Remarks on the security of the *Alpha1* stream cipher

Chris J. Mitchell

Technical Report
RHUL–MA–2001–8
19 December 2001

Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
http://www.rhul.ac.uk/mathematics/techreports

Abstract

A preliminary analysis of the Alpha1 stream cipher is given. Some undesirable properties are identified.

1 Introduction

This short note contains some observations on a recent paper [1] by Komninos, Honary and Darnell.

2 Specification issues

The following minor points of correctness arise with the respect to the specification of the sequence generator in [1].

- It is claimed that all four polynomials (determining the feedbacks for registers R1–R4) are primitive, which implies that the polynomials are irreducible. The polynomial for register R4 is listed in Table 1 of [1] (page 296) as being

  $$x^{35} + x^{30} + x^{22} + x^{11} + x^6 + 1.$$

  Unfortunately this polynomial is not irreducible and hence is not primitive. This is simple to see since

  $$(x + 1)|(x^{35} + x^{30} + x^{22} + x^{11} + x^6 + 1).$$

  In fact, when working over GF(2), $x + 1$ is a factor of any polynomial with an even number of non-zero coefficients, i.e. with an even Hamming weight. Hence, primitive polynomials must always have an odd number of non-zero coefficients.

  One effect of this choice for the R4 polynomial is that, if the register R4 is loaded with all ones, then it will always stay in this state (since there are an odd number of feedback taps), i.e. it will generate a sequence of all ones.

- It is stated on page 298 of [1] that the probability that R2, R3 and R4 will be clocked by the control mechanism is approximately 7/13. In

fact, if we assume that the six bits used to control the clocking mechanism are randomly distributed, then the probability will be precisely 9/16. This is simple to see by considering each of the $2^6 = 64$ possible 6-tuples of bits used to control the clocking mechanism. In precisely 36 of the 64 6-tuples, register R2 will be clocked (the same holds for registers R3 and R4). Hence the result follows, assuming each 6-tuple is equally probable and observing that $36/64 = 9/16$.

- It is not clear how many times registers R2, R3 and R4 are clocked in each clock cycle. Specifically, it is not completely clear whether they are always clocked once and sometimes clocked an additional time, or whether they are only clocked at most once.

  Based on conversations with the authors the assumption is made below that the latter option applies, i.e. registers R2, R3 and R4 are either clocked zero or once per output bit.

- A number of references are made to the GSM A5/1 cipher, and to attacks on this cipher. However, the authors of [1] fail to distinguish between the so called 'alleged A5' scheme, publicised in the mid 1990s, and the actual A5/1 cipher, which was only put into the public domain in the late 1990s. The alleged A5 scheme is clearly a rather weak scheme, and was broken even before its design was publicised. However, the 'genuine' A5/1 scheme was only broken some months after it was made public, using apparently novel techniques. This distinction is rather an important one.

3 Analysis

We start by defining some simple notation. The stream cipher output sequence is equal to the ex-or of the outputs of all four registers, ex-ored with the logical-and of the output of registers R2 and R3. That is, if we write x_{ij} for the jth output bit of register Ri, $(1 \leq i \leq 4,\ j \geq 0)$ then the jth output bit of keystream is

$$y_j = x_{1j} + x_{2j} + x_{3j} + x_{4j} + x_{2j}x_{3j}.$$

3.1 A correlation property

We note the following correlation property of the scheme, namely that the output sequence is strongly correlated to the complement of the sum of the

outputs of R1 and R4. Specifically we have the following result.

Lemma 1 *The probability that*

$$y_j = x_{1j} + x_{4j} + 1$$

is 0.75.

Proof. To see this, note that

$$y_j + (x_{1j} + x_{4j} + 1) = x_{2j} + x_{3j} + x_{2j}x_{3j}$$

which is 0 with probability 0.75. The result follows. □

3.2 Breaking register R1

The following analysis is based on the assumption that registers R2, R3 and R4 are clocked either zero or one times per clock cycle. We also assume, as previously, that the linear feedback shift registers R1–R4 will behave randomly, i.e. that the probability that any bit in a register state is 0 is 0.5.

Given these assumptions we now describe a known-plaintext attack of complexity 2^{29} which reveals the initial state of register R1. It operates as follows.

We first need the following result.

Lemma 2 *For any $j \geq 0$ and any i $(2 \leq i \leq 4)$*

$$x_{ij} + x_{i(j+1)} = 0$$

with probability approximately 23/32.

Proof. The probability that register Ri is clocked between outputting bits x_{ij} and $x_{i(j+1)}$ is 9/16 (as above). If the register is not clocked then these two bits will clearly be identical. If the register is clocked, then (using the randomness assumption) the probability that these two bits will be the same is 0.5. Hence the probability that $x_{ij} = x_{i(j+1)}$ is equal to

$$7/16 + 0.5 \times 9/16$$

and the result follows. □

This then leads to the following result.

Lemma 3 *For any* $j \geq 0$

$$(y_j + x_{1j}) + (y_{j+1} + x_{1(j+1)}) = 0$$

with probability approximately $71/128 = 0.5546875$.

Proof. We consider four cases.

(i) $y_j + x_{1j} = x_{4j} + 1$ and $y_{j+1} + x_{1(j+1)} = x_{4(j+1)} + 1$. In this case, by Lemma 2, the probability that $(y_j + x_{1j}) + (y_{j+1} + x_{1(j+1)}) = 0$ is $23/32$.

(ii) $y_j + x_{1j} = x_{4j} + 1$ and $y_{j+1} + x_{1(j+1)} = x_{4(j+1)}$. In this case, by Lemma 2, the probability that $(y_j + x_{1j}) + (y_{j+1} + x_{1(j+1)}) = 0$ is $9/32$.

(iii) $y_j + x_{1j} = x_{4j}$ and $y_{j+1} + x_{1(j+1)} = x_{4(j+1)} + 1$. In this case, by Lemma 2, the probability that $(y_j + x_{1j}) + (y_{j+1} + x_{1(j+1)}) = 0$ is $9/32$.

(iv) $y_j + x_{1j} = x_{4j}$ and $y_{j+1} + x_{1(j+1)} = x_{4(j+1)}$. In this case, by Lemma 2, the probability that $(y_j + x_{1j}) + (y_{j+1} + x_{1(j+1)}) = 0$ is $23/32$.

By Lemma 1, the probabilities of the four cases are respectively: $(3/4)^2 = 9/16$, $3/4 \times 1/4 = 3/16$, $1/4 \times 3/4 = 3/16$, and $(1/4)^2 = 1/16$. Hence the overall probability that $(y_j + x_{1j}) + (y_{j+1} + x_{1(j+1)}) = 0$ is equal to

$$23/32 \times 9/16 + 2 \times 9/32 \times 3/16 + 23/32 \times 1/16 = 284/512$$

and the result follows. □

We can now describe the attack to find the initial state of register R1. It involves searching though all the 2^{29} possible initial states for R1. It also supposes that the attacker knows a number of consecutive pairs of cipher sequence bits (e.g. as obtained from a known plaintext attack).

For each guess for the initial state of R1, it is trivial to compute the x_{1j} values for every value of j for which the cipher sequence bit y_j is known. For every value of j for which both y_j and y_{j+1} are known, now compute

$$(y_j + x_{1j}) + (y_{j+1} + x_{1(j+1)}).$$

If the guess for the initial state is correct, every such value will be zero with probability approximately 0.5546875. If the guess for the initial state is incorrect, every such value will be zero with probability approximately 0.5. Hence, given a sufficient number of known sequence bits, the correct initial state for R1 can be found by taking the candidate which yields the maximum number of zero values for

$$(y_j + x_{1j}) + (y_{j+1} + x_{1(j+1)}).$$

3.3 Obtaining probabilistic information on the plaintext

We conclude this short note by observing that, if the attack described immediately above is used to successfully deduce the initial contents of register R1, this can then be used to deduce probabilistic information regarding the plaintext corresponding to known ciphertext.

That is, suppose $z_0, z_1, \ldots, z_{t-1}$ are t bits of known ciphertext. Then we know that

$$z_i = m_i + y_i$$

for every i, where $m_0, m_1, \ldots, m_{t-1}$ are the t bits of plaintext enciphered to yield $z_0, z_1, \ldots, z_{t-1}$. Now, if the contents for register R1 have been deduced, the attacker can readily compute $x_{10}, x_{11}, \ldots, x_{1(t-1)}$, and hence can compute the sequence

$$z_0 + x_{10}, z_1 + x_{11}, \ldots, z_{t-1} + x_{1(t-1)}$$

which equals

$$m_0 + y_0 + x_{10}, m_1 + y_1 + x_{11}, \ldots, m_{t-1} + y_{t-1} + x_{1(t-1)}.$$

Call this sequence $u_0, u_1, \ldots, u_{t-1}$.

Hence, for any j $(0 \leq j < t-1)$ we know that

$$m_j + m_{j+1} = (u_j + u_{j+1}) + (y_j + x_{1j}) + (y_{j+1} + x_{1(j+1)}).$$

Thus, by Lemma 3 we know that

$$m_j + m_{j+1} = (u_j + u_{j+1})$$

with probability 71/128.

References

[1] N. Komninos, B. Honary, and M. Darnell. An efficient stream cipher Alpha1 for mobile and wireless devices. In B. Honary, editor, *Cryptography and Coding — 8th IMA International Conference, UK, December 2001*, number 2260 in Lecture Notes in Computer Science, pages 294–300. Springer-Verlag, Berlin, 2001.