Paper

Remarks on improved inversion attacks on nonlinear filter generators

Anna Górska and Karol Górski

Abstract — The subject of this paper are inversion attacks on stream ciphers (nonlinear filter generators), which were first introduced by Golić [3] and extended by Golić, Clark and Dawson [4]. These original attacks have computational complexity $O(2^M)$, where M is the so-called "memory size" – distance between outer taps to filter function. In [6] we have proposed improved inversion attacks which have computational complexity $O(2^{r-m})$, where r denotes the length of the shift register and m denotes the largest gap between cells with taps to filter function or to connection polynomial. In this paper we describe further extension of our previous results obtained by considering shifts of the feedback polynomial which maximise the largest gap between cells with taps to filter function or to connection polynomial. We show that the previously proposed set of design criteria [3, 6] does not prevent the new version of improved inversion attack and we propose an additional criterion based on the relationship between positions of taps to filter function and positions of taps to the multiples of the connection polynomial.

Keywords — stream cipher, shift register, nonlinear filter generator, inversion attack.

1. Introduction

Despite the growing importance of block ciphers, symmetric stream ciphers are still one of the fundamental tools in modern cryptography. Most designs are based on linear feedback shift registers (LFSR) combined by nonlinear boolean functions or filtered by nonlinear boolean functions (so-called nonlinear filter generators). Different variants exist: clock-controlled systems, multiplexed systems, memory combiners and decimated generators. Our work focuses on nonlinear filter generators (NFG) illustrated in Fig. 1. NFG can be used on its own [1] or as a building block in more complex generators.

Unfortunately there are no known, practical constructions of stream ciphers which offer unconditional security or provable computational security (the one time pad, an unconditionally secure stream cipher, cannot be regarded as practical). In practice, evaluation of the security of these ciphers is heuristic. Among the most powerful classes of attacks on stream ciphers which have to be considered are fast correlation attacks (beginning from [7]) and conditional correlation attacks [1]. The first class was initially used to attack combination generators but recently was successfully used to attack NFG [8]. In the second class best results were achieved by Golić [3] who introduced the inversion attacks (IA), which are the most powerful attacks on nonlinear filter generators. In the same paper Golić presented a set of design criteria for nonlinear filter generators which, when respected, should ensure large period, high linear complexity and good statistical properties of the output sequence as well as resistance to fast correlation attacks, conditional correlation attacks and inversion attacks.

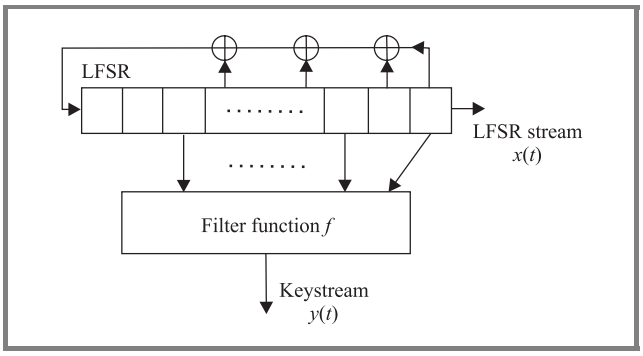

Fig. 1. Nonlinear filter generator.

In [6] we have introduced improved inversion attacks (IIA), which can have significantly lower computational complexity in comparison to basic inversion attacks. In this paper we propose an extension of this attack and a modification of the set of design criteria in order to prevent this new attack.

2. Notation and definitions

Let r be LFSR length, n $(n \leq r)$ – denote the number of non-degenerate input variables to filter function $f(z_1, \ldots, z_n)$ and $\gamma = (\gamma_i)_{i=1\ldots n}$ denote the tapping sequence, an increasing sequence of integers specifying positions of inputs to filter function, such that $\gamma_1 = 0$ and $\gamma_n \leq r-1$. Let $M = \gamma_n - \gamma_1$ denote the memory of the filter function. Let $x = (x(t))_{t=-r\ldots\infty}$ be a binary maximum-length sequence $(x = (x(t))_{t=-r+1\ldots 0}$ denotes LFSR initial state). Then the output sequence $y = (y(t))_{t=0\ldots\infty}$ is computed as:

$$y(t) = f\left(x(t-\gamma_1), \ldots, x(t-y_n)\right). \qquad (1)$$

In [3] it was proved that a filter function of one of the following forms:

$$f(z_1, \ldots, z_n) = z_1 \oplus g(z_2, \ldots, z_n) \tag{2}$$

or

$$f(z_1, \ldots, z_n) = z_n \oplus h(z_1, \ldots, z_{n-1}) \tag{3}$$

will produce (independently of the tapping sequence) a purely random output, given a purely random input, thus making the generator resistant to conditional correlation attacks.
In this case $y(t)$ takes the following form:

$$y(t) = x(t-\gamma_1) \oplus g(x(t-\gamma_2), \ldots, x(t-\gamma_n)), \tag{4}$$

or

$$y(t) = x(t-\gamma_n) \oplus h(x(t-\gamma_1), \ldots, x(t-\gamma_{n-1})). \tag{5}$$

Depending on the form of the function Golić proposed the forward inversion attack and the backward inversion attack, respectively. The average computational complexity of the attacks is $\boldsymbol{O}(2^{M-1})$ and the worst case complexity is $\boldsymbol{O}(2^M)$.

3. Inversion attack

The objective of the attack is to reconstruct the initial state of the LFSR, having a segment of keystream sequence, given the LFSR feedback polynomial, nonlinear filter function f and the tapping sequence γ. If the filter function is of the form (4) forward inversion attack is applied, which is given by the algorithm below.

Algorithm 1 (forward inversion attack):

1. Assume (not previously checked) M bits $\big(x(t)\big)_{t=-M\cdots-1}$ of unknown initial memory state.
2. By using (4), generate $\big(x(t)\big)_{t=r-M-1\ldots0}$ from a known segment $\big(y(t)\big)_{t=r-M-1\ldots0}$ of output sequence.
3. By using LFSR linear recursion, generate $\big(x(t)\big)_{t=r-M\ldots N-1}$ from first r bits of $\big(x(t)\big)_{t=-M\ldots r-M-1}$.
4. By using (1), compute $\big(y'(t)\big)_{t=r-M\ldots N-1}$ from $\big(x(t)\big)_{t=r-2M\ldots N-1}$ and compare with the known $\big(y(t)\big)_{t=r-M\ldots N-1}$. If they are the same then accept assumed initial memory state and stop. Otherwise go to 1.

When the filter function is of the form (5) backward inversion attack is applied, which is given by Algorithm 2.

Algorithm 2 (backward inversion attack):

1. Assume (not previously checked) M bits $\big(x(t)\big)_{t=-M-1\ldots0}$ of unknown initial memory state.
2. By using (5), generate $\big(x(t)\big)_{t=r-M-1\ldots0}$ from a known segment $\big(y(t)\big)_{t=r-M-1\ldots0}$ of output sequence.
3. By using LFSR linear recursion, generate $\big(x(t)\big)_{t=r-M\ldots N-1}$ from first r bits of $\big(x(t)\big)_{t=-M\ldots r-M-1}$.
4. By using (1), compute $\big(y'(t)\big)_{t=r-M\ldots N-1}$ from $\big(x(t)\big)_{t=r-2M\ldots N-1}$ and compare with the known $\big(y(t)\big)_{t=r-M\ldots N-1}$. If they are the same then accept assumed initial memory state and stop. Otherwise go to 1.

4. Improved inversion attack

The difference between the basic inversion attack and our first proposal of the improved inversion attack [6] relies on a modification of Steps 1 and 2. To make further improvement we will include an additional preprocessing phase in which we will find the largest gap between cells with taps to filter function and cells with taps to connection polynomial multiples (shifts of the polynomial). This idea was first suggested by Golić [5]. In Step 1 instead of guessing M bits of initial state we guess $r-m$ bits, where m denotes the size of the largest gap between cells of LFSR which have taps to filter function or to multiples of connection polynomial (when the largest such gap is between cells j and k, where $j<k$, then $m=k-j$). The average computational complexity of the improved attack is $\boldsymbol{O}(2^{r-m-1})$ and the worst case complexity is $\boldsymbol{O}(2^{r-m})$. The algorithm of the new attack is as follows.

Algorithm 3 (improved inversion attack with preprocessing phase):

1. (Preprocessing phase). Find the largest gap between cells of LFSR with taps to connection polynomial multiples and cells with taps to filter function. Denote the outer cells of the gap by k and $k-m$.

Rest of the attack is identical to improved inversion attack presented in [6]:

2. Assume (not previously checked) $r-m$ bits $\big(x(t)\big)_{t=-r+1\cdots-k-1,\,-k+m\ldots0}$ of unknown initial memory state.
3. By using (4), generate $\big(x(t)\big)_{t=-k\cdots-k+m-1}$ from a known segment $\big(y(t)\big)_{t=0\ldots m-1}$ of output sequence and the connection polynomial.
4. By using LFSR linear recursion, generate sequence $\big(x(t)\big)_{t=r-m\ldots N-1}$ from first r bits $\big(x(t)\big)_{t=-m\ldots r-m-1}$.
5. By using (1), compute $\big(y'(t)\big)_{t=r-m\ldots N-1}$ from $\big(x(t)\big)_{t=r-2m\ldots N-1}$ and compare with the known $\big(y(t)\big)_{t=r-m\ldots N-1}$. If they are the same then accept assumed initial memory state and stop. Otherwise go to 1.

We illustrate this attack by examples.

Example 1 (improved inversion attack (IIA) – Figs. 2 and 3). Let the connection polynomial[1] be:

$$p(x) = x^{130} + x^{63} + x^{56} + x^{35} + x^{28} + \\ + x^{21} + x^7 + x^3 + 1,$$

the tapping sequence: $\gamma = (127, 63, 31, 15, 7, 3, 1, 0)$, and let the filter function be linear in the last variable. (If the filter function would not be linear in any variable, which delimits the largest gap, we should apply an inversion attack with branching [4]). In the basic inversion attack the cryptanalyst needs to guess 127 bits, which gives computational complexity of $\boldsymbol{O}(2^{127})$ and makes the attack infeasible. In the improved version of the attack we only need to guess 66 bits $\big(x(t)\big)_{t=-129,-128,-63,\ldots,0}$ which gives the expected attack runtime of 2^{66} steps.
Let us describe how our attack works in Step 2.

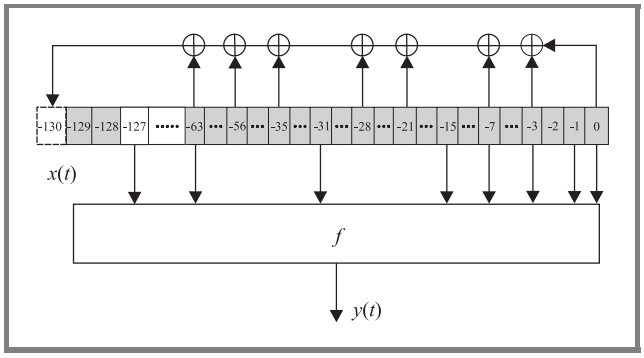

Fig. 2. Improved inversion attack on NFG (guessed (known) cells are filled with grey colour).

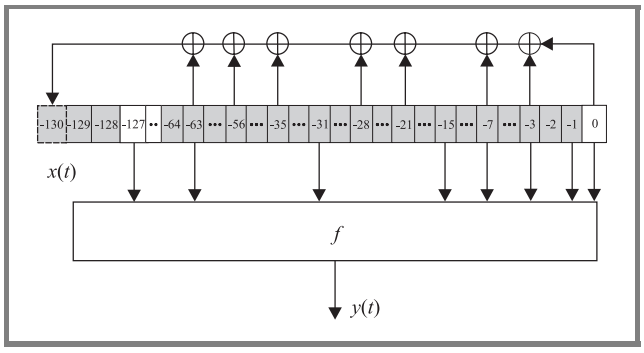

Fig. 3. Improved inversion attack on NFG cont. (guessed (known) cells are filled with grey colour).

First we calculate (identically as in IA) $x(-127)$:

$$x(-127) = y(63) \oplus g\big(x(0), x(-1), x(-3), x(-7), \\ x(-15), x(-31), x(-63)\big), \qquad (6)$$

[1]This feedback polynomial is very sparse, chosen to simplify the example.

then we clock backward the register state (so the content of cell i moves to cell $i-1$ and, after clocking we know $\big(x(t)\big)_{t=-130,-129,-128,-64\cdots-1}$).
We can calculate $x(0)$ from the connection polynomial:

$$x(0) = x(-3) \oplus x(-7) \oplus x(-21) \oplus x(-28) \oplus x(-35) \oplus \\ \oplus x(-56) \oplus x(-63) \oplus x(-130)$$

and then again calculate $x(-127)$ from the knowledge of output stream and filter function:

$$x(-127) = y(62) \oplus g\big(x(0), x(-1), x(-3), x(-7), \\ x(-15), x(-31), x(-63)\big).$$

Then again we clock the register state left (after which we know $\big(x(t)\big)_{t=-130,-129,-128,-65\cdots-1}$), calculate $x(0)$ from a connection polynomial, and so on. We continue this procedure until the LFSR state is reconstructed. Then we follow testing Steps 3 and 4.

Example 2 (improved inversion attack with preprocessing phase – Figs. 4 and 5). Let the connection polynomial be of the following form:

$$p(x) = x^{130} + x^{66} + x^{65} + x^{64} + x^{34} + x^{33} + x^{32} + x^{18} + \\ + x^{17} + x^{16} + x^{10} + x^9 + x^8 + x^6 + x^5 + x^3 + x^2 + x + 1,$$

the tapping sequence: $\gamma = (127, 63, 31, 15, 7, 3, 1, 0)$ identical to that in Example 1, and let again the filter function be linear in the last variable. The computational complexity of basic inversion attack is again $\boldsymbol{O}(2^{127})$, the complexity of IIA is $\boldsymbol{O}(2^{69})$ and the complexity of IIA with preprocessing phase is $\boldsymbol{O}(2^{66})$.

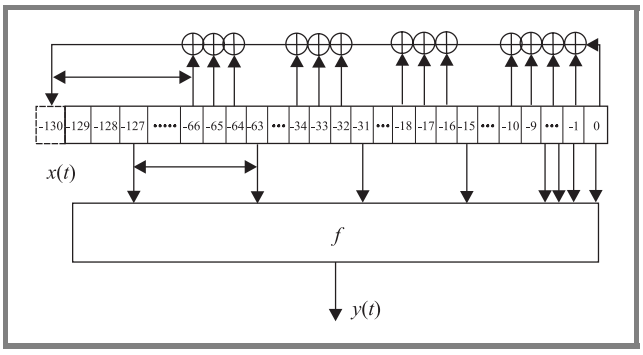

Fig. 4. Improved inversion attack with preprocessing phase on NFG.

In the preprocessing phase we find that the largest gap between taps to connection polynomial is between cells 130 and 66 and the length of that gap is equal to 64, similar as the gap between taps to filter function. The attack works as follows.
First we need to guess 66 bits $\big(x(t)\big)_{t=-63,\ldots,3}$, then we calculate $x(-127)$ from the filter function and known keystream segment:

$$x(-127) = y(66) \oplus g(x(0), x(-1), x(-3), x(-7), \\ x(-15), x(-31), x(-63)),$$

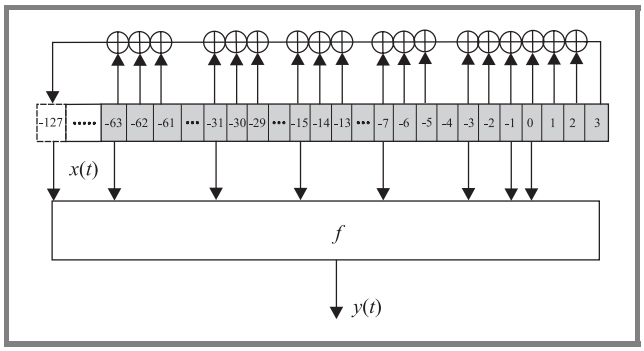

Fig. 5. Improved inversion attack with preprocessing phase on NFG (known cells are marked with grey colour).

then we clock backward the register state and, after clocking we know $\big(x(t)\big)_{t=--64\ldots2}$. We calculate $x(-127)$ from the filter function and known keystream segment:

$$x(-127) = y(65) \oplus g(x(0), x(-1), x(-3), x(-7),$$
$$x(-15), x(-31), x(-63)),$$

then we calculate $x(3)$ from the connection polynomial and so on.

5. Nonlinear filter generators design criteria

After introducing inversion attacks Golić [3] proposed a set of design criteria for NFG which were considered to ensure large period, high linear complexity and resistance to statistical attacks, inversion attacks, conditional correlation attacks and fast correlation attacks. To ensure properties from the first group (large period, high linear complexity) primitivity of connection polynomial and large algebraic order of function f are important. Good statistical properties can be ensured by the choice of filter function of the form (2) or (3).

Golić pointed out the fact that the computational complexity of inversion attack is exponential with the memory size M, rather than with the length of the register r. So, to make a cipher resistant to inversion attack he proposed to choose M as large as possible, preferably close to its maximum possible value $r-1$. Additionally, to avoid the possibility of effective reduction of memory size (by decimation technique), the tapping sequence should not be equidistant, preferably the greatest common divisor of elements of γ should be equal to one (assuming $\gamma_1 = 0$).

Resistance to conditional correlation attacks requires the number of nondegenerate inputs to f to be large enough, and the γ sequence chosen according to a full or λ-order positive difference set (with λ as small as possible for given n and r) and correlation immunity of f to be relatively large compared to λ.

To prevent fast correlation attack designers should ensure that the nonzero correlation coefficients of f to the set of linear functions are relatively small and close in magnitude. Finally, the number of nonzero terms in the feedback polynomial and in any of its low degree multiples should not be small.

The polynomial, tapping sequence and filter function used in Example 1 meet the above criteria. So, as we can see this set of design criteria does not prevent improved inversion attacks. So we propose to add the following criterion to the set:

Designers of stream ciphers should additionally minimise the largest gap between cells with taps to multiples of the connection polynomial or to the filter function.

6. Experiments

We have implemented the basic inversion attack and the improved inversion attack and we have conducted the following experiments on a typical Pentium II 400 MHz PC with 128 MB RAM:

1. Attacks on NFG with connection polynomial $p(x) = x^{33} \oplus x^{13} \oplus 1$, tapping sequence $\gamma = \{31, 15, 7, 3, 1, 0\}$ and filter function $f(x_{31}x_{15}x_7x_3x_1x_0) = x_{31} \oplus x_{15}x_3 \oplus x_7x_1 \oplus x_3x_1x_0$ for different initial states. Inversion attack on this generator takes up to few days and improved inversion attack takes up to 20 seconds (depending on initial state of the LFSR).

2. Attacks on NFG with connection polynomial $p(x) = x^{64} \oplus x^4 \oplus x^3 \oplus x^1 \oplus 1$, tapping sequence $\gamma = \{63, 31, 15, 7, 3, 1, 0\}$ and filter function $f(x_{63}x_{31}x_{15}x_7x_3x_1x_0) = x_{63} \oplus x_{31} \oplus x_{15}x_3 \oplus x_7x_1 \oplus x_3x_1x_0$. Inversion attack has computational complexity $\boldsymbol{O}(2^{63})$ so it is infeasible to conduct it on our PC. Improved inversion attack takes up to few days.

7. Conclusions and final remarks

We have proposed a powerful improvement of the inversion attacks. We have conducted several experiments which have confirmed theoretical predictions.

This attack is also effective when instead of regular LFSR, a modular LFSR is used (with inter cell feedback).

Our further research will concentrate on possible transformations of filter functions in such a way as to maximise the largest gap.

Acknowledgement

This work has been supported by grant no. 8 T11D 020 19 of the Polish Scientific Research Committee.

References

[1] R. J. Anderson, "Searching for the optimum correlation attack", in *Fast Software Encryption* – Leuven'94, *LNCS*. Springer, 1995, vol. 1008, pp. 137–143.

[2] J. Dj. Golić, "Correlation via linear sequential circuit approximation of combiners with memory", in *Advances in Cryptology – EUROCRYPT'92, LNCS*. Springer, 1993, vol. 658, pp. 113–123.

[3] J. Dj. Golić, "On the security of the nonlinear filter generators", in *Fast Software Encryption* – Cambridge'96, *LNCS*. Springer, 1996, vol. 1039, pp. 173–188.

[4] J. Dj. Golić, A. Clark, and E. Dawson, "Inversion attack and branching", in *Information Security and Privacy, ACISP'99, LNCS*. Springer, 1999, vol. 1587, pp. 88–102.

[5] J. Dj. Golić, Private communications, May 2002.

[6] A. Górska and K. Górski, "Improved inversion attacks on nonlinear filter generators", *IEE Electron. Lett.*, vol. 38, no. 16, pp. 870–871, 2002.

[7] W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers", *J. Cryptol.*, vol. 1, no. 3, pp. 159–176, 1989.

[8] M. Salmasizadeh, L. Simpson, J. Dj. Golić, and E. Dawson, "Fast correlation attacks and multiple linear approximations", in *Information Security and Privacy, ACISP'97, LNCS*. Springer, 1997, vol. 1270, pp. 228–239.

Anna Górska received the M.Sc. degree in 1995 and the Ph.D. degree in 2003 both from the Faculty of Electronics and Information Technology at Warsaw University of Technology, Poland. She is currently a cryptographer in the Cryptography Division at Enigma Information Security Systems Sp. z o.o. Her research interests include the design and cryptanalysis of ciphers. She is a member of the International Association for Cryptologic Research and the Institute of Electrical and Electronics Engineers.
e-mail: ania@enigma.com.pl
ENIGMA Information Security Systems Sp. z o.o.
Cryptography Division
Cietrzewia st 8
02-492 Warsaw, Poland

Karol Górski received the M.Sc. degree in 1991 from the Institute of Telecommunications, Faculty of Electronics and Information Technology at Warsaw University of Technology, Poland. He currently heads the Cryptography Division at Enigma Information Security Systems Sp. z o.o. His duties include the management of research and development activities undertaken by the company in the area of cryptography and cryptanalysis as well as the management of software development for cryptographic devices and specialised cryptographic systems. He is a member of the Technical Committee for Information Security in IT Systems at the Polish Standardisation Committee (PKN) and an expert of the Cryptographic Algorithms and Mechanisms Working Group (WG2) in the joint ISO/IEC subcommittee on Information Technology Security Techniques (ISO/IEC JTC1 SC27). He is a also a member of the International Association for Cryptologic Research and the Institute of Electrical and Electronics Engineers.
e-mail: karol@enigma.com.pl
ENIGMA Information Security Systems Sp. z o.o.
Cryptography Division
Cietrzewia st 8
02-492 Warsaw, Poland