ldots, \mathrm{Tr}(f(\alpha^{2^m-2})/G(\alpha^{2^m-2})))$$

A codeword of $\mathcal{C}$ can be viewed as a period of a binary sequence. The linear complexity is given by the below theorem, which is a direct consequence of the following lemma.

Lemma 3 *Let $G(z)$ be a polynomial of degree r over $\mathbb{F}_2$ that is irreducible over $\mathbb{F}_{2^m}$, and let $\alpha \in \mathbb{F}_{2^m}$ be a primitive element. The linear complexity $L(\mathfrak{g})$ of the sequence $\mathfrak{g} = \{\frac{1}{G(\alpha^n)}\}_{n=0}^{\infty}$ is lower bounded by $\frac{2^m-1}{m}$.*

PROOF. The linear complexity of $\mathfrak{g}$ is given by the smallest M that allows us to write

$$\frac{1}{G(\alpha^n)} = \sum_{i=1}^{M} \alpha^{nk_i} \quad \forall n = 0 \dots 2^m - 2 \tag{2}$$

for a suitable choice of the set $k_1, k_2, \dots k_M \in \mathbb{Z}$. Let us observe that $\frac{1}{G(\alpha^{n_1 2^j})} = \left(\frac{1}{G(\alpha^{n_1})}\right)^{2^j}$ for every j less than m, therefore every set of m integers $n = n_1 2^j,\ j = 0, \dots, m-1$ results in a single condition, furthermore, $\frac{1}{G(\alpha^0)} = 1$ implies that M is odd. Therefore equation (2) can be seen as a system of $\frac{2^m-1}{m}$ syndrome equations that must correct $\frac{2^m-1}{m}$ errors in positions k_i, thus $M \geq \frac{2^m-1}{m}$.

$\square$

Theorem 2 *Let $f(z)$ be a polynomial of degree $r-1$ with every coefficient $\zeta_j \in \mathbb{F}_{2^m}$ not zero. The limited sequence $\mathfrak{s} = \{z_n = \mathrm{Tr}(f(\alpha^n)/G(\alpha^n))\}_{n=0}^{2^m-2}$ generated by the trace function has a linear complexity $L(\mathfrak{s})$ that is lower bounded by $\frac{2^m-1}{m}$.*

PROOF. Since the conditions of Lemma 3 hold, the linear complexity of $\mathfrak{s}$ is not less than the linear complexity of $\mathfrak{g}$.

$\square$

Clock-controlling and puncturing

In the block $[x(0), x(1), \dots, x(T-1)]$ of length equal to one period, any symbol can be identified by the corresponding state $\mathbf{u}(n) = \alpha^{\ell(n)}$ of the shift register. Therefore we write

$$[x(0), x(1), \dots, x(T-1)] = \\ = [x(\alpha^{\ell(0)}), x(\alpha^{\ell(1)}), \dots, x(\alpha^{\ell(T-1)})]$$

with the convention that if $\mathbf{u}(n) = \mathbf{0}$ then $\ell(n) = \infty$.
The effect of the clock-control is to exclude some states from the sequence $\mathfrak{S}$, hence to exclude some $x(\alpha^i)$ from the generated sequence, an operation that corresponds to puncturing the period of the sequence, and conversely to generating a sequence with a better LCP.

5. LCP OF CLOCK-CONTROLLED LFSR GENERATED SEQUENCES

A clever dissertation on the importance of Linear Complexity Profile (LPC) in stream generators is given in [2], and we refer to that analysis to show that clock-controlled shift registers generate sequences with a good LCP.

Let A be the $m \times m$ companion matrix of a primitive irreducible binary generator polynomial $g(x) = x^m + g_1 x^{m-1} + \dots + g_m$ of a clock-controlled LFSR that produces the binary sequence $x(n) = \mathbf{r}^T \mathbf{u}(n)$, where $\mathbf{r}$ is a fixed m-dimensional vector and $\mathbf{u}(n)$ is the LFSR state at step n.
We have already observed that $\mathbf{u}(n) = A^{n+\ell_B C_1(n)}\mathbf{u}(0)$, furthermore A admits the following "*singular value decomposition*"

$$A = \sum_{i=0}^{m-1} \sigma_i(\alpha)\sigma_i(\mathbf{a})\sigma_i(\mathbf{b}^T)$$

where

1. the root α of $g(x)$ is an eigenvalue of A with column eigenvector $\mathbf{a} = (1, \alpha, \alpha^2, \dots, \alpha^{m-1})^T$;
2. $\sigma_i(z) = z^{2^i}$ is the Frobenius automorphism that is defined to act componentwise on a vector;
3. let $\mathcal{U} = (\mathbf{a}, \sigma_1(\mathbf{a}), \dots, \sigma_{m-1}(\mathbf{a}))$ be the matrix of conjugate eigenvectors. Therefore $\det(\mathcal{U}) = 1$, because $\mathcal{U}$ is nonsingular and the columns of $\sigma_1(\mathcal{U})$ are a cyclic permutation of the columns of $\mathcal{U}$. Let

$$\mathcal{V} = \mathcal{U}^{-1} = \begin{pmatrix} \mathbf{b}^T \\ \sigma_1(\mathbf{b}^T) \\ \vdots \\ \sigma_{m-1}(\mathbf{b}^T) \end{pmatrix}$$

be the inverse matrix of $\mathcal{U}$, where the conjugate rows have been evidenced. It follows that $\mathbf{b}^T\mathbf{a} = 1$ and $\sigma_i(\mathbf{b}^T)\mathbf{a} = 0$ for any $i \neq 0$, hence the state of the shift register at step n can be written in the form

$$\mathbf{u}(n) = \sum_{i=0}^{m-1} \sigma_i(\alpha^{n+\ell_B C_1(n)})\sigma_i(\mathbf{a})\sigma_i(d_0)$$

where we have set $d_0 = \mathbf{b}^T\mathbf{u}(0)$.

Observing that

$$A^h = \sum_{i=0}^{m-1} \sigma_i(\alpha^h)\sigma_i(\mathbf{a})\sigma_i(\mathbf{b}^T)$$

the n-th term in the output sequence has the form

$$\begin{aligned} x(n) &= \sum_{i=0}^{m-1} \sigma_i(\alpha^{n+\ell_B C_1(n)})[\mathbf{r}^T\sigma_i(\mathbf{a})][\sigma_i(\mathbf{a}_R^T)\mathbf{u}(0)] \\ &= \mathrm{Tr}([\mathbf{r}^T\mathbf{a}]d_0\alpha^{n+\ell_B C_1(n)}) \end{aligned}$$

Assuming $n > m$, these equations show that the linear complexity profile $L(n)$ of the sequence $x(n)$ is the same as that of the sequence $\alpha^{n+\ell_B C_1(n)}\mathbf{a}_R^T\mathbf{u}(0)$, which is given by the smallest integer $L(n)$ for which we can find a set of exponents k_j such that the system

$$\alpha^{\ell_B C_1(n)} = \sum_{j=1}^{t} d_0^{-1} c_j \alpha^{t(k_j-1)} \quad \forall t \leq n \tag{3}$$

has a solution with $c_j = 0$ for any $j > L(n)$. However, given that the coefficient matrix $D = (\alpha^{t(k_i-1)})$ is a Cauchy-Vandermonde matrix, it follows that $d_0^{-1}c_j$ is zero if and only if its corresponding column in D is equal to the column of known terms. This is possible for only one column. Therefore, except when $C_1(n) = kn$ so that we can take $k_j = k$, the number of non-zero coefficients c_j is n. In conclusion, the linear complexity is $L(n) = n$.

REFERENCES

[1] S.W. Golomb, *Shift Register Sequences*, Aegean Park Press, Laguna Hills, 1982.

[2] R.A. Rueppel, *Analysis and Design of Stream Cipher*, Springer, New York, 1986.

[3] J.L. Massey, Shift-Register Synthesis and BCH decoding, *IEEE Trans. on Inform. Th.*, IT-15, 1969, pp.122-127.

[4] R. Lidl and H. Niederreiter, *Finite Fields*, Addison-Wesley, Reading, Mass., 1983.

[5] I. Wegener, *The Complexity of Boolean Functions*, Wiley, New York, 1987.

[6] C.W. Curtis and I. Reiner, *Representation Theory of Finite Groups and Associative Algebras*, Wiley, New York, 1962.

[7] D. Jungnickel, *Finite Fields, Structure and Arithmetics*, Wissenshaftsverlag, Mannheim, 1993.

[8] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North Holland, New York, 1977.

[9] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, 1997.

[10] D.E. Knuth, *The Art of Computer Programming,* Seminumerical algorithms, vol. II, Addison-Wesley, Reading Massachussetts, 1981.

[11] R.J. McEliece, *Finite Fields for Computer Scientists and Engineers*, Kluwer Academic Press, Boston, 1987.

[12] M. Elia, G. Morgari and O. Steila, "A Note on the Linear Complexity of Products of Linear Recurring Sequences", *International Conference on Discrete Mathematics and Allied Topics*, India, November 10-13, 1997.

[13] E.R. Berlekamp, "Goppa Codes," *IEEE Transactions on Information Theory*, vol. IT-19, n. 5, September 1973, pp. 590-592.

[14] M. Elia, E. Viterbo and G. Bertinetti, "On Decoding Binary Separable Goppa Codes with the Berlekamp-Massey algorithm," *Electronics Letters*, Vol. 35, No. 20, September 1999, pp. 1720-1721.

[15] V.D. Goppa, "A new class of linear error-correcting codes", *Probl. Peredach. Inform.*, vol. 6, pp. 24-30, Sept. 1970.

[16] K. Huber, "Note on decoding binary Goppa codes," *Electronics Letters*, Vol. 32, No. 2, January 1996, p. 102-103.

[17] C.J. Moreno and O. Moreno, "The Algebraic Decoding of Goppa Codes," *IEEE Trans. on Information Theory*, vol. IT-38, no. 4, pp. 203-207, July 1992.

[18] M. Van der Vlugt, "The True Dimension of Certain Binary Goppa Codes," *IEEE Transactions on Information Theory*, vol. IT-36, n. 2, March 1990, pp. 397-398.

A. MATRIX REPRESENTATION OF FINITE FIELDS

A finite field $GF(2^m)$ admits a faithful representation by $m \times m$ binary matrices [7]. Let α be a root of a binary irreducible polynomial $g(x) = x^m + e_1x^{m-1} + e_2x^{m-2} + \ldots + a_{m-1}x + a_m$ of degree m, which has companion matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \ldots & 1 \\ e_r & e_{r-1} & e_{r-2} & \ldots & e_1 \end{bmatrix}$$

Definition 1 *Let $\mathfrak{M}(m, \mathbb{Z}_2)$ be the ring of the $m \times m$ matrices with binary entries.*
The natural representation ρ of $GF(2^m)$ is an homomorphism $\rho : \ GF(2^m) \to \mathfrak{M}(m, \mathbb{Z}_2)$ with kernel the zero element $0 \in GF(\alpha)$. Therefore ρ is a linear one-to-one mapping, and the image of $GF(\alpha)$ is precisely a subfield of $\mathfrak{M}(m, \mathbb{Z}_2)$ consisting of matrices obtained as follows:

- *(a)* $\rho(\alpha) = A$.
- *(b)* $\rho(1) = I_r$ *and* $\rho(0) = O_r$, *where* I_r *and* O_r *denote the identity and the all-zero matrix in* $\mathfrak{M}(m, \mathbb{Z}_2)$ *respectively.*
- *(c) If* $x = a_0 + a_1\alpha + \ldots + a_{m-1}\alpha^{m-1}$, $a_i \in \mathbb{Z}_2$, *then* $\rho(x) = a_0 I_t + a_1 A + \ldots + a_{m-1}A^{m-1}$

Definition 2 *Two binary representations ρ and ν of the same field $GF(2^m)$ are called equivalent if there is a matrix U with entries in $GF(2^m)$, such that $\rho(x) = U\nu(x)U^{-1}$ for every $x \in GF(2^m)$.*

Definition 3 *A representation ρ of $GF(2^m)$ is said to be reducible over some extension field $\mathbb{F}$ if there is a matrix Q with entries in $\mathbb{F}$ such that $Q\rho(x)Q^{-1}$ $\forall x \in GF(2^m)$ is a block diagonal matrix; otherwise it is said to be absolutely irreducible.*
A binary matrix representation ρ of $GF(2^m)$ is said to be $\mathbb{Z}_2$-irreducible if no matrix Q exists such that $Q\rho(x)Q^{-1}$, $\forall x \in GF(2^m)$ is a block diagonal matrix with binary entries; otherwise it is said $\mathbb{Z}_2$-reducible.

Proposition 2 *The natural representation ρ of $GF(2^m)$, with α a root of $g(x)$, is $\mathbb{Z}_2$-irreducible, and is completely reducible (diagonalizable) over $GF(2^m)$.*

PROOF. If ρ is reducible over $\mathbb{Z}_2$, then the polynomial $g(x)$ necessarily factors over $\mathbb{Z}_2$. Therefore the irreducibility of $g(x)$ over $\mathbb{Z}_2$ implies the $\mathbb{Z}_2$-irreducibility of ρ.
Let $\mathbf{v} = [1, \alpha, \ldots, \alpha^{m-1}]^T$ be a column vector, therefore

$$\alpha\mathbf{v} = A\mathbf{v}$$

and the same relation holds for every conjugate of $\sigma_i(\alpha) = \alpha^{2^i}$, where σ_i is an automorphism of the Galois group of $g(x)$. It follows that, if U is a matrix with columns the conjugated columns of $\mathbf{v}$, then we have

$$\mathrm{diag}(\sigma_1(\alpha), \sigma_2(\alpha), \ldots, \sigma_m(\alpha)) = U^{-1}AU \qquad \square$$

Proper subfields are represented by matrices having peculiar properties as specified in the following theorem.

Theorem 3 *If $GF(2^m)$ has a proper subfield $GF(2^n)$, then $n|m$, and:*

> *Any natural m-dimensional representation ρ of $GF(2^m)$ is equivalent to a representation ν in which the matrices associated to elements of $GF(2^n)$ are the outer tensor product of the $n \times n$ matrices of a natural integral representation ψ of $GF(2^n)$ by the $m/n \times m/n$ identity matrix. The characteristic polynomial of $\rho(x)$, $\forall x \in GF(2^m)$, is a power of exponent m/n of the characteristic polynomial of $\psi(x)$.*

PROOF. Since $GF(2^m)$ is a normal extension of degree m/n of $GF(2^n)$, then $GF(2^m)$ admits a natural (m/n)-dimensional representation Φ over $GF(2^n)$. A representation ν of $GF(2^m)$ over $\mathbb{Z}_2$ is obtained by the compound with ψ, namely, the entries of the matrices of Φ are substituted with the matrices of ψ whose entries are in $\mathbb{Z}_2$. Clearly in ν the elements of $GF(2^n)$ consist of block diagonal matrices where the same block occurs m/n times. The two representations ρ and ν are isomorphic.

□

Central algebra

The central algebra $\mathcal{A}_\rho$ of $Im(\rho)$ is the ring of matrices B that commute with every matrix $\rho(x)$, that is

$$\mathcal{A}_\rho = \{B \ : \ B\rho(x) = \rho(x)B, \quad \forall x \in GF(2^m)$$

Certainly $\mathcal{A}_\rho$ includes $Im(\rho)$, moreover since the representation ρ of $GF(2^m)$ is irreducible, we have the following:

Theorem 4 *The center algebra $\mathcal{A}_\rho$ coincides with $Im(\rho)$.*

PROOF. By Schur lemma [6, p.181], $Im(\rho)$ is completely reducible in $GF(2^m)$, therefore a matrix Q with entries in $GF(2^m)$ exists that diagonalizes any matrix in $Im(\rho)$. In particular

$$QAQ^{-1} = \mathrm{diag}\left(\alpha, \alpha^2, \ldots, \alpha^{2^{m-1}}\right)$$

where it is seen that all entries are conjugated for the Frobenius automorphism $\sigma_i(\alpha) = \alpha^{2^i}$.
The relation $QAQ^{-1}QBQ^{-1} = QBQ^{-1}QAQ^{-1}$ implies that also any QBQ^{-1} with $B \in \mathcal{A}_\rho$ is a diagonal matrix with Frobenius conjugate elements. Thus any $B \in \mathcal{A}_\rho$ belongs to $Im(\rho)$.

□