Feedback with Carry Shift Registers over Finite Fields (Extended Abstract)

Andrew Klapper*

Dept. of Computer Science
763H Anderson Hall
University of Kentucky, Lexington
KY 40506-0046 USA
klapper@cs.uky.edu.

1 Introduction

The ideal cryptosystem would be one that can be proved secure against all possible cryptanalytic attacks (at least computationally secure). In practice, of course, we settle for cryptosystems that are secure against all *known* attacks (and in some cases merely rely on intuition rather than proof to believe they are secure). This is particularly true of private key systems. Thus the history of research on stream ciphers repeats a cycle in which a new system is developed, and perhaps proved secure against many existing attacks, then a new method is developed for attacking the new system. Future systems must be secure against the new attack. If the new attack is highly specialized, this tends to be easy – random unrelated methods of encryption are likely to be secure against the attack, or the attack may simply make no sense outside the context of the cryptosystem it was designed for. In some cases, however, a general purpose attack is developed that can potentially be used against a large class of cryptosystems. Furthermore, in some cases such attacks come with numeric measures of resistence to the attack. Such is the case, for example with the Berlekamp-Massey algorithm and linear span. All sequences used in stream ciphers must have large linear spans in order to resist the Berlekamp-Massey attack. This algorithm is based on the idea of synthesizing a linear feedback shift register (LFSR) that generates a given sequence, given a small number of bits of the sequence. Essential to make the algorithm work is the existence of an algebraic framework for the analysis of LFSR sequences, based on power series and polynomials over $GF(2)$.

In the Cambridge Algorithms Workshop in 1993, I described joint work with Mark Goresky in which we developed a new type of feedback register, feedback

* Project sponsored in part by the National Security Agency under Grant Number MDA904-91-H-0012. The United States Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation hereon.

with carry shift registers (or FCSRs) [5]. These registers are equipped with an algebraic framework for analysis, analogous to that of LFSRs, but based on the 2-adic number rather than power series over $GF(2)$.

In that paper, we described the basic algebraic properties of FCSRs. As in the case of FCSRs, one can ask whether there is an algorithm which, given part of a binary sequence $\mathbf{a}$, synthesizes a (minimal length) FCSR that generates $\mathbf{a}$. We showed that the existence of such an algorithm implies that it is possible to crack Massey and Ruepell's summation combiner. We further argued that the *2-adic span*, the length of the smallest FCSR that generates a given sequence, is thus an important measure of security. A sequence must have large 2-adic span in order to be secure (though this of course does not guarantee security). At the time of the Cambridge workshop we believed that a variant of the Berlekamp-Massey algorithm for approximating rational numbers (due to Mandelbaum [6]) could be adapted to the case of LFSRs. This has proved to not work. However, we have since developed a provably correct analogue of the Berlekamp-Massey algorithm, based on De Weger's lattice theoretic approach to rational approximation [8]. We describe the algorithm here, but details of the proof of correctness and analysis will appear elsewhere.

We have further developed generalizations of FCSRs by replacing the 2-adic numbers by ramified extensions [2]. That is, by adjoining π, a real dth root of 2. We showed that these registers have a similar algebraic structure to that of FCSRs, and we thus get a security measure for each positive d. (Although it seems that the larger d is, the computationally harder, and hence less threatening, is an attack based on these registers.) Furthermore, we have recently shown that our rational approximation algorithm for FCSRs works at least in the case $d = 2$.

From the point of view of wanting to build more secure systems, we are thus left with the question of how we can generate sequences which resist these attacks. That is, sequences with large 2-adic span, and even large π-adic span, where π is a dth root of 2 with d small. We would further like to do so without sacrificing other measures of security.

We do not yet have an answer to this question, but the purpose of this paper is to introduce new feedback register based tools that may allow us to build such secure sequences. One method that has been used to increase linear span is to take a LFSR over an extension $GF(p^n)$ of $GF(p)$, p prime, and apply a nonlinear "feedforward" function to its output to obtain a binary sequence. When $p \neq 2$, these sequences can have very large linear spans [1], although they are vulnerable for other reasons [4]. When $p = 2$, their linear spans are only moderately larger than LFSR sequences [3]. We describe here an FCSR analogue of LFSRs over nonprime finite fields. Hopefully such registers can be used to build sequences that have large 2-adic span.

In this abstract we describe the algebraic basis for these registers; the construction of the registers; various algebraic properties of the sequences they generate; and conditions under which our rational approximation algorithm can be generalized. For each class of generalized FCSR for which the rational approximation algorithm works, we obtain a new cryptographic security measure.

2 Definitions

The constructions of both LFSRs and FCSRs over a field K can be based on the following algebraic machinery: A ring R with a valuation such that the maximal ideal I of the valuation is principal, $I = (\pi)$ and such that $R/I = K$. There must be a set $S \subseteq R$ that maps bijectively to K under reduction modulo I, and we must be able to write every element of R as a difference $x - y$ where x and y are finite sums of powers of π with coefficients in S. We denote by P the set of finite sums of powers of π with coefficients in S. Infinite sequences over K can then be identified with infinite sequences over S, which can be identified with power series in π with coefficients in S, which can be identified with elements of the completion R_π of R at the valuation. Every element of R_π can be identified with such a power series. The feedback register is then constructed to carry out division in R, producing such an element of R_π.

In the case of LFSRs, $R = K[x]$, the power series ring in one variable over K, $I = (x)$, and $S = K$. In the case of FCSRs, $R = \mathbf{Z}$, the integers, $I = (2)$ (or, more generally, any prime ideal), and $S = \{0, 1\}$. In the case of FCSRs over a ramified extension of $\mathbf{Z}$, if π is a real dth root of 2, then $R = \mathbf{Z}[\pi]$, $I = (\pi)$, and $S = \{0, 1\}$. It should be noted that in the first two cases R is a Euclidean domain (that is, we can carry out "division with remainder"), but in the third case, R is only a Euclidean domain for some values of d, and the question of which values of d give Euclidean domains is a quite subtle one. This turns out to have an impact on what can be done with these registers. In particular, our rational approximation algorithm only works in a Euclidean domain.

Definition 1. Let R, $I = (\pi)$, and S be as above. A *feedback with carry shift register over* R, I, S, or simply an R-FCSR, of length r, is specified by r elements of S, $q_1, \cdots, q_r$ (which can be identified with elements of $K = R/I$). The register consists of r cells for storing elements of S, some additional memory for storing a "carry", and, if the contents of the register are $(a_{n-1}, \cdots, a_{n-r})$ and the memory is m_{n-1}, circuitry for implementing the following operations:

A1. Form the number $\sigma_n = \sum_{k=1}^{r} q_k a_{n-k} + m$.
A2. Shift the contents one step to the right, outputting the rightmost element a_{n-r}.
A3. Let $\sigma_n = a_n + \pi m_n$, with $a_n \in S$ (σ_n can always be written this way). Place a_n into the leftmost cell of the shift register. Replace the memory m_{n-1} by m_n.

If K is finite, it is straightforward to design circuitry to implement these operations, though only practical if K is small. The memory, an arbitrary element of R, can be represented as a finite set of elements of S by writing it as a difference $x - y$, with $x, y \in P$. We refer to $q = \sum_{i=1}^{r} q_i 2^i - 1$ as the *connection number* because it is the analog to the connection polynomial in the usual theory of

LFSRs. Any periodic sequence of elements of S may be generated by such a FCSR,

In order to define FCSRs over extensions of $GF(2)$, it is necessary to find suitable R, I, and S. In this abstract we consider only rings in which the ideal (2) remains a prime ideal (extensions that are unramified at 2), and take $I = (2)$. In order for R to reduce to $K = GF(2^n)$, R must contain an element β that reduces modulo 2 to a primitive element of K. Thus the minimal polynomial $f(x)$ of β must reduce modulo 2 to a primitive polynomial over K. We assume that $R = \mathbf{Z}[\beta]$. Let $\bar{\beta}$ be the reduction of β modulo 2. Then $\bar{\beta}$ is primitive, so $1, \bar{\beta}, \bar{\beta}^2, \cdots, \bar{\beta}^{n-1}$ is a basis for K over $GF(2)$. Thus the set of linear combinations of $1, \beta, \beta^2, \cdots, \beta^{n-1}$ with coefficients in $\{0, 1\}$ maps bijectively to K modulo 2. We take this set as S. For example, if $n = 2$ then we can take $f(x) = x^2 - x - 1$. In the next section we assume R, I, and S are defined in this manner. As it turns out, the amount of auxiliary memory needed for a register tends to be smallest when the b_i are small, so we would like to take $b_i \in \{0, \pm 1\}$. However, other properties of the registers (such as the convergence rate of the rational approximation algorithm) may be superior for other choices of the b_i.

3 Properties of R-FCSRs

Many of the algebraic properties of FCSRs also hold for R-FCSRs. There are five different ways to view an infinite, eventually periodic sequence over $GF(2^n)$:

1. As a sequence $\mathbf{a} = a_0, a_1, \cdots, a_i \in GF(2^n)$.
2. As a sequence $\mathbf{a} = a_0, a_1, \cdots, a_i \in S$.
3. As an element α of the completion R_2 of R at 2.
4. As an R-rational number $p/q \in K$.
5. As the output stream of an R-FCSR.

Representations (1) and (2) are identified by reducing S modulo 2. Representations (2) and (3) are identified by associating the binary sequence $\mathbf{a}$ with the coefficients in the formal power series expression for α. The translation between representations (3) and (4) is essentially the same as the identification between real numbers whose decimal expansions are eventually periodic and rational numbers. To translate between representations (4) and (5) we have the following.

Theorem 2. *The output, $\mathbf{a}$, of an R-FCSR with connection number q, initial memory value m_{r-1}, and initial loading $a_{r-1}, a_{r-2}, \ldots, a_1, a_0$, is the bit sequence of the 2-adic representation of an R-rational number*

$$\alpha = \frac{\sum_{i=0}^{r-1} \sum_{j=0}^{r-i-1} q_i 2^i a_j 2^j - m_{r-1} 2^r}{q}. \tag{1}$$

Thus the denominator of α is equal to the connection integer q of the shift register.

Corollary 3. *Adding b to the memory adds $-b2^r/q$ to the output.*

The converse of Theorem 2, that every R-rational number p/q can be realized as the output of an R-FCSR, is true as well. To see this, we show how to construct the initial loading of an R-FCSR for certain p, and then use Corollary 3 to obtain initial loadings for other FCSR. Let $p = \sum_{i=0}^{r-1} p_i 2^i$ with $p_i \in S \cup -S$. Every element of R differs from some such a p by a multiple of 2^r. Thus if we can construct initial loadings for p/q with p of this type, then we can construct initial loadings for all p/q.

Theorem 4. *Given a connection number $q = \sum_{i=0}^{r} q_i 2^i$ with $q_0 = -1$ and $q_1, \cdots, q_r \in S$, and $p = \sum_{i=0}^{r-1} p_i 2^i$ with $p_i \in S \cup -S$, define $a_0, \cdots, a_{r-1}$ and m_{r-1} by the following procedure:*

B1. *Set $m_{-1} = 0$ and $\sigma_0 = 0$.*
B2. *For each $i = 0, 1, \ldots, r-1$ compute the following numbers:*

$$\sigma_i = \sum_{k=0}^{i-1} q_{i-k} a_k + m_{i-1} - p_i \in R.$$

Write

$$\sigma_i = a_i + 2m_i,$$

where $a_i \in S$, $m_i \in R$, and the empty sum in σ_0 is interpreted as zero.

If $(a_{r-1}, a_{r-2}, \ldots, a_1, a_0)$ is used as the initial loading, m_{r-1} is used as the initial memory in an R-FCSR with connection number q, then the output sequence will correspond to the R-rational number p/q.

Note that the memory may not be in P (the set of elements in R which are finite sums of powers of 2 with coefficients in S). However, it can always be represented as a difference of such elements since this is true of every element of R. Moreover, if $p \in -P$, i.e., if all the p_i are in $-S$, and $f(x) = x^n - \sum_{i=0}^{n-1} b_i x^i$ with $b_i \geq 0$, then $m_{r-1} \in P$.

It is natural to ask how large an auxiliary memory is needed. In the case where $p \in -P$, we have the following.

Theorem 5. *Suppose that the coefficients b_i in the polynomial $f(x) = x^n - \sum_{i=0}^{n-1} b_i x^i$ are all either 0 or 1. Let q be expressed is a polynomial in α, $q + 1 = \sum_{i=0}^{r-1} t_i \beta^i$, with $t_i \in \mathbf{Z}$. With the initial value constructed as in Theorem 4, the number of bits M needed for the initial memory value is bounded by*

$$M \leq n \log(\sum_{i=0}^{n-1} 2^i \mathrm{wt}(t_i) \leq n^2 + n \log(\max(\mathrm{wt}(t_i))),$$

where $\mathrm{wt}(t_i)$ is the Hamming weight of the binary expansion of t_i. This bound continues to hold for all later values of the memory.

For other choices of the b_i, a similar but higher bound can be given.

4 Rational Approximation and Security Measures

The role to be played by R-FCSRs in cryptography is two fold. First, as discussed in the introduction, they are potential tools for building sequences that have large 2-adic span, and thus that are secure against 2-adic rational approximation (Berlekamp-Massey) algorithms. However, there is also a possibility that rational approximation can be carried out over R – we would call this R-adic rational approximation – and that we can thus carry out this sort of attack on sequences over $GF(2^n)$.

As it turns out, the ingredients needed to generalize our 2-adic rational approximation algorithm appear to be quite restrictive. R must be a Euclidean domain, with a little extra. Specifically, we need a *norm function* $N : K-\{0\} \rightarrow \mathbf{Q}^+$ ($\mathbf{Q}^+$ denotes the positive rational numbers), that satisfies the following:

a. For all $a \in R$, $N(a) \in \mathbf{N}$ ($\mathbf{N}$ denotes the natural numbers).
b. For all $a, b \in K$, we have $N(ab) = N(a)N(b)$.
c. For all $a, b \in R$, there exist $q, p \in R$ so that $a = qb + p$ and $N(p) < N(b)$.

In addition, in order to ensure the algorithm converges rapidly, we need

d. There is a function $f : \mathbf{N} \rightarrow \mathbf{N}$ so that if $a \equiv b \bmod \pi^{f(k)}$, $N(a) < 2^k$, and $N(b) < 2^k$, then $a = b$.

For any pair of elements p and q of R, define

$$\Phi(p, q) = \max(N(p), N(q)).$$

Assume we have consecutive terms $a_0, a_1, \cdots$ of a sequence $\mathbf{a}$ of elements of K (or equivalently S). We can think of $\mathbf{a}$ as the R_π-adic expansion of a number α. We wish to determine a pair of elements (p, q) of R such that $\alpha = p/q$ and $\Phi(p, q)$ is minimal among all such pairs. In the rational approximation algorithm, given in Figure 1, the symbols $f = (f_1, f_2)$ and $g = (g_1, g_2)$ denote pairs of elements of R. With these ingredients, the rational approximation algorithm is described in Figure 1. Unlike De Weger's approach to rational approximation, our algorithm is adaptive. That is, the number of terms of known key does not need to be predetermined. The algorithm can continue to revise the approximation as long as new key terms are found. Note that the minimization steps can be carried out with a pair of divisions in R. These divisions can be computed since R is a Euclidean domain. An example of the execution of the algorithm is given in Table 1. The input used is the 2-adic expansion of $-252/269$, the first 30 bits of which are 001010000100100010000110010000. The table shows the values of k, α, g, and f through 15 iterations. The algorithm thus uses 17 bits of the sequence since the first two bits are zero. Note that $17 < 2\lceil \log(269) \rceil$. Thereafter the value of g remains unchanged.

Theorem 6. *Suppose such a norm function exists for R. There is a rational approximation algorithm which, when given $f(2\max(N(p), N(q)))$ terms of the expansion over S of an R-rational number p/q as input, will output an R-FCSR that outputs p/q.*

```
Rational_Approximation()
    begin
    Input a_i s until the first a_{k-1} ≠ 0
    α = a_{k-1} π^{k-1}
    f = (0, π)
    g = (π^{k-1}, 1)
    while more input do
        input a_k
        α = α + a_k π^k
        if α · g_2 − g_1 ≡ 0 ( mod π^{k+1}) then
            f = πf
        else if Φ(g) < Φ(f) then
            Let d minimize Φ(f + dg) with g + df ≡ 0 mod π^{k+1}
            ⟨g, f⟩ = ⟨f + dg, πg⟩
        else
            Let d minimize Φ(g + df) with g + df ≡ 0 mod π^{k+1}
            ⟨g, f⟩ = ⟨g + df, πf⟩
        fi fi
        k = k + 1
    od
    return g
    end
```

Fig. 1. Rational Approximation Algorithm.

It turns out that algebraic number fields that possess such norm functions are rare. For the fields that arise in the construction of FCSRs, as well as those that arise by taking extensions of $\mathbf{Z}$ that are ramified at 2, we have shown this in only two cases.

Corollary 7. *Such a norm function, and hence such a rational approximation algorithm exists for:*

1. *Ordinary FCSRs;*
2. *FCSRs based on* $R = \mathbf{Z}[\pi]$, *with* $\pi^2 + 2 = 0$, $S = \{0, 1\}$, *and* $K = GF(2)$;
3. *FCSRs based on* $R = \mathbf{Z}[\beta]$, *with* $\beta^2 + \beta + 1 = 0$, $S = \{0, 1, \beta, 1 + \beta\}$, *and* $K = GF(4)$.

In the unramified case (the case described in Sections 2 and 3), if $n = 2$ we have a Euclidean domain whenever b_0 is square free and divides b_1. However, if $b_0 = 1$, condition (d) does not hold. In other cases we do not yet know whether condition (d) holds. Thus the algorithm finds rational approximations, but we do not know how fast, or even whether, it converges. In the ramified case (that is, when R is formed by adjoining a real nth root of 2 to $\mathbf{Z}$), condition (d) always holds, but we do not know whether R is a Euclidean domain except when $n = 2$.

k	α	g	f
2	4	$(4,1)$	$(0,2)$
3	4	$(4,1)$	$(0,4)$
4	20	$(-4,3)$	$(0,8)$
5	20	$(-4,3)$	$(0,16)$
6	20	$(12,7)$	$(-8,6)$
7	20	$(4,13)$	$(-16,12)$
8	20	$(20,1)$	$(8,26)$
9	532	$(-12,25)$	$(40,2)$
10	532	$(28,27)$	$(-24,50)$
11	532	$(-52,23)$	$(56,54)$
12	4628	$(-52,23)$	$(112,108)$
13	4628	$(60,131)$	$(-104,46)$
14	4628	$(164,85)$	$(-208,92)$
15	4628	$(164,85)$	$(-416,184)$
16	70164	$(-252,269)$	$(328,170)$

Table 1. Execution of the Rational Approximation Algorithm for $-252/269$.

As a consequence of these considerations, we have new measures of cryptologic security.

Definition 8. If R, I, S is a ring over which FCSRs can be constructed, based on maximal ideal $I = (\pi)$ and lift S of R/I, then the *R-adic span* of a sequence $\mathbf{a}$ is the size of the smallest R-FCSR that outputs $\mathbf{a}$.

Notice that we have been somewhat vague as to what is meant by the size of a FCSR. One reasonable definition might be the integer r, the number of terms in the expansion $q+1 = \sum_{i=1}^{r} q_i 2^i$. This makes sense if p has fewer than r terms in its expansion. More generally, we might take the maximum number of bits (or S cells) required to store the contents of the register, including the extra memory, over the course of its execution. From the cryptographic point of view, what we want is that $\max(N(p), N(q))$ is bounded by the size of the register (or perhaps a multiple of the size). In this case, it follows that a sequence must have large R-adic span to be secure. This indeed holds in the two cases covered by Corollary 7.

5 Conclusions

We have described a general method for constructing feedback with carry shift registers over certain rings of algebraic integers, R. These registers are analogous to linear feedback shift registers. They can be thought of as generating sequences by carrying out division in the completion R_2 of the ring at the prime ideal (2). They carry similar algebraic structures to those of LFSRs.

The cryptographic importance of these registers is twofold. First, they are a potential source of cryptographically secure sequences for stream ciphers. As

with LFSR sequences, there are many possible (as yet unexplored) ways to attempt to modify these sequences to make them secure. Second, these registers can be used for cryptanalysis if a rational approximation algorithm can be devised for R_2. Such an algorithm exists if R is a Euclidean domain with an extra condition on its norm. For only a few Rs we have shown that these conditions occur. It remains to be seen whether other such Rs have these properties and, if not, whether there is a different rational approximation algorithm that works.

Finally, it should be mentioned that further generalizations are possible. We have only considered the cases where R is totally ramified or purely unramified at (2), but more general extensions can be considered. We have also only considered here sequences over finite fields of characteristic two, but the same constructions can be carried out for primes other than two. The advantage might be in obtaining rings for which rational approximation works.

References

1. A. Chan and R. Games, On the linear span of binary sequences from finite geometries, q odd, *IEEE Trans. Info. Theory,* vol. IT-36 (1990) pp. 548-552.
2. M. Goresky and A. Klapper, Feedback registers based on ramified extensions of the 2-adic numbers – extended abstract, *Proceedings of Eurocrypt '94*, Perugia, Italy, May 1994.
3. E. L. Key, An Analysis of the structure and complexity of nonlinear binary sequence generators, *IEEE Trans. Info. Theory,* vol. IT-22 no. 6 (1976) pp. 732-736.
4. A. Klapper, The Vulnerability of Geometric Sequences Based on Fields of Odd Characteristic, *Journal of Cryptology,* **7** (1994) pp. 33-51.
5. A. Klapper and M. Goresky, Feedback Shift Registers, Combiners with Memory, and Arithmetic Codes, *University of Kentucky, Deptartment of Computer Science Technical Report No. 239-93.* Presented at *1993 Cambridge Workshop on Algorithms.*
6. D. Mandelbaum, An approach to an arithmetic analog of Berlekamp's algorithm. *IEEE Trans. Info. Theory,* vol. IT-30 (1984) pp. 758-762.
7. R. Rueppel, *Analysis and Design of Stream Ciphers.* Springer Verlag, New York, 1986.
8. B. M. M. de Weger, *Approximation lattices of p-adic numbers. J. Num. Th.* vol. 24 (1986) pp. 70-88.