A Faster Attack on Certain Stream Ciphers[1]

Indexing term: Information Theory

Abstract: *A number of keystream generators can be attacked by guessing the contents of one shift register and then checking to see whether this guess is consistent with the observed keystream. Where the target register is n bits long, this gives an attack of complexity $2^{n-O(1)}$. We present a further optimisation which appears to reduce the complexity to about $2^{n/2}$ in many cases of practical interest.*

Introduction: Many stream cipher systems work by combining each successive bit of plaintext with a pseudo-random bit derived from a keystream generator, which will typically use a nonlinear function of one or more linear feedback shift register sequences to generate these pseudo-random bits. Examples are the multiplexer generator [1], the self-multiplexed generator [2], Geffe's generator [3] and the clock controlled or stop-and-go family of generators [4].

Such stream cipher algorithms are usually faster than block ciphers such as DES [5] and are often used in devices such as line encryptors. Performance issues may lead to their use in videoconferencing and other multimedia applications, and recent developments enable them to provide authentication as well as secrecy [6].

Current attack methods: We will illustrate both old and new methods by means of a concrete example, the multiplexer generator. This was first described in [1], recommended in a standard textbook [7], incorporated in various systems including an EBU standard for scrambled pay-TV [8], and broken independently (using different attacks) by Zeng, Yang and Rao [9] and the author [10]. A further attack was found recently [11].

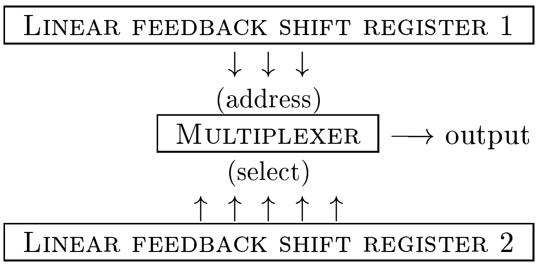

Figure 1 - the multiplex feedback shift register

In this generator, two feedback shift registers are employed, which generate sequences whose lengths are maximal and coprime. Some of the bits of register 1 are used as address lines to a multiplexer, which selects a bit from register 2; this bit becomes the next bit of the keystream sequence (*figure 1*).

[1] appeared in 'Electronics Letters' v 29 no 15 (22nd July 1993) pp 1322-1323

A concrete example of such a generator, which was actually used in a software encryption product, had register 1 with length 31 bits and feedback taps (31, 28) while register 2 was 32 bits long and had feedback taps (32, 31, 30, 10). Thus an exhaustive attack would have required on average 2^{62} operations, and this might seem to have placed an attack beyond the resources of most organisations.

However, it turned out that exhaustive search was not required. Our first improved attack consisted of guessing the contents of register 1 and then checking to see whether the observed keystream was consistent with the feedback taps of register 2 (for each keystream bit, calculate the multiplexer address, put the bit into the appropriate place in register 2, and if this causes a clash, reject the current guess for register 1) [10].

Recent computational work [12] has confirmed that this enables the above system to be broken in a few hours on a workstation. In fact, one only needs about a third of the expected 2^{30} trials, because it is often possible to reject several candidate keys at a time. Similar techniques have been also been applied to the summation generator [13] and can be used against many other generators as well.

Improved method: Until recently, it was common for stream cipher designers to choose a fixed feedback polynomial of low weight. This meant fewer wires and gates in hardware implementations of the algorithm, and faster running software versions as well. The choice of $x^{31} + x^{28} + 1$ as the feedback polynomial for register 1 in the above system is thus fairly typical. However, it enables us to greatly speed up our attack, as we do not need to guess all the bits of register 1, but only just over half of them.

Soppose we guess bits 1 through 18 of register 1, and label these b_1, ..., b_{18}. Then we know that $b_{32} = b_1 + b_4$, $b_{33} = b_2 + b_5$, and so on up to $b_{46} = b_{15} + b_{18}$; then we also have $b_{63} = b_{32} + b_{35}$, etcetera. In effect, guessing 18 bits gives us the values of 18 + 15 + 12 + 9 + 6 + 3 or 64 bits in total, which enables us to calculate 13 + 10 + 7 + 4 + 1 or 36 multiplexer addresses. Similiarly, guessing 24 bits gives us 109 bits in total and 71 addresses. Pending a test implementation, it would seem that we will need to guess somewhere between 18 and 24 bits in order to carry out our consistency check. Thus the overall attack complexity should be about 2^{21} operations.

This attack appears to apply to a number of other generators, including all those mentioned above [1-4]. In general, we might hope for a work factor of about $2^{n/2}$ where n is the length of the target register, although the exact figure will of course depend on the generator details: as the multiplexer generator has $logn$ address lines, 2^{21} - which is $2^{\frac{n}{2}+logn}$ - should not surprise us. In any case, we have shown that where the feedback taps are 'bunched', this reduces the cost of a reconstruction attack on many systems of practical interest to well below the previous $2^{n-O(1)}$.

A note on correlation immunity: Meier and Staffelbach have shown that many stream ciphers are vulnerable to correlation attacks if their feedback polynomials are of low Hamming weight [14]. Chepyzhov and Smeets have extended this to the case where the feedback polynomial divides any polynomial of low weight [15]. In most practical circumstances, polynomials whose nonzero terms are so bunched that a small number of guessed bits imply a much larger number of sequence bits, are likely to be of low weight.

Thus the attacks overlap to some extent. However, they are not the same, as divide-and-conquer attacks can work with much shorter lengths of keystream, and may even succeed where the combining function is correlation free.

For example, consider a multiplexer generator modified so that we take not five but six bits from register 1. The first five of these are used as before to select a bit from register 2, and the sixth bit is then exclusive-or'ed with this bit to give the next bit of keystream. This procedure reduces the probability that the keystream bit will be the same as the corresponding bit of register 2 from $\frac{17}{32}$ to $\frac{1}{2}$, thus preventing a straightforward correlation attack from succeeding; but our attack will clearly still work against this modified generator.

R J Anderson
University Computer Laboratory
Pembroke Street, Cambridge CB2 3QG

References

[1] Jennings SM, *'A Special Class of Binary Sequences'*, PhD Thesis, University of London 1980.

[2] PR Geffe, "How to protect data with ciphers that are really hard to break", in *Electronics*, January 4 1973, p 99 - 101

[3] Sun DF, "The Structure and Properties of YC-sequences", Proc. Chinacrypt 92 pp 122 - 131

[4] Gollmann D and WC Chambers, "Clock-controlled Shift Registers: A review", in *IEEE Transactions on Selected Areas in Communications* **SAC-7** (May 1989) pp 525 - 533

[5] *'Specifications for the Data Encryption Standard'*, FIPS PUB 46 (1977)

[6] Lai XJ, RA Rueppel, J Woolven, "A Fast Cryptographic Checksum Algorithm Based on Stream Ciphers", in *Proceedings of Auscrypt 92, to appear*

[7] Beker H and F Piper, *'Cipher Systems'*, Northwood 1982

[8] *'Access control system for the MAC/packet family: eurocrypt'*, EBU Technical Document (2nd issue - July 1991)

[9] Zeng KC, CH Yang and TRN Rao, "On the linear consistency test (LCT) in cryptanalysis and its applications", in *Advances in Cryptology - Crypto 89*, Springer LNCS **435** pp 164 - 174

[10] Anderson RJ, "Solving a class of stream ciphers", in *Cryptologia* **14** no 3 (1990) pp 285 - 288

[11] Daemen J, R Govaerts and B Preneel, "Cryptanalysis of MUX-LFSR Based Scramblers", presented at Fondazione Ugo Bordoni seminar on cryptology, Rome, February 1993

[12] Dawson E and A Clark, "Cryptanalysis of universal logic sequences", *preprint*

[13] Dawson E, "Cryptanalysis of summation generator", in *Proceedings of Auscrypt 92, to appear*

[14] Meier W and O Staffelbach, "Fast correlation attacks on certain stream ciphers", in *Journal of Cryptology* 1989 pp 159 - 176

[15] Chepyzhov V and B Smeets, " On a Fast Correlation Attack on Certain Stream Ciphers", in *Advances in Cryptology - Eurocrypt 91* Springer LNCS **547** pp 176 - 185