Pomaranch is Sound and Healthy

Cees J.A. Jansen[1] and Alexander Kholosha[2]

[1] Banksys NV
Haachtsesteenweg 1442
1130 Brussels, Belgium
[2] The Selmer Center
Department of Informatics, University of Bergen
P.O. Box 7800, N-5020 Bergen, Norway
cja@iae.nl; Alexander.Kholosha@ii.uib.no

Abstract. Recently two attacks on the Pomaranch stream cipher have been described pointing to two weaknesses in the original design, namely the IV initialization procedure, and the characteristic polynomial used in the jump registers. The latter weakness has already been repaired as described in a previous note by the authors [1]. In this note we provide a simple tweak which effectively counters the attack on the initialization procedure. By this update Pomaranch is sound and healthy again.

1 Introduction

The Pomaranch stream cipher algorithm [2] uses a cascade of jump controlled registers, which are linear finite state machines that can very efficiently be switched between two transition matrices [3]. Pomaranch uses nine 14-bit long jump registers, accounting for a state of 126 bits. The secret key used with Pomaranch has a length of 128 bits. In the submitted Pomaranch description the IV used for initialization of the stream cipher has a maximum length of 112 bits.

In [4] a chosen IV attack on the initialization procedure of Pomaranch is described of complexity $O(2^{65})$ as claimed by the authors. In the same paper it is also shown that the non-zero state forcing method used in the initialization leads to a low complexity distinguisher reducing the complexity of the attack to $O(2^{52})$. In the next sections of this note we describe some minor changes to the initialization procedure that effectively counter the chosen IV attack.

A key-recovery attack of complexity $O(2^{95.4})$ requiring $2^{71.8}$ bits of key stream was recently described in [5]. This attack was made possible due to a property of the characteristic polynomial of the jump register which we have called the Linear Equivalence Bias. Although other ways exist to remove this bias, this attack has been effectively countered by choosing a different characteristic polynomial for the jump register as described in [1].

2 Countering the Chosen IV Attack on Pomaranch

As noted in [4] the basic flaw in the initialization procedure of Pomaranch is that there is no diffusion of the IV bits into the entire 126-bit state. Moreover,

the non-zero state forcing method used allows an attacker to easily find two IV's generating the same key stream output. In this section we show that by limiting the IV-length and a modified loading of the IV-bits into the registers, the attack described can be countered.

The attack works because IV_8, which is entirely loaded into R_8, can be chosen independently from the contents of the other registers. Now suppose $IV_8 = IV_1$, then the attack could be executed on R_7 choosing different IV_7 values. The attack would then require additional guesses for k_7, R_7^K, n_8', n_8'' and n_7. Doing so, this would increase the complexity of the attack by at least $16 + 14 + 3 \cdot 5 = 45$ bits and would bring the complexity of the attack to $O(2^{110})$. This result indicates that the IV-length should be limited such that only the first six registers are loaded with IV-bits and the remaining registers are loaded with $IV_1, \ldots, IV_3$.

In order to make sure that different IV values produce different output keystreams a different method to escape from all-zero register states is required. To this end, the IV-bits are split into groups of 13 bits denoted by IV_i, $1 \leq i \leq 6$. These 13 bit IV-values are exored with the 13 most significant bits of the registers R_i, $1 \leq i \leq 6$. Registers R_7, R_8 and R_9 are filled in the same way with IV_1, IV_2 and IV_3. Now all registers are checked for the all-zero state and if all-zero the least significant bit of the register is set to 1. In this way different IV's will always give rise to different output streams and no distinguisher exists anymore. It is clear that the IV-length is further limited to 78 bits. Shorter IV's can be accommodated by cyclically repeating IV-bits until all nine registers have been filled.

3 An IV-bit Diffusing Initialization

Although the modified initialization procedure of the previous section effectively counters the chosen IV attack, we present an alternative initialization procedure that does diffuse all IV-bits into the entire state.

First note that Shift Mode as defined in [2] is a nonlinear, key dependent permutation of 126-bit states. It has been designed to provide quick diffusion and make all state bits depend on all other state bits and all key bits. Next observe that it might happen that after execution of the Shift Mode a jump register contains all zeroes. Assuming the key bits are loaded, we first initialize the registers as in the original proposal.

1. Load the 9 registers with 126 bits representing the binary expansion of the fraction of π.
2. Run the algorithm in the Shift Mode for 128 steps.
3. Save the contents of all registers for subsequent initializations.

This will make all register contents dependent on all key bits. Next we load the IV-bits as follows.

1. Compose a 126-bit vector from the IV-bits by cyclically repeating IV-bits if the IV-length is less than 126 bits.

2. EXOR the 126-bit vector with the saved register contents of step 3 above and load it into the registers.
3. Run the algorithm in the Shift Mode for 128 steps.
4. Check the 9 registers for all-zero content. If a register is all-zero then change its least significant bit in a 1.
5. Perform a run-up of 64 bits in the Key Stream Generation Mode, generating key stream bits that are discarded.

This IV loading procedure will ensure that all 126 state bits depend on all IV bits and all key bits, thereby frustrating chosen IV attacks. This is the preferred IV initialization mode that should have been used in the submitted proposal.

4 Conclusion

The Pomaranch stream cipher is healthy and sound after changing to a different characteristic polynomial and tweaking the IV loading procedure by using the already defined Shift Mode.

References

1. Jansen, C.J.A., Kholosha, A.: Countering the correlation attack on Pomaranch. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/070 (2005) http://www.ecrypt.eu.org/stream/papersdir/070.pdf.
2. Jansen, C.J.A., Helleseth, T., Kholosha, A.: Cascade jump controlled sequence generator (CJCSG). In: Symmetric Key Encryption Workshop, Workshop Record, ECRYPT Network of Excellence in Cryptology (2005) http://www.ecrypt.eu.org/stream/ciphers/pomaranch/pomaranch.pdf.
3. Jansen, C.J.A.: Stream cipher design based on jumping finite state machines. Cryptology ePrint Archive, Report 2005/267 (2005) http://eprint.iacr.org/2005/267/.
4. Cid, C., Gilbert, H., Johansson, T.: Cryptanalysis of Pomaranch. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/060 (2005) http://www.ecrypt.eu.org/stream/papersdir/060.pdf.
5. Khazaei, S.: Cryptanalysis of pomaranch (CJCSG). eSTREAM, ECRYPT Stream Cipher Project, Report 2005/065 (2005) http://www.ecrypt.eu.org/stream/papersdir/065.pdf.