Cryptanalysis of WG-7: A Lightweight Stream Cipher

Mohammad Ali Orumiehchiha[1], Josef Pieprzyk[1], and Ron Steinfeld[2]*

[1] Center for Advanced Computing – Algorithms and Cryptography, Department of Computing,
Faculty of Science, Macquarie University, Sydney, NSW 2109, Australia
{mohammad.orumiehchiha,josef.pieprzyk}@mq.edu.au

[2] Clayton School of Information Technology
Monash University, Clayton VIC 3800, Australia
ron.steinfeld@monash.edu

Abstract. WG-7 is a stream cipher based on WG Stream Cipher and has been designed by Y. Luo, Q. Chai, G. Gong, and X. Lai in 2010. This cipher is designed for low cost and lightweight applications (RFID tags and mobile phones, for instance). This paper addresses cryptographic weaknesses of WG-7 Stream Cipher. We show that the key stream generated by WG-7 can be distinguished from a random sequence after knowing $2^{13.5}$ keystream bits and with a negligible error probability. Also, we investigate the security of WG-7 against algebraic attacks. An algebraic key recovery attack on this cipher is proposed. The attack allows to recover both the internal state and the secret key with the time complexity about 2^{27}.

Keywords: WG-7 Stream Cipher, Cryptanalysis, Key Recovery Attack, Distinguishing Attack, WG Stream cipher.

1 Introduction

WG-7 [10] is a fast lightweight stream cipher whose design has been inspired by the family of WG stream ciphers [12]. The original WG is a synchronous stream cipher submitted to the ECRYPT call. Both WG-7 and WG are hardware-oriented stream ciphers that use a word-oriented linear feedback shift register (LFSR) and a filter function based on the Welch-Gong (WG) transformation [8]. The structure of WG-7 is similar to the WG stream cipher. Both ciphers use LFSRs and filtering functions, however, WG works in $GF(2^{29})$ but WG-7 in $GF(2^7)$. WG-7 uses a 80-bit secret key and a 81-bit initial vector (IV). WG-7 works as follows. First the secret key and IV are used to initialise the internal state of the cipher LFSR. Next, the LFSR with its nonlinear function is clocked 46 times. After this initialisation procedure the cipher generates an appropriate string of keystreams that is used for encryption.

We assume that the initialisation procedure of WG-7 is performed as prescribed. Consequently, the internal state consists of 161 bits. Note that the security level claimed by the designers is 80 bits. The cipher has been designed for encryption in resource restricted environments such as RFID applications, mobile phones and smart cards. The authors of the cipher analysed the design and concluded that WG-7 [10] is secure against time/memory/data tradeoff attacks, differential attacks, algebraic attacks and correlation attacks .

This paper is organized as follows. Section 2 describes a brief description of the keystream generator of WG-7 stream cipher. Section 3 deals with a cryptographic weaknesses of the algorithm, which leads to our distinguishing and key recovery attacks.

2 Description of WG-7

The structure of the WG-7 stream cipher is illustrated in Figure 1. It consists of a 23-word LFSR, where a single word is 7-bit long. The filter function WG is a nonlinear function defined for 7 boolean variables (a word). The word is an element of $\mathbb{F}_{2^7}$, where the finite field $\mathbb{F}_{2^7}$ is defined by the primitive

* This work was done while R.S. was with Macquarie University.

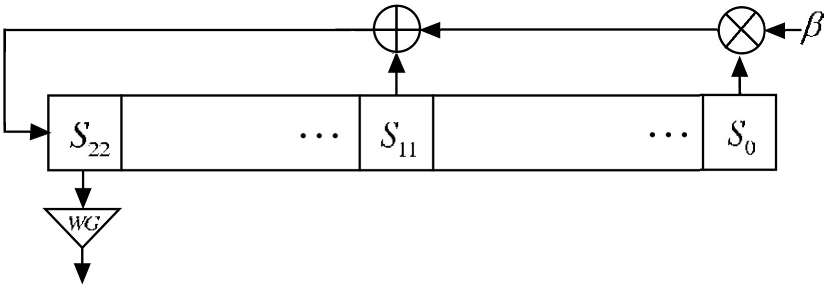

Fig. 1. The WG-7 Stream Cipher Scheme

polynomial $g(x) = x^7 + x + 1$ over $GF(2)$. The characteristic polynomial of LFSR is primitive over $\mathbb{F}_{2^7}$ and is given by:

$$f(x) = x^{23} + x^{11} + \beta, \tag{1}$$

where β is a root of $g(x)$. The nonlinear filter function WG(x) denoted in Figure 1 as WG is a transformation $\mathbb{F}_{2^7} \to \mathbb{F}_2$ and defined below:

$$WG7(x) = f(x) = Tr(x^3 + x^9 + x^{21} + x^{57} + x^{87}), \quad x \in \mathbb{F}_{2^7}. \tag{2}$$

3 Cryptanalysis of WG-7

In this section, we describe our two attacks for WG-7. The first attack distinguishes the WG-7 stream cipher from the random one. The attack exploits a bias in a linear approximation of the nonlinear filter function. The second attack is a variant of the fast algebraic attack. It permits to recover not only the internal state of the cipher but also the secret key.

3.1 Distinguishing Attack for WG-7

The WG-7 stream cipher has a relatively simple structure. The main component is LFSR that generates words that are later transformed in a nonlinear fashion by the filter function. The only nonlinear component in the cipher is the filter function. It seems to be a quite reasonable idea to check how well the filter function can be approximated by an affine function. In other words, we are looking for an affine function that approximates the filter function as close as possible (the best linear approximation). If we apply the well-known Walsh-Hadamard transform to the filter function, then we can obtain such linear approximation. Denote it by $\Gamma \cdot (x_0, ..., x_6) + \alpha$, where x_i is the i-th bit of the word x, the sign " $\cdot$ " is the inner product, Γ ($\Gamma \in \mathbb{F}_{2^7}$) is a constant (a vector of 7 binary constants) and α is a binary constant. In case of WG-7, we have found out that there are seven affine functions, which are the best linear approximation (one of such functions is $1 + x_0 + x_1 + x_4$). As the nonlinearity of the filter function is 52, we can find the following probability

$$Pr(WG(x) = (\Gamma \cdot x + \alpha)) = \frac{2^7 - 52}{2^7} = 0.59375 \tag{3}$$

From Equation (1), the following recursive relation can be derived:

$$S_{i+23} = S_{i+11} \oplus \beta \cdot S_i. \tag{4}$$

Consequently, we need to find the best linear approximation of the relation given below:

$$WG(S_{i+23}) \oplus WG(S_{i+11}) \oplus WG(S_i) = 0. \tag{5}$$

Remark 1: The Piling up Lemma cannot be used to compute the bias of Equation (5) because the input variables $(S_{i+23}, S_{i+11}, S_i)$ are not independent. In particular, S_{i+23} is correlated with other variables by Equation (1). In addition, $\beta \cdot S_i$ in Equation (4) is linear transformation of S_i. The precise linear relations are given below

$$\beta \cdot S_i = \beta \cdot (s_0^i, s_1^i, s_2^i, s_3^i, s_4^i, s_5^i, s_6^i) = \begin{vmatrix} s_1^i \oplus s_3^i \oplus s_4^i \\ s_2^i \\ s_2^i \oplus s_5^i \\ s_4^i \\ s_1^i \oplus s_2^i \\ s_6^i \\ s_0^i \oplus s_1^i \oplus s_2^i \oplus s_3^i \oplus s_4^i \oplus s_5^i \oplus s_6^i \end{vmatrix}^T \tag{6}$$

We need to determine the exact value of the bias ε in the following probability:

$$Pr(WG(S_{i+23}) \oplus WG(S_{i+11}) \oplus WG(S_i) = 0) = 0.5 + \varepsilon \tag{7}$$

One method to compute the bias in Equation (7) is as follows. We consider the bias between three output bits at clocks i, $i + 11$ and $i + 23$. So we get

$$\begin{aligned} z_{i+23} \oplus z_{i+11} \oplus z_i = & \\ & = WG(S_{i+23}) \oplus WG(S_{i+11}) \oplus WG(S_i) \\ \overset{From\ Eq.\ 4}{\Longrightarrow} & = WG(S_{i+11} \oplus \beta \cdot S_i) \oplus WG(S_{i+11}) \oplus WG(S_i) \end{aligned} \tag{8}$$

Observe that Equation (8) is a boolean function with 14 input variables (instead of 21 variables) and a single bit output. In other words, S_{i+23} depends on S_i and S_{i+11} based on Equations (4) and (6). Let $F \cdot GF(2^{14}) \rightarrow GF(2)$ be a non-linear boolean function in form of

$$F(S_i, S_{i+11}) = WG(S_{i+11} \oplus \beta . S_i) \oplus WG(S_{i+11}) \oplus WG(S_i) \tag{9}$$

Now, we focus on $F(s_0^i, s_1^i, ..., s_6^i, s_0^{i+11}, s_1^{i+11}, ..., s_6^{i+11})$ that is an unbalanced boolean function, where

$$Pr(F(s_0^i, s_1^i, ..., s_6^i, s_0^{i+11}, s_1^{i+11}, ..., s_6^{i+11}) = 0) = \frac{1}{2} - 2^{-7.145} \tag{10}$$

The relation given by Equation (9) defines a distinguisher, which is able to tell apart the output of the stream cipher from a truly random cipher with the probability expressed by Equation (10). The interesting question is: are there better biases to mount a distinguishing attack? We will discuss possible answers in the remaining part of this section.

Better biases: In the previous section, we have found a linear approximation leading us to a distinguishing attack. One would wonder whether it is possible to find a better linear approximation so that the bias is closer to the maximal value of 0.5.

Let us explore this issue in more detail. Repeated squaring of the characteristic polynomial of LFSR (see Equation 1) gives other linear recurrence polynomials. If we use the exponent 2^7, we get

$$x^{23 \cdot 2^7} + x^{11 \cdot 2^7} + \beta^{2^7} = 0 \tag{11}$$

Since $\beta = \beta^{2^7}$, $\beta \in \mathbb{F}_{2^7}$, the summation of Equations (1) and (11) gives:

$$x^{23 \cdot 2^7} + x^{11 \cdot 2^7} + x^{23} + x^{11} = 0 \tag{12}$$

$$\overset{divided\ by\ x^{11}}{\Longrightarrow} x^{23 \cdot 2^7 - 11} + x^{11 \cdot 2^7 - 11} + x^{12} + 1 = 0 \tag{13}$$

It means that the attacker can derive a bitwise linear equation, which is valid for the internal state of LFSR. Similar to the previous subsection, the function F can be built as follows:

$$\begin{aligned} & z_{i+23 \cdot 2^7 - 11} \oplus z_{i+11 \cdot 2^7 - 11} \oplus z_{i+12} \oplus z_i \\ & = WG(S_{i+23 \cdot 2^7 - 11}) \oplus WG(S_{i+11 \cdot 2^7 - 11}) \oplus WG(S_{i+12}) \oplus WG(S_i) \\ & = WG(S_{i+11 \cdot 2^7 - 11} \oplus S_{i+12} \oplus S_i) \oplus WG(S_{i+11 \cdot 2^7 - 11}) \oplus WG(S_{i+12}) \oplus WG(S_i) \end{aligned} \tag{14}$$

Equation (14) can be considered as a boolean function with 21 input variables (instead of 28 variables) and a single bit output. The boolean function $F : GF(2^{21}) \to GF(2)$ is an unbalanced boolean function, where

$$Pr(F(S_{i+11\cdot 2^7-11}, S_{i+12}, S_i) = 0) = \frac{1}{2} + 2^{-6.78} \tag{15}$$

The required data: Now, we explain the amount of output sequences required to distinguish WG-7 from a truly random cipher. The following theorem determines the required length of keystream needed to distinguish between two random sequences, where one is uniform (both binary values occur with $\frac{1}{2}$) and the other is biased (one value occurs with $\frac{1}{2}(1+\varepsilon)$) - see [11].

Theorem 1. *Given two binary random sequences, where the first is uniform and the other is biased, i.e. one binary value occurs with the probability $\frac{1}{2}(1+\varepsilon)$ while the other with the probability $\frac{1}{2}(1-\varepsilon)$. Then we need to observe $O(\frac{1}{\varepsilon^2})$ bits in order to distinguish the two distributions with a non-negligible probability of success.*

In this case, the amount of data required for proposed distinguishing attack is $2^{13.56}$ bits. This amount of data can be collected from consecutive (or non-consecutive) keystream and even from one session key or different session keys in various times.

The result of the implemented distinguishing attack on WG-7 stream cipher are shown in Table 1. We have repeated the experiment 1000 times to compute the success rate of distinguishing attack with different lengths of output sequences.

Table 1. Experimental results for applying distinguishing attack on WG-7

	Used Data (bits)	Success Rate
1	2^9	%68
2	$2^{9.8}$	%75
3	$2^{10.3}$	%85
4	$2^{11.5}$	%90
5	$2^{13.5}$	%99.99

3.2 Key Recovery Attack on WG-7

In this section, we apply an algebraic analysis to recover the initial state of the cipher and consequently the secret key. Our attack can recover internal states of WG-7 and then attacker is able to clock the LFSR backward and find the secret key correctly. The designers of the WG-7 stream cipher have claimed that there is no algebraic attack with the complexity smaller than the exhaustive search and with the data complexity smaller than 2^{24} of consecutive keystream bits. The idea of our attack is as follows. Let $L : GF(2^{161}) \to GF(2^{161})$ be a multivariate linear transformation that corresponds to the linear transformation defined by a single clock. This transformation is done on the whole state of 23 registers each holding 7 bits ($23 \cdot 7 = 161$).

Let z_t, $t = 0, 1, 2, ...$ be the keystream generated by the cipher after running the state initialization algorithm of WG-7. Assume also that f is the non-linear filter function WG illustrated in Figure 1. We consider f as a non-linear map defined from $GF(2^7) \to GF(2)$. As the output bit is calculated on the contents of the last register or bits from 154 to 160, we denote this by

$$f(T(s_0, ..., s_{160})),$$

where $T(s_0, ..., s_{160})$ extracts the 7-bit content of the last register. So, we can establish the following system of relations for the cipher:

$$\begin{cases} z_0 = f(T(s_0, ..., s_{160})) \\ z_1 = f(T(L(s_0, ..., s_{160}))) \\ ... \\ z_t = f(T(L^t(s_0, ..., s_{160}))) \end{cases} \tag{16}$$

where $f(T(L^t(s_0, ..., s_{160})))$ indicates the output keystream at the clock t, generated by the stream cipher. Now, the cryptanalytic problem can be converted into the problem of solving a system of nonlinear equations (see [1, 3, 4, 6, 7]).

Algebraic attack on WG-7: The simplest scenario to solve System (16) is known as the linearization technique [[5], [7]]. The function f is of degree 5. The number N of monomials of degree smaller or equal to 5 is

$$N = \sum_{i=1}^{5} \binom{161}{i} \approx \binom{161}{5} = 2^{29.65}.$$

Each of these monomials can be considered as a new variable and then the attacker can solve the non-linear system with $\approx 2^{29.65}$ equations and time complexity $\approx 2^{29.65 \times log_2^7}$ by the Gaussian elimination method. Consequently, the complexity of the attack is larger than the exhaustive key search.

The important idea to improve the efficiency of the above attack is to reduce the degree of the equations. To this end, the attacker tries to find an annihilator function so that $f \cdot g = 0$ and $\deg g < \deg f$. The steps to apply the attack can be described as follows:

1. Finding an annihilator g of f or $f \oplus 1$ with a low degree d.
2. Given multivariate equations of a low degree d on the initial state bits, there are $N = \sum_{i=1}^{d} \binom{n}{i}$ monomials of degree no bigger than d, where n is the length of internal state. Hence by the linearization method, time complexity to solve the non-linear system is $N^{log_2^7}$. The memory complexity of the attack is about N.

The algebraic normal form (ANF) of f is as follows:

$$\begin{aligned} & f(x_1, ..., x_7) = x_1 + x_1x_3 + x_2x_3 + x_4 + x_1x_4 + x_2x_4 + \\ & x_1x_2x_4 + x_3x_4 + x_1x_3x_4 + x_1x_2x_3x_4 + x_1x_3x_5 + x_4x_5 + x_1x_2x_4x_5 + \\ & x_1x_2x_3x_4x_5 + x_6 + x_2x_6 + x_1x_2x_6 + x_1x_2x_3x_6 + x_1x_2x_4x_6 + x_1x_2x_3x_4x_6 + \\ & x_1x_5x_6 + x_3x_5x_6 + x_1x_4x_5x_6 + x_3x_4x_5x_6 + x_7 + x_2x_7 + x_1x_2x_7 + x_2x_3x_7 + \\ & x_1x_4x_7 + x_1x_2x_4x_7 + x_1x_2x_3x_4x_7 + x_5x_7 + x_1x_5x_7 + x_1x_3x_5x_7 + x_1x_2x_3x_5x_7 + \\ & x_2x_4x_5x_7 + x_2x_3x_4x_5x_7 + x_6x_7 + x_1x_2x_6x_7 + x_1x_3x_6x_7 + x_1x_2x_3x_6x_7 + \\ & x_2x_4x_6x_7 + x_1x_3x_4x_6x_7 + x_2x_3x_4x_6x_7 + x_5x_6x_7 + x_2x_5x_6x_7 + x_1x_2x_5x_6x_7 + \\ & x_2x_3x_5x_6x_7 + x_1x_4x_5x_6x_7 + x_3x_4x_5x_6x_7. \end{aligned}$$

The best annihilator is of the form:

$$\begin{aligned} g(x_1, ..., x_7) = 1 & + x_1 + x_3 + x_1x_2x_3 + x_4 + x_1x_4 + x_2x_4 + x_1x_2x_4 + x_3x_4 + x_1x_3x_4 + x_2x_3x_4 + \\ & x_1x_3x_5 + x_4x_5 + x_1x_4x_5 + x_3x_4x_5 + x_6 + x_1x_6 + x_2x_6 + x_1x_2x_6 + x_3x_6 + \\ & x_2x_3x_6 + x_7 + x_3x_7 + x_1x_3x_7 + x_2x_3x_7 + x_4x_7 + x_2x_4x_7 + \\ & x_3x_4x_7 + x_3x_5x_7 + x_4x_5x_7 + x_6x_7 + x_1x_6x_7 + x_2x_6x_7 + x_3x_6x_7. \end{aligned}$$

It means that the attacker can reduce the degree of the relations to 3 and solve them with time complexity $\approx \binom{161}{3}^{log_2^7} = 2^{54.36}$ and memory complexity $\binom{161}{3} = 2^{19.38}$. It is obvious that the designers of WG-7 have ignored this attack, which breaks the cipher with the memory complexity smaller than 2^{24}.

Improved Attack on WG-7 Fast algebraic attacks (see [3, 7, 9]) on stream ciphers that use LFSR are based on equations of type $zX^e + X^d$ with $e < d$. This is a shorthand to describe that at least one equation of type

$$z \cdot g(s_0, ..., s_{n-1}) + h(s_0, ..., s_{n-1}) = 0 \tag{17}$$

exists, where g and h are some multivariate polynomials of degree e and d $(e < d)$ respectively, and $z = f(s_0, ..., s_{n-1})$. The attack can be summarized as follows:

$$\sum_{i=t}^{t+D} \alpha_{t+i}.z_i.g(T(L^i(s_0, ..., s_{160}))) \tag{18}$$

for some linear combination $(\alpha_0, ..., \alpha_{D1}) \in GF(2)^D$, where $D = \sum_{i=1}^{d} \binom{n}{i}$. The same equation applies to each window of D consecutive steps and we will write it E times, for E overlapping intervals, with $E = \sum_{i=1}^{e} \binom{n}{i}$. This is because we need to get the final system of the degree e that is solvable by linearisation (with the complexity $E^{log_2^7}$). This approach is discussed in [1, 2, 7, 9]. The steps of our improved attack are summarized as follows:

1. Relation step: One searches g and h with small degrees such that $f \cdot g = h$. The lower bound on the complexity of solving a linear system with $D + E$ equations is $O((D + E)^{log_2^7})$. In general one considers $e < d$.
2. Pre-computation step: Computation of linear relations to eliminate the terms of degrees greater than e in the equations. This needs $2D$ bits of stream bits with the complexity $O(Dlog^2(D))$.
3. Substitution step: One eliminates the monomials of degree greater than e. The time complexity is $O(E^2D)$ [2] but by DFT [9] it can be further reduced to $O(E.D.log(D))$.
4. Solving step: One solves the system with E linear equations in $O(E^{log_2^7})$.

In case of WG-7, to apply the fast algebraic attack, we have found the boolean functions g and h and they are:

$$\begin{aligned}
&g(x_1, ..., x_7) = 1 + x_1 + x_3 + x_7.\\
&h(x_1, ..., x_7) = x_1x_2x_3 + x_4 + x_1x_4 + x_2x_4 + x_1x_2x_4 + x_3x_4 +\\
&x_1x_3x_4 + x_2x_3x_4 + x_1x_3x_5 + x_4x_5 + x_1x_4x_5 +\\
&x_3x_4x_5 + x_6 + x_1x_6 + x_2x_6 + x_1x_2x_6 + x_3x_6 + x_2x_3x_6 +\\
&x_3x_7 + x_1x_3x_7 + x_2x_3x_7 + x_4x_7 + x_2x_4x_7 + x_3x_4x_7 +\\
&x_3x_5x_7 + x_4x_5x_7 + x_6x_7 + x_1x_6x_7 + x_2x_6x_7 + x_3x_6x_7.
\end{aligned}$$

The data complexity of the fast algebraic attack on WG-7 is $\binom{161}{d} = \binom{161}{3}$ and the time complexity is approximately $\binom{161}{e}^{log_2^7} \approx \binom{161}{1}^{2.807}$. Table 2 summarizes the results of our attacks.

Table 2. Comparison of different algebraic attacks against WG-7

	Attack type	n	d	e	Time Complexity	Data Complexity	Memory	Pre-Computation
1	Trivial Attack	161	5	-	$2^{83.02}$	$2^{29.65}$	-	-
2	Algebraic Attack	161	3	-	$2^{54.36}$	$2^{19.38}$	-	-
3	Fast Algebraic Attack	161	1	3	$2^{26.73}$	$2^{19.38}$	$2^{14.66}$	$2^{26.87}$

3.3 Conclusions

In this paper, the security of the WG-7 stream cipher has been investigated. We have shown that distinguishing attack works with a high probability of success after observing $2^{13.5}$ keystream bits. Additionally, the key recovery attack has been described that can recover the secret key with the time complexity about 2^{27} and the data complexity $2^{19.38}$. The presented results have proved that the WG-7 stream cipher is not secure and therefore, it is not recommended to be used.

References

1. F. Armknecht. Improving fast algebraic attacks. In *FSE*, pages 65–82, 2004.
2. F. Armknecht and G. Ars. Introducing a new variant of fast algebraic attacks and minimizing their successive data complexity. In *Mycrypt*, pages 16–32, 2005.
3. N. Courtois and W. Meier. Algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology - EUROCRYPT 2003,Warsaw, Poland, 2003, Proceedings*, pages 345–359. Springer, 2003.
4. N. Courtois and J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In *ASIACRYPT*, pages 267–287, 2002.
5. N. T. Courtois. Higher order correlation attacks, xl algorithm and cryptanalysis of toyocrypt. In *ICISC 2002*, pages 182–199. Springer-Verlag, 2002.
6. N. T. Courtois. Algebraic attacks on combiners with memory and several outputs. In *Proc. of ICISC04*, pages 3–20, 2004.
7. N. T. Courtois and W. Meier. Fast algebraic attacks on stream ciphers with linear feedback. In *Crypto 2003, LNCS 2729*, pages 177–194. Springer.
8. G. Gong and A. M. Youssef. Cryptographic properties of the welch-gong transformation sequence generators. *IEEE Transactions on Information Theory*, 48(11):2837–2846, 2002.
9. P. Hawkes and G. G. Rose. Rewriting variables: The complexity of fast algebraic attacks on stream ciphers. In *CRYPTO*, pages 390–406, 2004.
10. Y. Luo, Q. Chai, G. Gong, and X. Lai. A lightweight stream cipher wg-7 for rfid encryption and authentication. In *GLOBECOM*, pages 1–6, 2010.
11. I. Mantin, , and A. Shamir. A practical attack on broadcast rc4. In *Proc. of FSE01*, pages 152–164. Springer-Verlag, 2001.
12. Y. Nawaz and G. Gong. Wg: A family of stream ciphers with designed randomness properties. *Inf. Sci.*, 178(7):1903–1916, 2008.