Finding Characteristic Polynomials with Jump Indices

Steve Babbage

Vodafone Group R&D, Newbury, UK
steve.babbage@vodafone.com

Matthew Dodd

Independent consultant
matthew@mdodd.net
www.mdodd.net

6th January 2006

Abstract: Jansen introduced a technique for building LFSRs that can be clocked a large number of times with a single simple operation. These may be useful in the construction of stream ciphers based on clock-controlled LFSRs. However, for LFSR sizes of typical interest, it appears generally hard to find such jumping LFSRs with particular desired parameters. In this note we explain a trick which we used to find the jumping LFSRs in MICKEY and MICKEY-128, and which may be useful for future applications.

Keywords: MICKEY, stream cipher, ECRYPT, irregular clocking.

1. Introduction

In [2], Cees Jansen introduces a technique for use in the construction of keystream generators based on clock-controlled LFSRs. This technique is used in the stream ciphers MICKEY and MICKEY-128 [1].

Using naïve methods, it appears impractical to construct applications of this technique except on very short LFSRs. In this note we explain how the LFSRs in MICKEY and MICKEY-128 were created, using a simple algebraic trick which may be useful for future applications.

2. Jumping LFSRs

Jansen's technique is easy to explain. Suppose that the characteristic polynomial of an *n*-stage binary LFSR is $C(x)$; and suppose that $C(x) \mid x^J + x + 1$ for some integer *J*. We call *J* a *jump index* of $C(x)$.

Then the following two operations produce the same result:

- clocking the LFSR *J* times;
- clocking the LFSR once, and then XORing on the original state.

The following example is taken from [2]. Consider the LFSR shown in Figure 1, with characteristic polynomial $C(x) = x^7 + x^6 + 1$. Now $x^7 + x^6 + 1$ divides $x^{121} + x + 1$, so 121 is a jump index for $C(x)$. The operation shown in Figure 2 is therefore equivalent to clocking the LFSR 121 times.

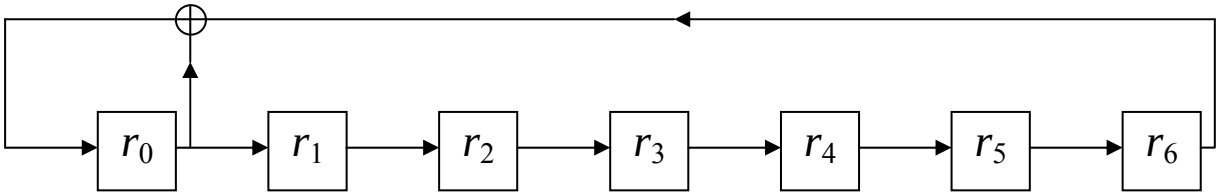

Figure 1: An LFSR with characteristic polynomial $x^7 + x^6 + 1$

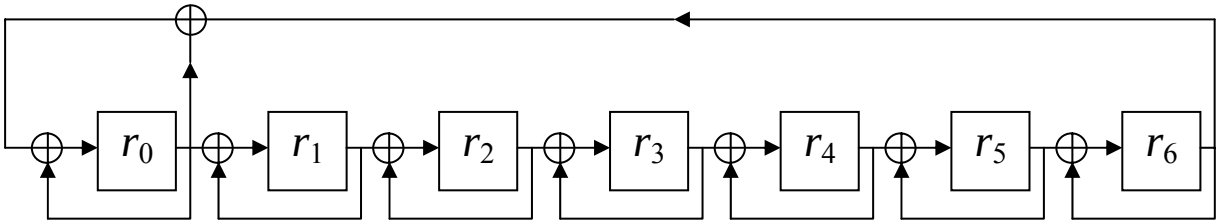

Figure 2: The "clock and XOR" operation equivalent to clocking the same register 121 times

[2] talks only about LFSRs with Fibonacci-style clocking, but it is clear that the same approach is valid with Galois-style clocking (which is used in the MICKEY ciphers).

This construction may be used in stream cipher designs based on clock-controlled LFSRs. By using a clock control bit to select between the types of clocking shown in Figure 1 and Figure 2, we can clock the LFSR either 1 or J times.

It is important to note that:

(a) not all characteristic polynomials have a jump index;

(b) for a given characteristic polynomial, it is not generally easy to determine the jump index[1], if it exists (this is essentially a discrete log problem).

3. Finding the characteristic polynomials for the MICKEY ciphers

For MICKEY, we wanted an LFSR of degree 80, whose primitive characteristic polynomial $C(x)$ had a jump index J close to 2^{40}. Using naïve search methods, it would be impossibly time-consuming to find such a polynomial. However, we were able to apply a simple algebraic trick to make the problem much more tractable.

So let $J = 2^{40} - \delta$, where δ is a small positive integer. $C(x)$ must divide $x^{2^{40}-\delta} + x + 1$; hence

$$C(x) \mid x^{2^{40}} + x^{\delta+1} + x^{\delta} \tag{1}$$

We want $C(x)$ to be primitive, so it must divide $x^{2^{80}} + x$; so

$$C(x) \mid \left(x^{2^{40}}\right)^{2^{40}} + x \tag{2}$$

[1] If J is a jump index for $C(x)$, i.e. $C(x) \mid x^J + x + 1$, and if $J' > J$ is another jump index, then necessarily $C(x) \mid x^{J'} + x^J$, and hence $C(x) \mid x^{J'-J} + 1$. From this we can see that the jump indices for a polynomial $C(x)$ are all congruent modulo p, where p is the least positive integer such that $C(x) \mid x^p + 1$. p is in fact the period of the LFSR with characteristic polynomial $C(x)$. When we talk about *the* jump index (if one exists at all) we mean the unique jump index J in the range $1 < J < p$.

By (1), we can substitute $x^{\delta+1}+x^{\delta}$ for $x^{2^{40}}$ on the right hand side of this expression, so

$$C(x) \mid \left(x^{\delta+1}+x^{\delta}\right)^{2^{40}}+x \tag{3}$$

which, using the fact that exponentiation to the power 2^i is linear, we can rewrite as

$$C(x) \mid \left(x^{2^{40}}\right)^{\delta+1}+\left(x^{2^{40}}\right)^{\delta}+x \tag{4}$$

Then we can again substitute $x^{\delta+1}+x^{\delta}$ for $x^{2^{40}}$ to give

$$C(x) \mid \left(x^{\delta+1}+x^{\delta}\right)^{\delta+1}+\left(x^{\delta+1}+x^{\delta}\right)^{\delta}+x \tag{5}$$

We have thus found a very low degree polynomial (parameterised by δ) which must have $C(x)$ as a factor.

To find a suitable the characteristic polynomial $C(x)$ for MICKEY, we therefore applied the following algorithm, starting at $\delta = \lceil\sqrt{80}\rceil - 1 = 8$:

- Construct $G_{\delta}(x)=\left(x^{\delta+1}+x^{\delta}\right)^{\delta+1}+\left(x^{\delta+1}+x^{\delta}\right)^{\delta}+x$, and see whether it has any factor $F(x)$ of degree 80
- If it does, check whether $F(x)$ is primitive
- If it is, then check whether $F(x)$ really does divide $x^{2^{40}-\delta}+x+1$
- If it does, then set $C(x)=F(x)$
- If $C(x)$ not yet found, then increment δ and start again

With this method we quickly found a degree 80 characteristic polynomial with jump index $J=2^{40}-23$. For MICKEY-128, we also found a degree 128 characteristic polynomial with jump index $J=2^{40}-55$.

4. Generalisations

Let *n* be the desired degree of a characteristic polynomial.

Although we will not fill in any details here, the trick shown above can easily be adapted to finding:

- a characteristic polynomial with jump index $J=2^{an/b}-\delta$, where *a* and *b* are small positive integers (*b* a divisor of *n*) with $a < b$;
- a characteristic polynomial with jump index $J=2^{an/b}+\delta$.

5. References

[1] S.H.Babbage, M.W.Dodd, stream ciphers MICKEY and MICKEY-128, available at http://www.ecrypt.eu.org/stream.

[2] C.J.A.Jansen, *Streamcipher Design: Make your LFSRs jump!*, presented at the ECRYPT SASC (State of the Art in Stream Ciphers) workshop, Bruges, October 2004, and in the workshop record at http://www.isg.rhul.ac.uk/research/projects/ecrypt/stvl/sasc-record.zip.