Cryptanalysis of Stream Cipher Alpha1

Hongjun Wu

Laboratories for Information Technology
21 Heng Mui Keng Terrace, Singapore 119613
hongjun@lit.org.sg

Abstract. Komninos, Honary and Darnell recently proposed stream cipher Alpha1. Alpha1 is based on A5/1 and is claimed to be much safer than A5/1. However Alpha1 is insecure: a 29-bit feedback shift register could be recovered with about 3000 known plaintext bits with $O(2^{29})$ operations. The rest of the message could be recovered in the broadcast application in which the same message is encrypted with different keys.

1 Introduction

The stream cipher Alpha1 [6] was proposed recently to strengthen A5/1. A5/1 is used to protect the mobile communication privacy and is with extremely simple structure: three LFSRs (64 bits in total) are used and the output is obtained by XORing the most significant bits of those registers. The LFSRs go/stop according to the majority rule: each register provides one bit (clocking tap) and the majority bit is computed. The registers whose clocking taps agreeing with the majority bit are clocked.

There have been several attacks on A5/1 [1,5,3,4]. The real time known-plaintext cryptanalysis of A5/1 were given by Biryukov, Shamir and Wagner [4] by exploiting two major weaknesses of A5/1: the relatively small number of internal states (2^{64}) and the poor choice of the clocking taps. As a cipher with small number of internal states, A5/1 is vulnerable to the general time-memory tradeoff attack on stream ciphers (which is discovered independently by Golic [5] and Babbage [2]). The poor choice of clocking taps enables special states being accessed easily and the efficiency of the time-memory tradeoff attack can be improved significantly.

To strengthen A5/1, Komninos, Honary and Darnell proposed Alpha1. Alpha1 is with four LFSRs (128 bits in total). Three registers go/stop according to two groups of clocking taps while one register is regularly clocked. The attacks against A5/1 seem no longer applicable to Alpha1. Unfortunately, an additional binary AND operation is used in Alpha1 in order to achieve high linear complexity. This AND operation causes a disaster to Alpha1: the combination of the output of those irregularly clocked LFSRs is not uniformly distributed. The statistical information could be applied to recover the register which is regularly clocked. The design of Alpha1 reminds us that it is dangerous to modify the internal structure of a cipher.

L. Batten and J. Seberry (Eds.): ACISP 2002, LNCS 2384, pp. 169–175, 2002.

After recovering the regularly clocked register, information of the message is leaked. If a message is encrypted with a number of different keys and those ciphertext are known (in some broadcast applications), the rest of the message could be recovered. If a message is longer than $2^{29}-1$ bits and is encrypted with different keys, the ciphertext only attack is applicable to recover the message.

This paper is organized as follows. Section 2 introduces the Alpha1 stream cipher. The attack against Alpha1 is given in Section 3. The message recovery in the broadcast application is studied in Section 4. Section 5 concludes this paper.

2 Stream Cipher Alpha1

Alpha1 is built from four short linear feedback shift registers (LFSRs) of lengths $29, 31, 33$ and 35 bits, which are denoted as R_1, R_2, R_3 and R_4 respectively. Four primitive polynomials $p_1(x)$, $p_2(x)$, $p_3(x)$ and $p_4(x)$ are used for R_1, R_2, R_3 and R_4 respectively. They are given in Table 1.

Table 1. Primitive Polynomials

$p_1(x)$	$x^{29}+x^{27}+x^{24}+x^8+1$
$p_2(x)$	$x^{31}+x^{28}+x^{23}+x^{18}+1$
$p_3(x)$	$x^{33}+x^{28}+x^{24}+x^4+1$
$p_4(x)$	$x^{35}+x^{30}+x^{22}+x^{11}+x^6+1$

We denote R_1, R_2, R_3 and R_4 as

$$R_1 = \sum_{i=0}^{28} R_{1,i} \cdot x^i, \qquad R_2 = \sum_{i=0}^{30} R_{2,i} \cdot x^i$$

$$R_3 = \sum_{i=0}^{32} R_{3,i} \cdot x^i, \qquad R_4 = \sum_{i=0}^{34} R_{4,i} \cdot x^i$$

The output y of Alpha1 is given as

$$y = R_{1,28} \oplus R_{2,30} \oplus R_{3,32} \oplus R_{4,34} \oplus (R_{2,30} \ \& \ R_{3,32})$$

where '&' is the binary AND operator.

R_1 is shifted every clock cycle. R_2, R_3 and R_4 are clocked in a stop/go fashion according to the following majority rule: R_2 is clocked if $R_{2,10}$ agrees with the majority bit of $(R_{2,10}, R_{3,22}, R_{4,11})$ and $R_{2,21}$ agrees with the majority bit of $(R_{2,21}, R_{3,10}, R_{4,24})$; R_3 is clocked if $R_{3,22}$ agrees with the majority bit of $(R_{2,10}, R_{3,22}, R_{4,11})$ and $R_{3,10}$ agrees with the majority bit of $(R_{2,21}, R_{3,10}, R_{4,24})$; R_4 is clocked if $R_{4,11}$ agrees with the majority bit of $(R_{2,10}, R_{3,22}, R_{4,11})$ and $R_{4,24}$ agrees with the majority bit of $(R_{2,21}, R_{3,10}, R_{4,24})$. We list in Table 2 all the 16 cases that R_2, R_3 and R_4 stop/go. The designers of Alpha1 listed only 13 cases and estimated wrongly that each R_2, R_3 and R_4 move with probability $\frac{7}{13}$ [6]. We need to mention here that those 16 cases do not occur independently. We will discuss the probabilities of those cases in Section 3.

Table 2. Registers (R_2, R_3, R_4) Being Shifted At One Clock Cycle

	Condition 1	Condition 2	Registers Being Shifted
Case 1	$R_{2,10} = R_{3,22} \neq R_{4,11}$	$R_{2,21} = R_{3,10} \neq R_{4,24}$	R_2, R_3
Case 2	$R_{2,10} = R_{3,22} \neq R_{4,11}$	$R_{2,21} \neq R_{3,10} = R_{4,24}$	R_3
Case 3	$R_{2,10} = R_{3,22} \neq R_{4,11}$	$R_{2,21} = R_{4,24} \neq R_{3,10}$	R_2
Case 4	$R_{2,10} = R_{3,22} \neq R_{4,11}$	$R_{2,21} = R_{3,10} = R_{4,24}$	R_2, R_3
Case 5	$R_{2,10} \neq R_{3,22} = R_{4,11}$	$R_{2,21} = R_{3,10} \neq R_{4,24}$	R_3
Case 6	$R_{2,10} \neq R_{3,22} = R_{4,11}$	$R_{2,21} \neq R_{3,10} = R_{4,24}$	R_3, R_4
Case 7	$R_{2,10} \neq R_{3,22} = R_{4,11}$	$R_{2,21} = R_{4,24} \neq R_{3,10}$	R_4
Case 8	$R_{2,10} \neq R_{3,22} = R_{4,11}$	$R_{2,21} = R_{3,10} = R_{4,24}$	R_3, R_4
Case 9	$R_{2,10} = R_{4,11} \neq R_{3,22}$	$R_{2,21} = R_{3,10} \neq R_{4,24}$	R_2
Case 10	$R_{2,10} = R_{4,11} \neq R_{3,22}$	$R_{2,21} \neq R_{3,10} = R_{4,24}$	R_4
Case 11	$R_{2,10} = R_{4,11} \neq R_{3,22}$	$R_{2,21} = R_{4,24} \neq R_{3,10}$	R_2, R_4
Case 12	$R_{2,10} = R_{4,11} \neq R_{3,22}$	$R_{2,21} = R_{3,10} = R_{4,24}$	R_2, R_4
Case 13	$R_{2,10} = R_{3,22} = R_{4,11}$	$R_{2,21} = R_{3,10} \neq R_{4,24}$	R_2, R_3
Case 14	$R_{2,10} = R_{3,22} = R_{4,11}$	$R_{2,21} \neq R_{3,10} = R_{4,24}$	R_3, R_4
Case 15	$R_{2,10} = R_{3,22} = R_{4,11}$	$R_{2,21} = R_{4,24} \neq R_{3,10}$	R_2, R_4
Case 16	$R_{2,10} = R_{3,22} = R_{4,11}$	$R_{2,21} = R_{3,10} = R_{4,24}$	R_2, R_3, R_4

3 Cryptanalysis of Alpha1 – Recover R_1

In this section, we show that R_1 could be determined with about 3000 known plaintext bits by exploiting the non-uniform distribution of the output of R_2, R_3 and R_4. We write the Alpha1 output y alternatively as

$$y = R_{1,28} \oplus (R_{2,30}|R_{3,32}) \oplus R_{4,34}$$

where '$|$' is the binary inclusive OR operator. Let

$$y' = (R_{2,30}|R_{3,32}) \oplus R_{4,34},$$

then

$$y = R_{1,28} \oplus y'.$$

Denote the output of Alpha1 as Y and let the sequences consisting of y' and $R_{1,28}$ be Y' and Y'' respectively. Obviously $Y = Y' \oplus Y''$. Since R_1 is clocked regularly, there are only 2^{29} possible values for Y''. If the bits of Y' are not uniformly distributed, those statistical properties could be used to recover Y'' $(= R_1)$. We will show below that 00, 01, 10 and 11 in Y' are not uniformly distributed.

We start with computing the probabilities of those 16 cases. By analyzing Table 2, we know that: if Case 2 or 5 appears, only Case 1, 2, 5 and 6 may appear in the next clock cycle and each one appears with equal probability $\frac{1}{4}$; if Case 3 or 9 appears, only Case 1, 3, 9 and 11 may appear in the next clock cycle and

each one appears with equal probability $\frac{1}{4}$; if Case 7 or 10 appears, only Case 6, 7, 10 and 11 may appear in the next clock cycle and each one appears with equal probability $\frac{1}{4}$; if Case 1, 4, 6, 8, 11, 12, 13, 14, 15 or 16 appears, in the next clock cycle each of those 16 cases appears with equal probability $\frac{1}{16}$. These 16 cases form a Markoff Chain. We compute the steady state probabilities and list them in Table 3.

Table 3. The Probability of Each Case

	Probability		Probability
Case 1	$\frac{3}{28}$	Case 9	$\frac{2}{28}$
Case 2	$\frac{2}{28}$	Case 10	$\frac{2}{28}$
Case 3	$\frac{2}{28}$	Case 11	$\frac{3}{28}$
Case 4	$\frac{1}{28}$	Case 12	$\frac{1}{28}$
Case 5	$\frac{2}{28}$	Case 13	$\frac{1}{28}$
Case 6	$\frac{3}{28}$	Case 14	$\frac{1}{28}$
Case 7	$\frac{2}{28}$	Case 15	$\frac{1}{28}$
Case 8	$\frac{1}{28}$	Case 16	$\frac{1}{28}$

At the ith clock cycle, if Case 1, 4 or 13 appears, only R_2 and R_3 are clocked, (y'_i, y'_{i-1}) would be 00, 01, 10, 11 with probability $\frac{5}{16}$, $\frac{3}{16}$, $\frac{3}{16}$, $\frac{5}{16}$ respectively; if Case 2, 3, 5 or 9 appears, only one of R_2 and R_3 is clocked, (y'_i, y'_{i-1}) would be 00, 01, 10, 11 with probability $\frac{3}{8}$, $\frac{1}{8}$, $\frac{1}{8}$, $\frac{3}{8}$ respectively; if Case 6, 7, 8, 10, 11, 12, 14, 15 or 16 appears, R_4 is clocked and R_2 or R_3 may be clocked, (y'_i, y'_{i-1}) would be 00, 01, 10, 11 with equal probability $\frac{1}{4}$. Thus (y'_i, y'_{i-1}) would be 00, 01, 10, 11 with probability $\frac{19}{64}$, $\frac{13}{64}$, $\frac{13}{64}$, $\frac{19}{64}$ respectively.

In the following we compute how much information of R_1 could be recovered if $(n+1)$ bits of the key stream Y are known. Denote this $(n+1)$-bit key stream as Y_n. Randomly set the initial value of R_1 as a 29-bit non-zero z and generate an $(n+1)$-bit output $Y''_{n,z}$ which is related to Y_n. Let

$$Y'_{n,z} = Y_n \oplus Y''_{n,z}$$

Let n_{00}, n_{01}, n_{10} and n_{11} represent the number of 00, 01, 10 and 11 in $Y'_{n,z}$ respectively. If z is the correct R_1, 00, 01, 10 and 11 would appear in $Y'_{n,z}$ with probability $\frac{19}{64}$, $\frac{13}{64}$, $\frac{13}{64}$, $\frac{19}{64}$ respectively. Let

$$G_{n,x} = \{(n_{00}, n_{01}, n_{10}, n_{11}) \mid n_{00}, n_{11} \geq n \cdot (\frac{19}{64} - x), n_{01}, n_{10} \leq n \cdot (\frac{13}{64} + x) \\ \text{and } n_{00} + n_{01} + n_{10} + n_{11} = n\}$$

x is chosen such that if z is the correct R_1, $(n_{00}, n_{01}, n_{10}, n_{11}) \in G_{n,x}$ with probability P_1 close to 1. For the wrong z, $(n_{00}, n_{01}, n_{10}, n_{11}) \in G_{n,x}$ with probability P_2. These two probabilities are given as:

$$P_1 = \sum_{(n_{00}, n_{01}, n_{10}, n_{11}) \in G_{n,x}} \mathrm{C}(n; n_{00}, n_{01}, n_{10}, n_{11}) \left(\frac{19}{64}\right)^{n_{00}+n_{11}} \left(\frac{13}{64}\right)^{n_{01}+n_{10}} \quad (1)$$

$$P_2 = \sum_{(n_{00},n_{01},n_{10},n_{11})\in G_{n,x}} \mathrm{C}(n; n_{00}, n_{01}, n_{10}, n_{11}) \left(\frac{1}{4}\right)^n \tag{2}$$

where $\mathrm{C}(n; n_{00}, n_{01}, n_{10}, n_{11})$ is the multinomial coefficient $n!/(n_{00}!n_{01}!n_{10}!n_{11}!)$. The information leakage Δs of R_1 is given as

$$\Delta s = 29 - (-\sum_{i=1}^{|A|} \frac{P_1}{|A|} \cdot \log_2 \frac{P_1}{|A|} - \sum_{i=1}^{|B|} \frac{1-P_1}{|B|} \cdot \log_2 \frac{1-P_1}{|B|}) \tag{3}$$

where

$$A = \{z \mid \text{ For } Y'_{n,z},\ (n_{00}, n_{01}, n_{10}, n_{11}) \in G_{n,x}\}$$
$$B = \{z \mid \text{ For } Y'_{n,z},\ (n_{00}, n_{01}, n_{10}, n_{11}) \notin G_{n,x}\}$$

Let $|A| \approx 2^{29} \cdot P_2 + 1$ and $|B| = 2^{29} - |A|$, (3) is approximated as

$$\Delta s \approx P_1 \cdot \log_2 \frac{P_1}{P_2 + 2^{-29}} + (1 - P_1) \cdot \log_2 \frac{1 - P_1}{1 - P_2 - 2^{-29}} \tag{4}$$

It is difficult to compute P_1 and P_2 directly for n with large value. We approximate (1) and (2) with the multivariate normal distribution:

$$P_1 = \int_{n\cdot\left(\frac{19}{64}-x\right)-\frac{1}{2}}^{n} \int_{n\cdot\left(\frac{19}{64}-x\right)-\frac{1}{2}}^{n} \int_{0}^{n\cdot\left(\frac{13}{64}+x\right)+\frac{1}{2}} f(z_1, z_2, z_3) dz_1 dz_2 dz_3 \tag{5}$$

where

$$f(z_1, z_2, z_3) = \frac{1}{\sqrt{(2\pi)^3(\det \boldsymbol{V})}} e^{-(\boldsymbol{z}-\boldsymbol{u})^T \boldsymbol{V}^{-1} (\boldsymbol{z}-\boldsymbol{u})/2}$$
$$\boldsymbol{z} = \begin{pmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \boldsymbol{z}_3 \end{pmatrix} \qquad \boldsymbol{u} = \begin{pmatrix} \frac{13}{64}n \\ \frac{19}{64}n \\ \frac{19}{64}n \end{pmatrix}$$
$$\boldsymbol{V} = \begin{pmatrix} \frac{13}{64} \times \frac{51}{64}n & -\frac{13}{64} \times \frac{19}{64}n & -\frac{13}{64} \times \frac{19}{64}n \\ -\frac{19}{64} \times \frac{13}{64}n & \frac{19}{64} \times \frac{45}{64}n & -\frac{19}{64} \times \frac{19}{64}n \\ -\frac{19}{64} \times \frac{13}{64}n & -\frac{19}{64} \times \frac{19}{64}n & \frac{19}{64} \times \frac{45}{64}n \end{pmatrix}$$

P_2 is approximated in a similar way as P_1 except that

$$\boldsymbol{u} = \begin{pmatrix} \frac{n}{4} \\ \frac{n}{4} \\ \frac{n}{4} \end{pmatrix} \qquad \boldsymbol{V} = \begin{pmatrix} \frac{3}{16}n & -\frac{1}{16}n & -\frac{1}{16}n \\ -\frac{1}{16}n & \frac{3}{16}n & -\frac{1}{16}n \\ -\frac{1}{16}n & -\frac{1}{16}n & \frac{3}{16}n \end{pmatrix} \tag{6}$$

From (4), (5) and (6), we compute the information leakage Δs of R_1 for different values of n. The numerical integration function in MATHEMATICA is used in the computation. Different values of x are tested for each n so that the maximum value of Δs could be obtained. The results are given in Table 4.

Table 4. $\Delta s_{\max}$ vs n

n	x	P_1	P_2	$\Delta s_{\max}$
512	$\frac{10}{512}$	0.645	$2^{-9.953}$	5.477
768	$\frac{14}{768}$	0.706	$2^{-13.41}$	8.588
1024	$\frac{17}{1024}$	0.728	$2^{-17.47}$	11.88
1280	$\frac{20}{1280}$	0.752	$2^{-21.44}$	15.32
1536	$\frac{24}{1536}$	0.796	$2^{-24.58}$	18.79
1792	$\frac{29}{1792}$	0.847	$2^{-26.92}$	21.94
2048	$\frac{35}{2048}$	0.897	$2^{-28.53}$	24.39
2304	$\frac{42}{2304}$	0.937	$2^{-29.45}$	26.09
2560	$\frac{49}{2560}$	0.962	$2^{-30.42}$	27.21
2816	$\frac{56}{2304}$	0.977	$2^{-31.44}$	27.93
3072	$\frac{64}{3072}$	0.987	$2^{-31.84}$	28.35

From Table 4 about 28 bits of R_1 could be recovered with about 3000 known plaintext bits. We implemented the attack for $n = 3072$, $x = \frac{64}{3072}$. The attack is repeated for 100 times for different initial values of R_1, R_2, R_3 and R_4. Each experiment takes about 3.6 hours on Pentium IV 1.7GHz. These 100 experiments give 90 correct R_1 and 11 wrong R_1 (80 experiments give only the correct R_1; 11 experiments give one correct R_1 and one wrong R_1; 9 experiments give no value for R_1). The experiment results are close to the estimated results given in Table 4.

4 Message Recovery in the Broadcast Applications

In Section 3, we recovered R_1 with known plaintext attack. With the knowledge of R_1 each bit of the ciphertext leaks $2^{-5.3}$-bit information since 00, 01, 10 and 11 appear in the output of R_2, R_3 and R_4 with probabilities $\frac{19}{64}$, $\frac{13}{64}$, $\frac{13}{64}$ and $\frac{19}{64}$ respectively. In some broadcast applications, one message is encrypted with different keys and sent to different users. For 31, 63, 127 and 255 different keys,the information leakage for each plaintext bit is 0.405, 0.650, 0.880 and 0.986 bit respectively.

If the message is longer than $(2^{29} - 1)$ bits, even the ciphertext only attack could be applied to recover a lot of information if the message is encrypted with sufficiently large number of different keys. Assume that the message is $(2^{30} - 2)$ bits long, denote the plaintext and ciphertext as $m_1 \parallel m_2$ and $c_1 \parallel c_2$ respectively, where m_1, m_2, c_1 and c_2 are $(2^{29} - 1)$-bit long and $\parallel$ denotes concatenation. The output of R_2, R_3 and R_4 is denoted as $Y_1' \parallel Y_2'$. Since the output of R_1 is with period $2^{29} - 1$, we obtain $c_1 \oplus c_2 = (Y_1' \oplus Y_2') \oplus (m_1 \oplus m_2)$. The 00, 01, 10 and 11 appear in $Y_1' \oplus Y_2'$ with probabilities $\frac{265}{1024}$, $\frac{247}{1024}$, $\frac{247}{1024}$ and $\frac{265}{1024}$ respectively. The information leakage is $2^{-10.1}$ for each bit of $m_1 \oplus m_2$. For 255, 511, 1023 and 2047 different keys, the information leakage for each bit of $m_1 \oplus m_2$ is 0.135, 0.252, 0.442 and 0.690 bit respectively.

5 Conclusions

The attack given in this paper recovers the 29-bit register R_1 from around 3000-bit known plaintext. The rest of the message could be recovered in the broadcast applications. It is an open problem whether R_2, R_3 and R_4 could be recovered by analyzing their non-uniformly distributed output.

References

1. R. Anderson, and M. Roe, A5, `http://jya.com/crack-a5.htm`, 1994.
2. S. Babbage, *A Space/Time Tradeoff in Exhaustive Search Attacks on Stream Ciphers*, European Convention on Security and Detection, IEE Conference publication, No. 408, May 1995.
3. M. Briceno, I. Goldberg, D. Wagner, "A pedagogical implementation of A5/1", `http://www.scard.org`, May 1999.
4. A. Biryukov, A. Shamir, D. Wagner, "Real Time Cryptanalysis of A5/1 on a PC", in *Fast Software Encryption*, LNCS 1978, pp. 1-18, Springer-Verlag 2000.
5. J. Golic, "Cryptanalysis of Alleged A5 Stream Cipher", in *Advances in Cryptology – Eurocrypt'97*, LNCS 1233, pp. 239 - 255, Springer-Verlag 1997.
6. N. Komninos, B. Honary, and M. Darnell, "An Efficient Stream Cipher Alpha1 for Mobile and Wireless Devices", in *Proceedings of the 8th IMA Conference on Cryptography and Coding*, LNCS 2260, pp. 294-300.