http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (24): 8005-8009, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.8005.8009

Improved Distinguishers on Stream Cipher Achterbahn-v2

Li Shun-Bo, Wang Yan and Peng Jia-Long
School of Science, Xi'an University of Architecture and Technology, Xi'an, 710055, Shaanxi, China

Abstract: Stream cipher Achterbahn-v2 is one of 34 candidates submitted to the eSTREAM project which is based on several nonlinear feedback shift registers (NLFSRs) and a nonlinear filter function. By finding the better parity check equations, an improved distinguishing attack on Achterbahn-v2 is proposed. Achterbahn-v2 can be distinguishable from truly random cipher by about $O(2^{516})$ keystream bits. Moreover, this method reduces the data complexity of Plasencia's results by a factor of about 0.77.

Key words: Cryptanalysis, stream cipher, distinguishing attack, achterbahn, parity check

INTRODUCTION

Achterbahn (Gammel *et al*., 2005a) is one of 34 candidates submitted to the eSTREAM project, there have been some successful attacks on Achterbahn (Johansson *et al*., 2006). As a response to these attacks, the cipher was updated to a more secure version, denoted Achterbahn-v2 (Gammel *et al*., 2005b) which was designed to resist approximations of the output functions, linear approximations as well as quadratic and cubic approximations. Recently, there had some results to Achterbahn-v2, such as distinguishing attack (Coppersmith *et al*., 2002; Hell *et al*., 2012) and linear cryptanalysis.

Hell and Johansson (2007) presented a decimation attack based on a quadratic approximation of the out Boolean function, this key-recovery attack on Achterbahn-v2 using a meet-in-the-middle attack requires time complexity $O(2^{59})$ and $O(2^{59})$ keystream bits. Huang and Wu (2007) gave a linear distinguishing attack on Achterbahn-v2 with time complexity $O(2^{48})$ and $O(2^{50.4})$ keystream bits. However, this attack can not recover the initial keys. Naya-Plasencia (2007) uses linear approximations key-recovery attacks against Achterbahn-v2 which requires time complexity $O(2^{53})$ and $O(2^{52})$ keystream bits.

In this study, we propose a key-recovery attack on stream cipher Achterbahn-v2. Firstly, a more effective distinguishing attack is introduced and need more exact data complexity. Then we give the best parity check and decimation with remarkable biases. Lastly, a distinguisher is built. The attack against Achterbahn-v2 requires about $O(2^{51.6})$ keystream bits and $O(2^{53})$ time complexity. Therefore, Achterbahn-v2 is vulnerable to the linear distinguishing attack.

PRELIMINARIES

A distinguishing attack is a common attack on stream ciphers. Its main idea is to tell whether a sequence has been generated by the keystream generator or random generator. In linear distinguishing attacks, in order to distinguish the keystream from a random sequence, the attacker tries to construct a distinguisher and find statistical bias in the sequence that is obtained after a linear transform has been applied to the original sequence. Although distinguishing attack does not mean that a cipher is broken, it is still a potential weakness to a stream cipher. Up until now, it has been successfully used on HC-128 (Stankovski *et al*., 2012), Rakaposhi (Orumiehchiha *et al*., 2013), Helix (Shi *et al*., 2013), WG-7 (Orumiehchiha *et al*., 2012), RC4 (Bhateja and Din, 2013) and so on.

Let l(t) be a linear approximation of the combining output sequence z(t), if the probability:

$$p = Pr[l(t) = z(t)] = \frac{1}{2}(1+\varepsilon)$$

then ε is called the bias.

In other words, $\varepsilon = 2p-1$. From piling-up lemma, when n independent bits are xored the bias of the sum is given by ε^n.

Generally, distinguishing attacks need about $1/\varepsilon^2$ keystream bits to distinguish the keystream from a random sequence. However, Plasencia proposed a more effective distinguishing attack, it can be seen as a decoding problem and the received word can be seen as the result of the transmission of a codeword through a binary symmetric channel with cross-over probability p. When:

Corresponding Author: Li Shun-Bo, School of Science, Xi'an University of Architecture and Technology, Xi'an, 710055, Shaanxi, China

$$p = \frac{1}{2}(1+\varepsilon)$$

with bias $|\varepsilon| << 1$, the number of samples N required for decoding is:

$$N \approx \frac{2\lambda \cdot \ln 2}{\varepsilon^2}$$

where, λ is the initial states bits of the guessing registers. In this study we will use this method to improve the Plasencia's results.

DESCRIPTION ACHTERBAHN-V2

Achterbahn is a hardware-oriented binary additive synchronous stream cipher, designed by Gammel, Gottfert and Kniffler. The design of Achterbahn is based on the idea of a nonlinear combiner, but using NLFSRs instead of registers with linear feedback. Achterbahn comes in two variants, denoted reduced Achterbahn and full Achterbahn. In reduced Achterbahn the input bit to the Boolean function from shift register R_i is simply the output bit of R_i. In full Achterbahn the bit used in the Boolean function is a key dependent linear combination of a few bits in R_i. In this study, we will consider the full Achterbahn-v2.

Achterbahn-v2 uses 10 NLFSRs, denoted by $R_1, \ldots, R_{10}$. Their sizes are N_i = 19, 22, 23, 25, 26, 27, 28, 29, 31 and 32, respectively. All registers are primitive, hence the period is denoted by. At the every clock cycle, each register produces one output bits, denote, respectively by $x_1,\ldots, x_{10}$ (Fig. 1). Then the output keystream $z_t = S(x_1,\ldots, x_{10})$ and the Boolean output function S is defined as:

$$\begin{aligned} &S(x_1,\ldots,x_{10}) \\ &= x_1 \oplus x_2 \oplus x_3 \oplus x_9 \oplus G(x_4,x_5,x_6,x_7,x_{10}) \\ &\oplus (x_8 \oplus x_9)(G(x_4,x_5,x_6,x_7,x_{10}) \\ &\oplus H(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_{10})) \end{aligned}$$

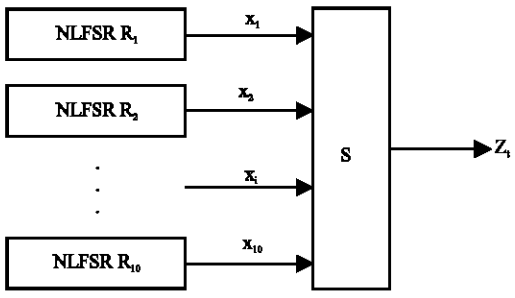

Fig. 1: Schematic of the achterbahn-v2

Where:

$$\begin{aligned} G(x_4,x_5,x_6,x_7,x_{10}) &= x_4(x_5 \vee x_{10}) \oplus x_5(x_6 \vee x_7) \\ &\oplus x_6(x_4 \vee x_{10}) \oplus x_7(x_4 \vee x_6) \oplus x_{10}(x_5 \vee x_7) \end{aligned}$$

and:

$$\begin{aligned} H(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_{10}) &= x_2 \oplus x_5 \oplus x_7 \oplus x_{10} \\ &\oplus (x_3 \oplus x_4)\overline{x_6} + (x_1 \oplus x_2)(x_3\overline{x_6} \oplus x_6(x_4 \oplus x_5)) \end{aligned}$$

The Boolean function S has resiliency 5, nonlinearity 448 and algebraic degree 4. Meanwhile, the output keystream length is limited to $O(2^{63})$.

Initialization: The internal state of Achterbahn is initialized from a secret key K of size 80 bits and from an initialization vector IV of length 80 bits. These steps are the following:

Step 1: The state of each register R_i is loaded in parallel with the first N_i bits of the key bits

Step 2: The registers are updated and the remaining key and the IV bits are xored with the input to the registers

Step 3: The least significant bit of each NLFSR is set to 1. This prevents the NLFSRs to be initialized with all zeros

Step 4: The registers are clocked such that the total number of clocks for each register in the initialization phase is 112+|IV|, where |IV| is the chosen length of the IV

Step 5: A key and IV dependent vector is prepared, defining which of the positions in the registers that are to be xored to form the input to the Boolean combining function

DISTINGUISHING ATTACK on ACHTERBAHN-v2

Linear approximation: For Achterbahn-v2, the Boolean output function S has resiliency 5, then any biased linear approximation has at least 6 terms and the corresponding parity-check (Canteaut and Naya-Plasencia, 2012) will also be the bias of the linear approximation raised to the power of the number of terms in the parity-check. So, the best linear approximations of the Boolean combining function S is:

$$l(x_1,\ldots,x_{10}) = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_8 \qquad (1)$$

Then, for instant t>0:

$$l(t) = x_1(t) \oplus x_2(t) \oplus x_3(t) \oplus x_4(t) \oplus x_6(t) \oplus x_8(t) \quad (2)$$

We have:

$$\Pr(z(t) = l(t)) = \frac{1}{2}(1 + 2^{-3})$$

so the bias is $\varepsilon = 2^{-3}$.

Parity-check and decimation: Decimation attack is one attack method of stream cipher which is presented by Filiol in 2000. The main idea is by selecting a decimation distance d such that the linear complexity L_d of the decimated sequence less than its length. So, we propose the following framework for key recovery attacks based on the dedicated sample decimation against the nonlinear combination keystream generators. We denote by T_i the period of register R_i, so $T_1 = 2^{19}$-1, $T_2 = 2^{22}$-1, $T_3 = 2^{23}$-1, $T_4 = 2^{25}$-1, $T_6 = 2^{27}$-1, $T_8 = 2^{29}$-1. The period of the sequence obtained by combining the registers R_1 and R_8 is the least common multiple of T_1 and T_8, is equal to lcm(T_1, T_8), denoted by T_1T_8 which is about 2^{48}-1. Similarly, T_3T_6 is about 2^{50}-1. Then, we build a parity check equation as follows:

$$\begin{aligned} L(t) &= l(t) \oplus l(t + T_1T_8) \oplus l(t + T_3T_6) \oplus l(t + T_1T_8 + T_3T_6) \\ &= \sum_{\tau \in <T_1T_8, T_3T_6>} l(t + \tau) \end{aligned} \quad (3)$$

Since, the period of the x_3, x_6 are T_3, T_6, respectively, $x_3(t + T_3T_6) = x_3(t + T_3) = x_3(t)$, $x_6(t + T_3T_6) = x_6(t + T_6) = x_6(t)$.

So, the expression $l(t) \oplus l(t + T_3T_6)$ does not contain any term in x_3 and x_6. Therefore, the terms containing the sequences x_1, x_3, x_6 and x_8 will disappear from L(t). Thus L(t) depends uniquely on the sequence x_2 and x_4.

From the Piling-up Lemma, adding 4 times the approximation has the effect of multiplying the bias 4 times. Let us denote:

$$\sigma(t) = \sum_{\tau \in <T_1T_8, T_3T_6>} z(t + \tau) \quad (4)$$

We have:

$$\Pr(\sigma(t) \oplus L(t)) = \frac{1 + \varepsilon^4}{2} = \frac{1}{2}(1 + 2^{-12})$$

This means that we need:

$$\frac{2 \cdot (N_4 - 1) \cdot \ln 2}{(2^{-12})^2} \approx 2^{29}$$

values of $\sigma(t) \oplus L(t)$ to detect this bias when we perform an exhaustive search on registers R_2 and R_4.

We now improve Plasencia's algorithm for computing the sum $\sigma(t) \oplus L(t)$ over all values of instant t. This algorithm has a low complexity to compute the 2^{29} parity-checks for all the initial states of the registers R_2 and R_4. Here we present a heuristic decimation attack on register R_2:

$$\begin{aligned} L(tT_2) &= l(tT_2) \oplus l(tT_2 + T_1T_8) \oplus l(tT_2 + T_3T_6) \\ &\quad \oplus l(tT_2 + T_1T_8 + T_3T_6) \\ &= x_4(tT_2) \oplus x_4(tT_2 + T_1T_8) \oplus x_4(tT_2 + T_3T_6) \\ &\quad \oplus x_4(tT_2 + T_1T_8 + T_3T_6) \oplus \gamma(t) \end{aligned} \quad (5)$$

Where:

$$\begin{aligned} \gamma(t) &= x_2(tT_2) \oplus x_2(tT_2 + T_1T_8) \oplus x_2(tT_2 + T_3T_6) \\ &\quad \oplus x_2(tT_2 + T_1T_8 + T_3T_6) \end{aligned} \quad (6)$$

is a constant. If the value of $\gamma(t) = 0$, then the probability of Eq. 4 is:

$$\frac{1}{2}(1 + 2^{-12})$$

If the value of $\gamma(t) = 1$, then the probability of Eq. 4 is:

$$\frac{1}{2}(1 - 2^{-12})$$

In any case, the total amount of keystream required in this approach will increase with a factor of T_2 that is:

$$\begin{aligned} N &= 2^{29}T_2 + T_1T_8 + T_3T_6 \\ &= 2^{29} \times (2^{22} - 1) + (2^{19} - 1) \times (2^{29} - 1) + (2^{23} - 1) \times (2^{27} - 1) \\ &\approx 2^{51.6} \end{aligned}$$

This value is less than the maximum length 2^{63}. So, the time complexity will be:

$$T = 2^{29} \times 2^{R_4 - 1} = 2^{29} \times 2^{24} = 2^{53}$$

as we only guess the initial state of register R_4.

We have some annotates in the following:

Note 1: Our results are better than Plasencia's. Moreover, if we use the general distinguishing model (Huang and Wu, 2007), we will get about $O(2^{49.8})$ keystream bits which is less than the Huang xiaoli's result $O(2^{50.4})$. So, our result is optimal

Note 2: We can select another parity-check, such as:

Table 1: Attacks complexities against the achterbahn-v2

References	Data complexity	Time complexity
Gammel	$O(2^{64})$	$O(2^{73})$
Hell	$O(2^{59})$	$O(2^{59})$
Plasencia	$O(2^{52})$	$O(2^{53})$
Ours	$O(2^{51.6})$	$O(2^{53})$

$$L(t) = l(t) \oplus l(t + T_2T_8) \oplus l(t + T_3T_6) \oplus l(t + T_2T_8 + T_3T_6)$$

and get the same results

Recovering the key: Once we have found the initial states of all the registers, we can invert all the initializing steps until the end of the second step. So, we use a meet-in-the-middle attack with time-memory trade-off to recover the key. The idea is that we do not need to invert all the clocking steps in the meet-in-the-middle attack, we first find the state of R_2 since R_4 is known. When both R_2 and R_4 are know, the expected number of key candidates becomes $2^{80-22-25} = 2^{33}$. All these key candidates can be tested and the correct key has been found. So, for Achterbahn-version 2 we need $O(2^{47})$ in memory and $O(2^{33})$ in computational complexity. From the above, the time complexity is much smaller than the distinguishing attack which we need to obtain the initial states of three registers. Then the time complexity of the total key-recover attack is the same one for the distinguishing attacks.

In the following Table 1, we can compare the results on Achterbahn-v2, our attack is more effective than any others.

CONCLUSION

Achterbahn-v2 was designed to resist linear approximations of the output functions and the amount of keystream that is allowed to be generated is limited to $O(2^{63})$. This study presents a linear approximation key- recover attack on stream cipher Achterbahn-v2 decreased the data complexity from $O(2^{52})$ to $O(2^{51.6})$. Our attack shows that Achterbahn-v2 can not counteract attack based on linear approximations and is not secure enough.

ACKNOWLEDGMENT

Project supported by the Scientific Research Foundation of Education of Department of Shaanxi Provincial Government of China (No. 2013JK0589) and Talent Education Research Fund of Xi'an University of Architecture and Technology (No. RC1221, RC1318, JC1321).

REFERENCES

Bhateja, A. and M. Din, 2013. ANN based distinguishing attack on RC4 stream cipher. Proceedings of the 7th International Conference on Bio-Inspired Computing: Theories and Applications, Volume 2, December 14-16, 2012, Indian Institute of Information Technology and Management, Gwalior, India, pp: 101-109.

Canteaut, A. and M. Naya-Plasencia, 2012. Parity-check relations on combination generators. IEEE Trans. Inform. Theory, 58: 3900-3911.

Coppersmith, D., S. Halevi and C. Jutla, 2002. Cryptanalysis of stream ciphers with linear masking. Proceedings of the 22nd 22nd Annual International Cryptology Conference on Advances in Cryptology-Crypto, August 18-22, 2002, California, USA., pp: 515-532.

Gammel, B.M., R. Gottfert and O. Kniffler, 2005. The Achterbahn stream cipher. In?neon Technologies AG St.-Martin-Str. 76 81541 Munich, Germany. http://www.ecrypt.eu.org/stream/ciphers/achterbahn/achterbahn.pdf

Gammel, B.M., R. Gottfert and O. Kniffler, 2005. Improved Boolean combining functions for Achterbahn. eSTREAM, the ECRYPT Stream Cipher Project, Report 2005/072, 2005.

Hell, M. and T. Johansson, 2007. Cryptanalysis of Achterbahn-version 2. Proceedings of the 13th International Workshop on Selected Areas in Cryptography, August 17-18, 2006, Montreal, Canada, pp: 45-55.

Hell, M., T. Johansson, L. Brynielsson and H. Englund, 2012. Improved distinguishers on stream ciphers with certain weak feedback polynomials. IEEE Trans. Inform. Theory, 58: 6183-6193.

Huang, X.L. and C.K. Wu, 2007. Cryptanalysis of achterbahn-version 1 and version 2. J. Comput. Sci. Technol., 22: 469-475.

Johansson, T., W. Meie and F. Muller, 2006. Cryptanalysis of achterbahn. Proceedings of the 13th International Workshop on Fast Software Encryption, March 15-17, 2006, Graz, Austria, pp: 1-14.

Naya-Plasencia, M., 2007. Cryptanalysis of achterbahn-128/80. Proceedings of the 14th International Workshop on Fast Software Encryption, March 26-28, 2007, Luxembourg, Luxembourg, pp: 73-86.

Orumiehchiha, M.A., J. Pieprzyk and R. Steinfeld, 2012. Cryptanalysis of WG-7: A lightweight stream cipher. Cryptography Commun., 4: 277-285.

Orumiehchiha, M.A., J. Pieprzyk, E. Shakour and R. Steinfeld, 2013. Security evaluation of Rakaposhi stream cipher. Proceedings of the 9th International Conference on Information Security Practice and Experience, May 12-14, 2013, Lanzhou, China, pp: 361-371.

Shi, Z., B. Zhang and D. Feng, 2013. Cryptanalysis of helix and phelix revisited. Proceedings of the 18th Australasian Conference on Information Security and Privacy, July 1-3, 2013, Brisbane, Australia, pp: 27-40.

Stankovski, P., S. Ruj, M. Hell and T. Johansson, 2012. Improved distinguishers for HC-128. Des. Codes Cryptography, 63: 225-240.