Index Tables of Finite Fields and Modular Golomb Rulers

Ana Sălăgean, David Gardner, and Raphael Phan

Loughborough University, UK
Email: {A.M.Salagean, D.Gardner2, R.Phan}@lboro.ac.uk

Abstract. For a Galois field GF(2^n) defined by a primitive element α with minimal polynomial f, the index table contains in row i the coordinates of α^i in the polynomial basis $\alpha^{n-1}, \alpha^{n-2}, \ldots, \alpha, 1$. Each column i in this table equals the m-sequence with characteristic polynomial f, shifted cyclically by some offset h_i.
In this paper we show that the set of the n shifts h_i contains large subsets which are modular Golomb rulers modulo $2^n - 1$ (i.e. all the differences are different). Let D be the set of integers j such that the coefficient of x^j in f is non-zero. We prove that the set H_D of shifts corresponding to columns $j \in D$ can be partitioned into two subsets (the columns in the left half of the table and the ones in the right half) each of which is a modular Golomb ruler. Based on this result and on computational data, we conjecture that in fact the whole set H_D is a modular Golomb ruler. We give a polynomial time algorithm for deciding if given a subset of column positions, the corresponding shifts are a modular Golomb ruler. These results are applied to filter generators used in the design of stream ciphers. Golić recommends that in order to withstand his inversion attack, one of the design requirements should be that the inputs of the nonlinear filtering function are taken from positions of a Fibonacci LFSR which form a Golomb ruler. We propose using a Galois LFSR instead and selecting positions such that the corresponding shifts form a modular Golomb ruler. This would allow for a larger number of inputs to be selected (roughly $n/2$ rather than $\sqrt{2n}$) while still satisfying Golić's requirement.

1 Preliminaries

First we recall the definitions of linear recurrent sequences and m-sequences.

Definition 1. *An infinite sequence $\tilde{s} = s_0, s_1, \ldots$ with elements in a field K is called a* linear recurrent sequence *if there exists a relation of the form $s_{i+n} = c_{n-1}s_{i+n-1} + \cdots + c_1 s_{i+1} + c_0 s_i$ for all $i = 0, 1, \ldots$, where $c_0, c_1, \ldots, c_{n-1} \in K$ are constants. We associate to it a characteristic polynomial $f(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_1 x + c_0$. If n is minimal for the given sequence we call n the linear complexity of the sequence. A sequence which has a primitive polynomial as characteristic polynomial is called an* m-sequence.

Recall that a binary m-sequence of linear complexity n has period $2^n - 1$.

We now introduce a notation for (cyclic) shifts of sequences:

Definition 2. *Given a sequence $\tilde{s} = s_0, s_1, \ldots$, we denote by $\tilde{s} \ll k$ the sequence obtained by shifting $\tilde{s}$ by k positions to the left, i.e. the sequence $s_k, s_{k+1}, \ldots$.*

If $\tilde{s}$ is periodic with period N we denote by $\tilde{s} \gg k$ the sequence obtained by cyclicly shifting $\tilde{s}$ by k positions to the right, i.e. the sequence $s_{N-k}, s_{N-k+1}, \ldots, s_{N-1}, s_0, s_1, \ldots$.

Obviously $(\tilde{s} \gg k) = (\tilde{s} \ll (N-k))$.

Next we recall a few facts about the construction of a finite field with 2^n elements, denoted $\mathrm{GF}(2^n)$.

Throughout the paper we fix $f = x^n + c_{n-1}x^{n-1} + \cdots + c_1 x + c_0 \in \mathrm{GF}(2)[x]$ to be a primitive polynomial of degree n (hence $c_0 = 1$) and denote by α a root of f. We define $\mathrm{GF}(2^n)$ as $\mathrm{GF}(2)[x]/\langle f \rangle$, or equivalently as the algebraic extension field of $\mathrm{GF}(2)$ by α.

The elements of $\mathrm{GF}(2^n)$ can be represented in different ways; we are interested in the two most common representations: firstly we have the representation in the polynomial basis $\alpha^{n-1}, \alpha^{n-2}, \ldots, \alpha, 1$, whereby

$$\mathrm{GF}(2^n) = \{r_{n-1}\alpha^{n-1} + r_{n-2}\alpha^{n-2} + \cdots + r_1\alpha + r_0 | r_0, \ldots, r_{n-1} \in \mathrm{GF}(2)\}.$$

Secondly we have the representation as powers of the primitive root α, whereby

$$\mathrm{GF}(2^n) = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^n-2}\}.$$

Since the first representation is convenient for addition and the second is convenient for multiplication (and multiplicative inverse), implementations often use lookup tables for conversion between the two representations, also called log/antilog tables or index tables. When n is large however, such tables can no longer be computed/stored due to their exponential size.

Converting from a power of α to the polynomial basis representation is relatively easy (polynomial time). However the reverse problem (given $r_{n-1}, \ldots, r_0$ find i such that $\alpha^i = r_{n-1}\alpha^{n-1} + r_{n-2}\alpha^{n-2} + \cdots + r_1\alpha + r_0$) is difficult and it is known as the Discrete Logarithm Problem (DLP) in $\mathrm{GF}(2^n)$.

We will study the index table that gives the representation of $1, \alpha, \alpha^2, \ldots, \alpha^{2^n-2}$ in the polynomial basis. That is, if we denote

$$\alpha^i = r_i^{(n-1)}\alpha^{n-1} + r_i^{(n-2)}\alpha^{n-2} + \cdots + r_i^{(1)}\alpha + r_i^{(0)},$$

the index table is the $2^n - 1$ by n matrix whose rows are indexed from 0 to $2^n - 2$ and the i-th row is the vector $(r_i^{(n-1)}, r_i^{(n-2)}, \ldots, r_i^{(1)}, r_i^{(0)})$. Note that the rows of this table are precisely all the n-bit vectors except the all-zero one. We will denote column j by $\tilde{r}^{(j)}$ and it will be convenient to view it as a periodic sequence of period $2^n - 1$.

It is known, and not difficult to prove, that each sequence $\tilde{r}^{(j)}$ (being the image under a projection homomorphism of the sequence $1, \alpha, \alpha^2, \ldots$) has characteristic polynomial f. Since f is primitive, $\tilde{r}^{(j)}$ is an m-sequence. For different values of j we obtain different cyclic shifts of this same m-sequence. We will choose $\tilde{r}^{(n-1)}$ as a reference point.

Definition 3. *For $j = 0, \ldots, n-1$ we denote by h_j the integer modulo $2^n - 1$ such that $\tilde{r}^{(j)} = (\tilde{r}^{(n-1)} \gg h_j)$. We denote by H the set $\{h_{n-1}, h_{n-2}, \ldots, h_1, h_0\}$.*

Determining H seems difficult for large fields where the index table cannot be computed in full. This problem was considered by Blackburn in [1]. In [1, Definition 3] he defines a set $\sum(f)$ that would correspond to $H \cup \{h_i - h_j | h_i, h_j \in H\}$, and searches for suitable values in this set in order to increase the rate of output of m-sequences by interleaving. In the next section we will prove certain properties of the elements of H without explicitly computing them.

It will be convenient to use the trace representation for m-sequences:

Theorem 1. *[5, Theorem 6.21] The elements of an m-sequence $\tilde{s} = s_0, s_1, \ldots$ over* GF(2) *can be expressed as $s_i = Tr(a\alpha^i) = \sum_{j=0}^{n-1} a^{2^j} (\alpha^{2^j})^i$, where α is a primitive root of the primitive characteristic polynomial of s and $a \in \mathrm{GF}(2^n)$, $a \neq 0$, is a constant, uniquely determined by the first n elements of the sequence.*

Since we will work with a fixed primitive polynomial f, it is only the constant a in the theorem above that determines which of the $2^n - 1$ shifts of the m-sequence we are dealing with. It is therefore convenient to introduce the following notation:

Definition 4. *We define* $\mathrm{Seq}_\alpha(a)$ *(also denoted* $\mathrm{Seq}(a)$ *if α is clear from the context) as the sequence $\tilde{s}$ whose i-th element is represented by*

$$s_i = Tr(a\alpha^i) = \sum_{j=0}^{n-1} a^{2^j} (\alpha^{2^j})^i. \tag{1}$$

Seq is linear, i.e. for any $a, b \in \mathrm{GF}(2^n)$ and $c \in \mathrm{GF}(2)$ we have:

$$\mathrm{Seq}(a) + \mathrm{Seq}(b) = \mathrm{Seq}(a+b) \tag{2}$$

$$c\,\mathrm{Seq}(a) = \mathrm{Seq}(ca) \tag{3}$$

The effect of shifting on sequences $\mathrm{Seq}(a)$ can be described as follows:

Lemma 1. *Let $a, a_1, a_2 \in GF(2^n)^*$ and h an integer. Then:*
(i) $(\mathrm{Seq}(a) \ll h) = \mathrm{Seq}(a\alpha^h)$ *and* $(\mathrm{Seq}(a) \gg h) = \mathrm{Seq}(a\alpha^{-h})$
(ii) $\mathrm{Seq}(a_2) = (\mathrm{Seq}(a_1) \gg h)$ *where h is the discrete logarithm of $a_1 a_2^{-1}$.*

Proof. (i) The i-th element of $(\mathrm{Seq}(a) \ll h)$ is the $(i+h)-th$ element of $\mathrm{Seq}(a)$,

$$s_{i+h} = \sum_{j=0}^{n-1} a^{2^j} (\alpha^{2^j})^{i+h} = \sum_{j=0}^{n-1} (a\alpha^h)^{2^j} (\alpha^{2^j})^i$$

which is indeed the i-th element of the sequence $\mathrm{Seq}(a\alpha^h)$ as in (1).

(ii) Write $\mathrm{Seq}(a_2) = \mathrm{Seq}(a_1 a_2 a_1^{-1}) = \mathrm{Seq}(a_1 \alpha^{-h})$ and then use (i). □

The following notion appears in the literature in different equivalent forms and under different names: Golomb ruler, finite Sidon set, full positive difference set, etc.

Definition 5. *A* Golomb ruler *of order m is a set of integers $\{b_0, \ldots, b_{m-1}\}$ with $b_0 < b_1 < \cdots < b_{m-1}$, such that all the positive pairwise differences of elements are unique,i.e. $b_j - b_i \neq b_l - b_k$, for all $(i,j) \neq (k,l)$, $i < j$ and $k < l$.*

A modular Golomb ruler *modulo N is a set $\{b_0, \ldots, b_{m-1}\}$ of numbers modulo N such that all the pairwise differences of elements are unique modulo N, i.e. $(b_j - b_i) \bmod N \neq (b_l - b_k) \bmod N$, for all $(i,j) \neq (k,l)$.*

There is no general construction for optimal (modular) Golomb rulers (i.e. minimum $b_{m-1} - b_0$ for given order m); tables for the currently known optimal values are available see [4], the Online Encyclopedia of Integer Sequences and the references therein. The following is immediate:

Lemma 2. *Let $B = \{b_0, \ldots, b_{m-1}\}$ with $0 \leq b_0 < b_1 < \cdots < b_{m-1} < N$ be a Golomb ruler. If $b_{m-1} - b_0 < N/2$ then B is also a modular Golomb ruler modulo N.*

Proof. For all $i < j$, we consider a first set of differences as the differences $b_j - b_i$. These are all different because B is a Golomb ruler. Moreover, $b_j - b_i \leq b_{m-1} - b_0 < N/2$. The second set of differences $b_i - b_j = N - (b_j - b_i) > N/2$ are all different among themselves, and also different from the first set of differences. □

2 Modular Golomb rulers within the set of shifts of the index table of a Galois ring

In this section we show that certain non-trivial subsets of H (where H is defined in Definition 3) are modular Golomb rulers. Moreover, we show that for suitable choices of the primitive polynomial f these subsets contain about half the elements of H.

For a start, all h_j are different. (If we assumed there exist $h_i = h_j$ then in each row of the index table entries i and j are identical. However, this is not possible as the table contains as rows all the possible binary vectors except the all-zero one.) As an easy consequence $h_j - h_i \neq h_k - h_i$ for all $j \neq k$.

Lemma 3. *$\tilde{r}^{(0)} = (\tilde{r}^{(n-1)} \gg 1)$ and $\tilde{r}^{(j)} = ((\tilde{r}^{(j-1)} + c_j \tilde{r}^{(n-1)}) \gg 1)$ for $1 \leq j \leq n-1$.*

Proof.

$$\begin{aligned}
\alpha^{i+1} &= r_{i+1}^{(n-1)}\alpha^{n-1} + r_{i+1}^{(n-2)}\alpha^{n-2} + \cdots + r_{i+1}^{(1)}\alpha + r_{i+1}^{(0)} = \alpha\alpha^i \\
&= \alpha(r_i^{(n-1)}\alpha^{n-1} + r_i^{(n-2)}\alpha^{n-2} + \cdots + r_i^{(1)}\alpha + r_i^{(0)}) \\
&= r_i^{(n-1)}\alpha^n + r_i^{(n-2)}\alpha^{n-1} + \cdots + r_i^{(1)}\alpha^2 + r_i^{(0)}\alpha \\
&= r_i^{(n-1)}(c_{n-1}\alpha^{n-1} + \ldots + c_1\alpha + c_0) + r_i^{(n-2)}\alpha^{n-1} + \cdots + r_i^{(1)}\alpha^2 + r_i^{(0)}\alpha \\
&= (r_i^{(n-1)}c_{n-1} + r_i^{(n-2)})\alpha^{n-1} + \cdots + (r_i^{(n-1)}c_1 + r_i^{(0)})\alpha + (r_i^{(n-1)}c_0).
\end{aligned}$$

Since $\alpha^{n-1}, \alpha^{n-2}, \ldots, \alpha, 1$ is a vector space basis, we have $r_{i+1}^{(j)} = r_i^{(n-1)}c_j + r_i^{(j-1)}$ and $r_{i+1}^{(0)} = r_i^{(n-1)}c_0$. □

Corollary 1. *(i)* $h_{n-1} = 0, h_0 = 1$ *and* $h_j = h_{j-1} + 1$ *for all* j *for which* $c_j = 0$. *(ii) If* $f = x^n + x^j + 1$ *is a trinomial, then* $H = \{0, 2^n - 2, 2^n - 3, \ldots, 2^n - (n - j), j, j - 1, \ldots, 3, 2, 1\}$

Hence for determining H it suffices to determine those h_j for which $c_j \neq 0$.

Let a be such that $\tilde{r}^{(n-1)} = \mathrm{Seq}(a)$. The value of a can be computed from the initial terms of $\tilde{r}^{(n-1)}$ but this will not be necessary for our purposes.

Theorem 2. *Let* $z_j = c_{j+1} + c_{j+2}\alpha + \cdots + c_{n-1}\alpha^{n-j-2} + \alpha^{n-j-1}$. *Then:*
(i) $z_0, z_1, \ldots, z_{n-1}$ *form a vector space basis for* $\mathrm{GF}(2^n)$.
(ii) $\tilde{r}^{(j)} = \mathrm{Seq}(az_j)$, *i.e.* $\alpha^{-h_j} = z_j$ *for all* $j = 0, \ldots n - 1$.
(iii) $h_j - h_i$ *equals the discrete logarithm of* $z_i z_j^{-1}$.
(iv) If j *is such that* $c_j \neq 0$ *then* $h_j = h_{j-1} + 1 - h$ *where* h *equals the discrete logarithm of* $1 + z_{j-1}^{-1}$.

Proof. For (i), note that the z_j have different degrees. The proof of (ii) is by induction on j using Lemmas 1 and 3 as well as the linearity of Seq, i.e. equations (2) and (3). For (iii), write $\alpha^{h_j - h_i} = \alpha^{-h_i}\alpha^{h_j} = z_i z_j^{-1}$. Finally, (iv) is a particular case of (iii). □

Determining H is therefore equivalent to solving the particular instances of the DLP problem $\alpha^{-h_j} = z_j$, for $j = 0, 1, \ldots, n - 1$ or alternatively solving particular instances of the State-based DLP as defined by Giuliani and Gong in [2, Definition 7]. Namely, given the n initial terms of $\tilde{r}^{(j)}$, determine the starting position h_j where the n terms 0,0,...,0,1 appear in $\tilde{r}^{(j)}$. It is shown in [2, Theorem 3] that the State-based DLP is equivalent to the DLP.

Theorem 3. *Let* $D \subseteq \{0, 1, \ldots, n-1\}$ *be a set of indices and* $H_D = \{h_i | i \in D\}$ *be the set of corresponding values of shifts. The set* H_D *is a modular Golomb ruler (modulo* $2^n - 1$*) if and only if for all distinct pairs* (i, j), (k, l) *of elements in* D *with* $i < j$, $k < l$, $j - i \leq l - k$ *we have*

$$z_i z_j^{-1} \neq z_k z_l^{-1} \tag{4}$$

$$z_i z_j^{-1} \neq z_k^{-1} z_l \tag{5}$$

Proof. Use Theorem 2(iii) and Definition 5. □

Based on the theorem above, Algorithm 1 decides whether H_D is a modular Golomb ruler for a given D.

Theorem 4. *Algorithm 1 has a time complexity of* $\mathcal{O}(n^4)$ *and needs* $\mathcal{O}(n^3)$ *extra memory space.*

Proof. Computing the polynomial basis representation of $z_i z_j^{-1}$ and of $z_i^{-1} z_j$ takes $\mathcal{O}(n^2)$ steps. The list L has at most $n(n-1)$ elements of n bits each, i.e. a total of $\mathcal{O}(n^3)$ bits. With an appropriate data structure, we can maintain the elements of L in lexicographic order and we do binary search to find out if an element is in the list or to insert a new element. We would then need $\log(n^2) = 2 \log n$ list element comparisons, and each comparison takes n steps. Hence all operations inside the two nested **for** loops take $\mathcal{O}(n^2)$ steps. □

Algorithm 1 GolombRulerDecision(f, D)

Input: f a primitive polynomial of degree n; $D \subseteq \{0, 1, \ldots, n-1\}$.
Output: True/False signifying whether $\{h_j | j \in D\}$ is a modular Golomb ruler.
begin
Initialise L to the empty list
for $i = 0, 1, \ldots, n-1$ **do**
  for $j = i+1, \ldots, n-1$ **do**
    Compute the polynomial basis representation of $z_i z_j^{-1}$ and of $z_i^{-1} z_j$
    if ($z_i z_j^{-1}$ is in L) or ($z_i^{-1} z_j$ is in L) **then**
      return(False)
    else
      Insert $z_i z_j^{-1}$ and $z_i^{-1} z_j$ in L
    end if
  end for
end for
return(True)
end

For certain subsets of H we will be able to show that they are always modular Golomb rulers. Intuitively, runs of zero coefficients in f correspond to runs of consecutive integers in the corresponding shifts h_j by Corollary 1(i). In such regions of consecutive integers we can only choose very small subsets which are Golomb rulers. A much more promising source of Golomb ruler subsets comes from those h_j for which $c_j \neq 0$.

Next we will gather sufficient conditions for (4) and (5) to hold.

Lemma 4. *We use the notations of Theorem 3 and assume c_i, c_j, c_k, c_l are all non-zero.*
Each of the following conditions is sufficient for (4) to be satisfied:
(i) $j - i = l - k$
(ii) $i + l \leq n$
(iii) $j + k \geq n - 1$
Each of the following conditions is sufficient for (5) to be satisfied:
(iv) $j + l \leq n$
(v) $i + k \geq n - 1$.

Proof. We write $z_i = \alpha^{-(i+1)} v_i$ where $v_i = 1 + c_1\alpha + c_2\alpha^2 + \ldots + c_i\alpha^i$. We denote by next($i$) the smallest index $u > i$ such that $c_u \neq 0$. Note that next(i) $\leq j$.

The general idea of these proofs is that we assume for a contradiction that equality holds in (4) or in (5), respectively. We then simplify this equation to the point that only powers of α between $\alpha^0 = 1$ and α^{n-1} appear. Since this is a vector space basis of GF(2^n), an equality holds if and only if for all i the coefficients of the corresponding α^i are identical on the two sides of the equality. We then prove that this is not the case for our equality, obtaining thus a contradiction.

(i) Note that in this case we cannot have $i = k$, because $j - i = l - k$ would then imply $j = l$ and therefore $(i, j) = (k, l)$. Assuming equality in (4) and using

$j-i=l-k$, this simplifies to $v_k v_j = v_i v_l$. Writing $v_j = v_i + \alpha^{i+1}(c_{i+1} + c_{i+2}\alpha + \cdots + c_j\alpha^{j-i-1})$ and similarly for v_l the equality further simplifies to either

$$v_k(c_{i+1} + c_{i+2}\alpha + \cdots + c_j\alpha^{j-i-1}) = \alpha^{k-i}v_i(c_{k+1} + c_{k+2}\alpha + \cdots + c_l\alpha^{l-k-1})$$

for the case $i < k$, or

$$\alpha^{i-k}v_k(c_{i+1} + c_{i+2}\alpha + \cdots + c_j\alpha^{j-i-1}) = v_i(c_{k+1} + c_{k+2}\alpha + \cdots + c_l\alpha^{l-k-1})$$

for the case $i > k$. In the case of $i < k$, on the l.h.s. the smallest power of α is $\text{next}(i)$ and the highest is $l-1$ and on the r.h.s. the smallest power is $k - i + \text{next}(k)$ and the highest is $l-1$. Since all powers of α are below n, all the corresponding coefficients of the powers of α must coincide on the l.h.s and r.h.s. This implies $\text{next}(i) = k - i + \text{next}(k)$, i.e. $i + \text{next}(i) = k + \text{next}(k)$. However, one can see that this is a contradiction because $i < k$, which due to the way we defined next implies $\text{next}(i) \leq k < \text{next}(k)$. The case $i > k$ leads to a contradiction in a similar way.

(ii) We may assume $l - k > j - i$, as the case $l - k = j - i$ was covered by (i). Assuming equality in (4) we obtain $\alpha^{(l-k)-(j-i)}v_j v_k = v_l v_i$. On the l.h.s. the lowest power of α is $(l-k)-(j-i) > 0$ and the highest is $l+i$. On the r.h.s. the lowest is 0 and the highest is $l+i$. The highest powers on both sides cancel out, leaving only powers of at most $l+i-1 \leq n-1$. Since all powers of α are below n, all the corresponding coefficients of the powers of α must coincide on the l.h.s and r.h.s. However this cannot be the case as the lowest powers with a non-zero coefficient are different on the two sides.

(iii) Again we may assume $l-k > j-i$, as the case $l-k = j-i$ was covered by (i). Note $i+l > j+k \geq n-1$. Assuming equality in (4) gives $z_i z_l = z_j z_k$. The powers of α range from some integer ≥ 0 to $2(n-1)-(i+l) < n-1$ on the l.h.s. and from some integer ≥ 0 to $2(n-1)-(j+k) \leq n-1$ on the r.h.s.. That means the coefficients must be identical, hence $2(n-1)-(i+l) = 2(n-1)-(j+k)$. But that implies $l-k = j-i$, which is not true.

(iv) Assuming equality in (5) gives $v_j v_l = \alpha^{l-k+j-i}v_i v_k$. Again, the powers of α range on the l.h.s. from 0 to $j+l$ and on the r.h.s from $l-k+j-i > 0$ to $j+l$, with the highest ones canceling out and leaving powers of at most $j+l-1 \leq n-1$. The range needs to be the same on both sides. Contradiction.

(v) Note $j+l > i+k \geq n-1$. Assuming equality in (5) gives $z_j z_l = z_i z_k$. The powers of α range from some integer ≥ 0 to $2(n-1)-(j+l) < n-1$ on the l.h.s. and from some integer ≥ 0 to $2(n-1)-(i+k) \leq n-1$ on the r.h.s.. Therefore the coefficients must be identical, which is not true as $2(n-1)-(j+l) < 2(n-1)-(i+k)$. □

Theorem 5. *Let* $D = \{i | c_i \neq 0\}$ *and let* $D_1 = \{i \in D, i \leq \frac{n}{2}\}$ *and* $D_2 = \{i \in D, i \geq \frac{n-1}{2}\}$. *Then* $H_{D_1} = \{h_i | i \in D_1\}$ *and* $H_{D_2} = \{h_i | i \in D_2\}$ *are modular Golomb rulers (modulo* $2^n - 1$*).*

Proof. For H_{D_1} all indices satisfy conditions (ii) and (iv) in Lemma 4. For H_{D_2} all indices satisfy conditions (iii) and (v) in Lemma 4. □

Conjecture 1. $H_D = \{h_i | c_i \neq 0\}$ is a modular Golomb ruler (modulo $2^n - 1$).

In view of Lemma 4, the missing cases for proving the conjecture are: showing that (4) holds when $j + k < n - 1$ and it also holds when $n < i + l$; showing that (5) holds when $i + k < n - 1$ and also when $n < j + l$. The experiments in the following section support this conjecture. Moreover, they allow us to state:

Proposition 1. *For all primitive polynomials f of degree 23 or less, $H_D = \{h_i | c_i \neq 0\}$ is a modular Golomb ruler (modulo $2^n - 1$).*

Finally, note that these results mean that H can have very large subsets which are modular Golomb rulers. One of the subsets in Theorem 5 will have at least $\lceil (\mathrm{wt}(f)-1)/2 \rceil$ elements, where $\mathrm{wt}(f)$ is the Hamming weight of f (number of non-zero coefficients). If Conjecture 1 is true for a particular f (and this can be checked by Algorithm 1), the subset obtained is even larger, namely it has $\mathrm{wt}(f) - 1$ elements.

For many, but not all n, there exists a primitive polynomial of weight n for n odd or of weight $n-1$ for n even. It seems likely that for all n there are primitive polynomials of weight close to n, and therefore H contains in these cases a modular Golomb ruler subset consisting of almost the whole H (if Conjecture 1 is true). Moreover, it seems likely that for any n there are primitive polynomials f for which all or almost all coefficients in the lower half of f are non-zero, and therefore H contains in these cases a modular Golomb ruler subset consisting of half or almost half of the elements of H (by Theorem 5, so regardless whether Conjecture 1 is true).

3 Experiments

Brute force experimentation was performed on all Galois fields $\mathrm{GF}(2^n)$ with n from 2 to 23, examining all the different primitive polynomials for each n. In each case the full index table was produced, and the shifts $h_{n-1}, h_{n-2}, \ldots, h_1, h_0$ were computed by direct examination of the table. Some examples are described in Table 1, with the primitive polynomial f represented as $1c_{n-1}c_{n-2}\ldots c_1 1$. It was then verified (using Definition 5) that removing those h_j for which $c_j = 0$ (shown in brackets in Table 1) leaves indeed a subset which is a modular Golomb ruler. Thus it was verified that Conjecture 1 holds for all primitive polynomials up to degree $n = 23$. For $24 \leq n \leq 29$ we ran Algorithm 1 for all primitive polynomials f with $\mathrm{wt}(f) \geq n - 1$ and again Conjecture 1 was verified.

4 An Application to Galois LFSRs and filter generators

Linear recurrent sequences are often generated in practice by hardware devices called Linear Feedback Shift Registers (LFSR). There are two common types of LFSR, usually called the Fibonacci LFSR and the Galois LFSR. We recall these notions here. The registers of a Fibonacci LFSR of length n will be denoted by $Q_0, Q_1, \ldots Q_{n-1}$. The content of register Q_j at time i will be denoted $q_i^{(j)}$ and

Table 1. A selection of primitive polynomials f and the corresponding shifts H.

$n = 7,\ f = 11111101,\ \mathrm{wt}(f) = 7,$ $H = \{0, 18, 119, 54, 39, (2), 1\}$
$n = 9,\ f = 1111000111,\ \mathrm{wt}(f) = 7,$ $H = \{0, 326, 461, (467), (466), (465), 464, 328, 1\}$
$n = 15,\ f = 1100000111100111,\ \mathrm{wt}(f) = 9,$ $H = \{0, (3971), (3970), (3969), (3968), (3967), 3966,$ $30091, 12457, 28329, (24624), (24623), 24622, 3973, 1\}$
$n = 21,\ f = 101010101110110001101,\ \mathrm{wt}(f) = 13,$ $H = \{(0), 2097150, (1796558), 1796557, (1333708),$ $1333707, (1195372), 1195371, 1508706, 363026, 820032,$ $(1536625), 1536624, 543838, (134466), (134465), (134464),$ $134463, 1796561, (2), 1\}$
$n = 23,\ f = 11111110111111111111111,\ \mathrm{wt}(f) = 23,$ $H = \{0, 873419, 3430060, 2620257, 1534122, 7733539,$ $3311431, (6113933), 6113932, 7496295, 3308273, 7951902,$ $226119, 3941673, 4712702, 6113941, 3311438, 7733545,$ $1534127, 2620261, 3430063, 873421, 1\}$

the contents of all the registers at time i are called the state at time i. The initial state is the state at time 0. The sequence $\tilde{q}^{(j)}$ consists of the values of register Q_j in time, i.e. $q_0^{(j)}, q_1^{(j)}, \ldots$. Similarly for a Galois LFSR we denote the registers by $R_{n-1}, R_{n-2}, \ldots, R_0$ and the contents of the register R_j in time by $\tilde{r}^{(j)}$.

Definition 6. *A Fibonacci LFSR of length n (see Fig. 1) with characteristic polynomial* $f(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_1 x + c_0$ *will update itself at each clock interval i according to the following*

$$q_{i+1}^{(j)} = \begin{cases} c_{n-1}q_i^{(n-1)} + \cdots + c_1 q_i^{(1)} + c_0 q_i^{(0)} & \text{if } j = n-1 \\ q_i^{(j+1)} & \text{otherwise.} \end{cases}$$

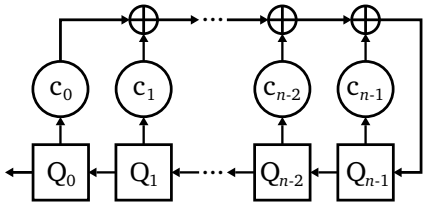

Fig. 1. A Fibonacci style LFSR

Definition 7. *A Galois LFSR of length n (see Fig. 2) with characteristic polynomial $f(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_1x + c_0$ will update itself at each clock interval i according to the following*

$$r_{i+1}^{(j)} = \begin{cases} c_0 r_i^{(n-1)} & \text{if } j = 0 \\ r_i^{(j-1)} + c_j r_i^{(n-1)} & \text{otherwise.} \end{cases}$$

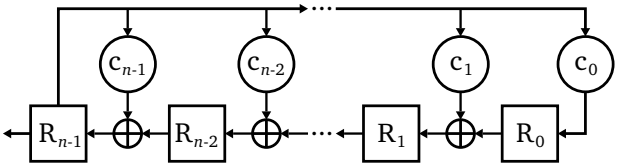

Fig. 2. A Galois style LFSR

The output of the Fibonacci LFSR is taken from register Q_0, i.e. equals $\tilde{q}^{(0)}$; the output of the Galois LFSR is taken from register R_{n-1}, i.e. equals $\tilde{r}^{(n-1)}$. It is known that a Fibonacci LFSR and a Galois LFSR with the same characteristic polynomial will produce the same output sequence provided the initial states are suitably chosen. We now fix the characteristic polynomial f to be the same primitive polynomial in both LFSRs, so both produce the same m-sequence.

In the Fibonacci LFSR each sequence representing the content of a register is equal to the neighbouring sequence shifted by one position. More precisely, $\tilde{q}^{(j)} = (\tilde{q}^{(j-1)} \lll 1)$. Taking the output sequence $\tilde{q}^{(0)}$ as reference, $\tilde{q}^{(j)} = (\tilde{q}^{(0)} \lll j) = (\tilde{q}^{(0)} \ggg (2^n - 1 - j))$.

For a Galois LFSR with a primitive polynomial f which has a primitive root α, the state $(r_i^{(n-1)}, r_i^{(n-2)}, \ldots, r_i^{(0)})$ at time i can be interpreted as the coefficients of the element $r_i^{(n-1)}\alpha^{n-1} + r_i^{(n-2)}\alpha^{n-2} + \cdots + r_i^{(0)}$ of $\mathrm{GF}(2^n)$. Then the state at time i will be α^{i+k} where k is such that α^k corresponds to the initial state. We can see now that each sequence $\tilde{r}^{(j)}$ coincides with the sequence $\tilde{r}^{(j)}$ defined in Section 1, shifted by k positions to the left. We are only interested in the relative shifts of different $\tilde{r}^{(j)}$, hence the shifts by k will cancel out. Taking the output sequence $\tilde{r}^{(n-1)}$ as reference point, the other sequences $\tilde{r}^{(j)}$ can be obtained by shifting $\tilde{r}^{(n-1)}$ to the right by h_j positions, where h_j is as defined in Definition 3.

For designing stream ciphers, one of the classical constructions for the keystream generator is the filter generator (see Fig. 3). It consists of a binary LFSR (usually Fibonacci LFSR) generating an m-sequence of period $2^n - 1$ and a boolean function $g : \mathrm{GF}(2)^k \to \mathrm{GF}(2)$ with $k \leq n$, called a non-linear filtering function. The output of the generator is obtained by applying the function g to k selected registers of the LFSR, say $j_1, j_2, \ldots, j_k$. Hence the output at time i equals $g(q_i^{(j_1)}, q_i^{(j_2)}, \ldots, q_i^{(j_k)})$.

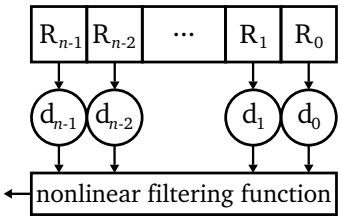

Fig. 3. An LFSR fed NFF

There is a large number of results concerning the recommended choice of the function g and the "tapped" positions $j_1, j_2, \ldots, j_k$ in order to avoid various cryptanalysis attacks. Here we are interested in the results of Golić [3]. In [3, Theorem 2], Golić gives a sufficient condition for a non-linear filtering function g to produce a purely random output provided its inputs come from a purely random sequence $\tilde{z}$ in such a way that the output at time i equals $g(z_{i-j_1}, z_{i-j_2}, \ldots, z_{i-j_k})$ for fixed tapping positions $j_1, \ldots, j_k$. In our notation, the input of g at time i consists of the i-th elements of the sequences $(\tilde{z} \gg j_1), (\tilde{z} \gg j_2), \ldots, (\tilde{z} \gg j_k)$. The sufficient condition is that g is linear in the first or last variable. Golić conjectured the condition was also necessary, see [6] for further results on this conjecture. Based on this result, Golić introduces an inversion attack for generators which use a Fibonacci LFSR and tapping position $j_1, \ldots, j_k$ together with a non-linear filtering function which is linear in the first or last variable. He recommends that for withstanding this attack one design criterion is that the tapped positions of the Fibonacci LFSR should form a full positive difference set (Golomb ruler). Note that since the tapped positions are in the range 0 to $n-1$, they also form a modular Golomb ruler modulo $2^n - 1$, by Lemma 2, as $n - 1 < (2^n - 1)/2$ for all $n \geq 2$.

Golić's results still apply if we use an m-sequence of complexity n and we buffer t terms for some $t \geq n$; we can then tap any positions $j_1, j_2, \ldots, j_k$ provided $\max_{u=1,\ldots,k}(j_u) - \min_{u=1,\ldots,k}(j_u) \leq t$. We suspect that in that case Golić's design criterion would need to be enhanced, requiring that the tapped positions be a modular Golomb ruler modulo $2^n - 1$ (the period of the m-sequence) rather than simply a Golomb ruler. The two are no longer equivalent if the range $\max_{u=1,\ldots,k}(j_u) - \min_{u=1,\ldots,k}(j_u)$ exceeds $2^{n-1} - 1$. Buffering $t > n$ terms would allow a larger number of positions to be tapped while still satisfying Golić's design criterion. This would come at the cost of extra storage.

We propose constructing a filter generator that uses a Galois LFSR with a dense primitive polynomial f. We then select positions $D = \{i_1, i_2, \ldots, i_k\} \subseteq \{0, 1, \ldots, n-1\}$ as inputs to the filtering function in such a way that $\{h_{i_1}, h_{i_2}, \ldots, h_{i_k}\}$ is a modular Golomb ruler. The filter generator thus constructed would be equivalent to tapping positions $j_1 = h_{i_1}, j_2 = h_{i_2}, \ldots, j_k = h_{i_k}$ of a buffered section of length $t = \min_{l=1,\ldots,k}(\max_{u=1,\ldots,k}((j_l - j_u) \bmod (2^n - 1))$ of the m-sequence, with the advantage that we do not need to actually buffer such a long section, we are only using the n memory registers of the Galois

LFSR. This construction would satisfy Golić's design criterion. It remains to be seen whether it would be susceptible to other forms of attack.

According to the discussion at the end of Section 2, we can choose $D = \{i|c_i \neq 0\}$ and check whether Conjecture 1 is true in this case by running Algorithm 1. If the answer is positive, we have $k = \mathrm{wt}(f) - 1$, which can be very close to n for suitably chosen f. If Algorithm 1 returns a negative result, we can still chose $D = \{i|c_i \neq 0, i \leq n/2\}$ and H_D is guaranteed to be a modular Golomb ruler by Theorem 5. For suitably chosen f we can then have k equal, or lower but very close to $\lfloor n/2 \rfloor + 1$. If we had to choose inputs from a Fibonacci LFSR of length n so that they are a Golomb ruler, the well known bound $n \geq k(k-1)/2$ would mean $k < \sqrt{2n} + 1$, hence a much smaller number of inputs are available. Equivalently, if we required some fixed number k of inputs, we would need a much larger length n for the Fibonacci LFSR, namely more than $k(k-1)/2$ compared to approximately $2k$ for the Galois LFSR. The following example illustrates this:

Example 1. The first example in Table 1, after removing the elements in brackets, produces a modular Golomb ruler of order $k = 6$. A Fibonacci LFSR of same length $n = 7$ would allow us to produce a Golomb ruler (which by Lemma 2 would also be a modular Golomb ruler modulo $2^n - 1$) of only $k = 4$ elements. For $k = 6$ elements we would need a Fibonacci LFSR of length $n = 17$ (see [4]).

The last example in Table 1 is a Galois LFSR of length $n = 23$ and after removing the elements in brackets, produces a modular Golomb ruler of order $k = 22$. A Fibonacci LFSR of same length $n = 23$ will allow us to produce a Golomb ruler (which by Lemma 2 would also be a modular Golomb ruler modulo $2^n - 1$) of order only $k = 6$. For order $k = 22$ we would need a Fibonacci LFSR of length $n = 356$ (see [4]).

Acknowledgements We would like to thank Simon Blackburn for a useful discussion regarding his paper [1].

References

1. Blackburn, S.R.: Increasing the Rate of Output of m-Sequences. Information Processing Letters 51, 73–77 (1994)
2. Giuliani, K. and Gong, G.: New LFSR-Based Cryptosystems and the Trace Discrete Log Problem (trace-DLP). In: Helleseth, T., Sarwate, D., Song, H., Yang, K. (eds.) Sequences and Their Applications - SETA 2004. LNCS, vol. 3486, pp. 298–312. Springer, Berlin / Heidelberg (2005)
3. Golić, J.D.: On the Security of Nonlinear Filter Generators. In: Gollmann, D. (ed.) Fast Software Encryption 1996. LNCS, vol. 1039, pp. 173–188. Springer, Berlin / Heidelberg (1996)
4. Graham, R.L. and Sloane, N.J.A.: On Additive Bases and Harmonious Graphs. Siam Journal on Algebraic and Discrete Methods 1, 382–404 (1980).
5. Lidl, R. and Niederreiter, H.: Introduction to Finite Fields and Their Applications. Cambridge University Press (1994)
6. Smyshlyaev, S.V.: Perfectly Balanced Boolean Functions and Golić Conjecture. Journal of Cryptology. 1–20 (2011)