BSEA-1 - A Stream Cipher Backdooring Technique*

Eric Filiol

Operational Cryptology and Virology Lab, ESIEA, 38 rue des Drs Calmette et Guérin, 53000 Laval, France, {filiol}@esiea.fr

Keywords: Cryptography, Encryption Algorithms, Backdoor, Trapdoor, Cryptanalysis, Stream Cipher

Abstract. Recent years have shown that more than ever governments and intelligence agencies try to control and bypass the cryptographic means used for the protection of data. Backdooring encryption algorithms is considered as the best way to enforce cryptographic control. Implementation backdoors (at the protocol/implementation/management level) are generally considered. In this paper we propose to address the most critical issue of backdoors: mathematical backdoors or by-design backdoors, which are put directly at the mathematical design of the encryption algorithm. Considering a particular family (among all the possible ones) of backdoors, we present BSEA-1, a stream cipher algorithm which contains a design backdoor enabling an effective cryptanalysis. The BSEA-1 algorithm uses a 120-bit key. The exploitation of the backdoor enables to break the cipher with around 2 Kbits of knowplaintext in a few seconds.

1 Introduction

Despite the fact that in the late 90s/early 2000s, citizens have partially obtained the freedom for using cryptography, the recent years have shown that more than ever, governments and intelligence agencies still try to control and bypass the cryptographic means used for the protection of data and of private life. Snowden's leaks were a first upheaval. A tremendous number of secret projects (from NSA, GCHQ) have been revealed to the public opinion which proves this situation.

While the need for the security of everyday life activities (for companies, for citizens) requires more and more cryptography and highly secure communications means, recent bothering initiatives or decision by political decision-makers ask for an even stronger control over cryptography not to say preparing the simple prohibition or ban of cryptographic application such as telegram. The most recent decision towards the control over cryptography is that of the Australian government [11] which makes mandatory the use of backdoor in encryption systems. At the same time, the EU as well as a number of security agencies (such as French ANSSI, German BSI) confirmed that it is nonsense and militate for the mandatory use of end-to-end encryption.

In this paper we address the most critical issue of backdoors: mathematical or by-design backdoors. In other words, the backdoor is put directly at the mathematical design of the encryption algorithm. The RSA's Dual_EC_DRBG standard case falls within this category [3]. Other non-public examples are known within the military cryptanalysis community, and partially revealed to the public thanks to the 1995 Hans Bühler case [17]. There is quite no public work on that topic. It is the technical realm of a few among the most eminent intelligence agencies (namely NSA, GCHQ) which moreover have the ability and power to step in and to influence the

^{*} This work has been presented at the Ruscrypto 2019 conference in Moscow

international standardization processes in one direction or another. Recently Bannier & Filiol [1] have published a block cipher algorithm (BEA-1) which is similar to the AES and which contains a mathematical backdoor enabling an effective cryptanalysis. This block cipher algorithm (80-bit block, 120-bit key size, 11 rounds) was designed to resist to linear and differential cryptanalyses.

This paper is organized as follows. In Section 2 we discuss the comparative feasibility of backdoors in stream ciphers and block ciphers. We also present the state-of-the-art, history and previous work regarding backdoors. In Section 3, we describe our backdoored stream cipher algorithm BSEA-1 (standing for *Backdoored Stream Encryption Algorithm 1*) and address the cryptographic security of this cipher, with respect to known cryptanalyses. In Section 4 we explain how to exploit the backdoor when considering known plaintext and ciphertext only cryptanalyses. Finally in Section 5 we summarizes our work and present future work.

2 The Concept of Backdoor in Symmetric Cryptology

The general concept of backdoor has been addressed in [1]. In this section, we just deal with this concept when considering stream ciphers in comparison with block ciphers.

2.1 Stream Ciphers vs Block Ciphers

Their respective complexity is totally different, especially with respect to their combinatorial complexity. We can define the combinatorial complexity as the number of internal configurations that can be realized during the different steps of operation (key setup, encryption, decryption).

- Stream ciphers. Their design complexity is rather low since they mostly rely on algebraic primitives (LFSRs and Boolean functions which have intensely been studied in the open literature). The cryptographic properties of these primitives are well-known may it be for the LFSRs or the Boolean functions used since the latter have generally a limited dimension (the number of input bits, 4 or 5 at most). Until the late 70s, backdoors relied on the fact that quite all algorithms were proprietary and hence secret. It was then easy to hide non primitive polynomials, weak combining Boolean functions or more exotic designs. The Hans Bühler case in 1995 [17] shed light on that particular case. Being secret, anyone who use the algorithm may only perform a deep statistical analysis. In this respect, the pseudo-running key (which is combined to the plaintext or the ciphertext) must always exhibit excellent randomness properties.
- Block ciphers. This class of encryption algorithms is rather recent (end of the 70s for the public part). They exhibit so a huge combinatorial complexity that it is reasonable to think to backdoors. As described in [5] for a k-bit secret key and a m-bit input/output block cipher there are $((2^m)!)^{2^k}$ possible such block ciphers. For such an algorithm, the number of possible internal states (which involves both the key and the input block whereas stream ciphers just input a secret key) is so huge that we are condemned to have only a local view of the system, that is, the round function or the basic cryptographic primitives. We cannot be sure that there is no degeneration effect at a higher level. This point has been addressed in [5] when considering linear cryptanalysis. Therefore, it seems reasonable to think that this combinatorial richness of block cipher may be used to hide backdoors.

2.2 Previous Work

While a few research work does exist regarding backdoors in block ciphers (see [1, Section 2]), there is not public research work on stream ciphers, to the author's knowledge. It is somehow

surprising when considering that from the mid-80s to the early 2000s, this class of encryption systems has been widely studied. At the industry level (that of encryption machines sold to governments), stream ciphers were also the vast majority of systems used throughout the world. Since, there are still used, at least partly, in payTV systems, telecommunications and satellite communication (where fast encryption is more than ever required), access control systems, subway tickets, and various other security-related applications [14, 13, 12]... With the rise of IoT, stream ciphers seem also to know some sort of come back.

As far as the intelligence world is concerned, NSA and GCHQ — among possibly a few others — have conducted an intense research activity with regards to backdoors for this class of ciphers. The Bühler case in 1995 [17] revealed that Crypto AG, a Swiss company and the main provider of cipher machines for nearly 120 governments and international entities, was working closely with the NSA to introduce backdoors in the encryption systems sold. So did a handful of other European companies selling crypto-machines.

Another kind of backdoors in stream ciphers relates to implementations that enables the secret key to be reused or to be the same with a high probability. For instance, the message key (used along with a base key) may have a very short entropy. In this case the cryptanalysis becomes rather easy [7].

3 Description of BSEA-1

The BSEA-1 algorithm (standing for *Backdoored Stream Encryption Algorithm version 1*) is based on our research work and the analysis of the rare available technical details and exchanges with experts in the field.

This section is intended to describe this cipher precisely. It is a classical combination generator [15, Section 5.2] which uses a 120-bit secret key (Figure 1). The essential difference with this design lies in the fact that the truth table of the combining Boolean function is modified at each time instant t by one of the registers which is moreover irregularly clocked.

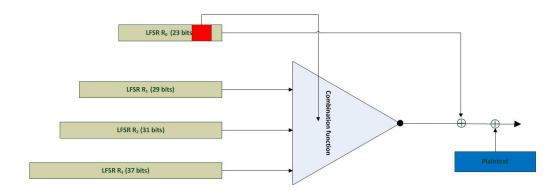


Fig. 1. General structure of BSEA-1 Algorithm

The cryptographic primitives are the following:

- Four Linear Feedback Shift Registers R_0, R_1, R_2 and R_3 of respective length $L_0 = 23, L_1 = 29, L_2 = 31$ and $l_3 = 37$. Their initialization is the secret key at time instant t = 0.

- The four feedback polynomials are primitive and given by

$$\begin{split} P_0(x) = & x^{23} \oplus x^{22} \oplus x^{20} \oplus x^{18} \oplus x^{17} \oplus x^{13} \oplus x^{11} \oplus x^{10} \oplus x^9 \oplus x^8 \oplus x^4 \oplus x^3 + \oplus x^2 \oplus x \oplus 1 \\ P_1(x) = & x^{29} \oplus x^{28} \oplus x^{27} \oplus x^{25} \oplus x^{24} \oplus x^{23} \oplus x^{22} \oplus x^{21} \oplus x^{18} \oplus x^{17} \oplus x^{13} \oplus x^{11} \oplus x^{10} \oplus x^6 \\ & \oplus x^5 \oplus x^3 \oplus x^2 \oplus x \oplus 1 \\ P_2(x) = & x^{31} \oplus x^{30} \oplus x^{27} \oplus x^{25} \oplus x^{24} \oplus x^{23} \oplus x^{22} \oplus x^{21} \oplus x^{20} \oplus x^{16} \oplus x^{15} \oplus x^{13} \oplus x^{12} \oplus x^{11} \\ & \oplus x^{10} \oplus x^9 \oplus x^8 \oplus x^4 \oplus x^3 \oplus x \oplus 1 \\ P_3(x) = & x^{37} \oplus x^{34} \oplus x^{33} \oplus x^{32} \oplus x^{30} \oplus x^{29} \oplus x^{26} \oplus x^{24} \oplus x^{20} \oplus x^{19} \oplus x^{18} \oplus x^{17} \oplus x^{16} \oplus x^{13} \\ & \oplus x^{11} \oplus x^8 \oplus x^7 \oplus x^6 \oplus x^4 \oplus x^2 \oplus 1 \end{split}$$

- The initial value of the Boolean function at time instant t is given by

$$0x6B = (0, 1, 1, 0, 1, 0, 1, 1)$$

A pseudo-random sequence $(\sigma_t)_{1 \le t \le N}$ is produced and xored to the plaintext (encryption) or to the ciphertext (decryption). The encryption algorithm is given hereafter:

Algorithm 1 Pseudo-random sequence generation (base version, encryption)

Require: Secret 120-bit key K and $P = (p_1, \ldots, p_N)$ a plaintext of length N

```
1: Key setup (R_0, R_1, R_2, R_3) \leftarrow K
 2: Combining function f \leftarrow 0x6B \{(1,1,0,1,0,1,1,0)\}
 3: for t from 1 to N do
       Compute S = (R_0 \& 3) + 1 {Step value in [1, 4]}
       for i from 1 to S do
 5:
           Clock register R_0 once
 6:
 7:
        end for
       X_0^t \leftarrow R_0 \& 1 \{R_0 \text{ output } x_0^t\}
       \tau \leftarrow (R_0 >> 3) \& 0x7
        f \leftarrow f \oplus (R_0 >> \tau) \& 0xFF \{ \text{modification pattern for } f \}
        Clock registers R_1, R_2, R_3 once and output x_1^t, x_2^t, x_3^t
        \sigma_t = f(x_1^t + (x_2^t << 1) + (x_3^t << 2)) \oplus x_0^t
13: end for
14: return (c_t = \sigma_t \oplus p_t)_{1 \leq t \leq N}
```

The base algorithm presented here is the base version. A large number of variants can be derived from the base version:

- Feedback polynomials and Boolean function initial value can be changed.
- Registers R_1R_2 , R_3 can be irregularly clocked by register R_0 according to different clocking settings.
- Value S (step value, line 4 in Algorithm 1) and τ (line 9 in Algorithm 1) can be computed according a lot of different ways.

3.1 BSEA-1 Security Analysis

In stream cipher theory, a number of cryptographic properties for the core primitives must be achieved:

- All feedback polynomials are primitive [15].
- Feedback polynomial degrees are co-prime $(L_0 = 23, L_1 = 29, L_2 = 31, L_3 = 37)$ [15].
- Each feedback polynomial has a prime degree (in order prevent the decimation attack [6]).
- Combination Boolean function (initial value) has relatively good cryptographic properties.

The variability over the time provides a **false sense** of cryptographic security. Indeed, since the truth table is constantly changing, it **seems** impossible to build the data required for known attacks:

- Noisy equations for correlation attacks [16] and fast correlation attacks [9] and similar variants.
- Non-linear equations to be solved in algebraic attacks and similar variants [2].

The statistical analysis of the pseudo-random sequence expanded from the secret key is also a very important cryptographic property. Since the design may be most of the time secret and embedded in a device (crypto-machine for instance or a IoT device), it is however possible to check the randomness properties in a black-box approach. In this respect BSEA-1 has been tested with different suites: FIPS 140-2/STS (US NIST standard), TestUI01 [8] and DieHarder [4]. The final conclusion is that BSEA-1 is statistically compliant with FIPS 140-2. It means that BSEA-1 would then pass all classical cryptographic validation which are generally considered by the industry.

4 BSEA-1 Cryptanalysis with the Backdoor Knowledge

4.1 Description of the Backdoor

The value of the Boolean function varies over the time. So does its Walsh spectrum.

 The Walsh transform summarizes the correlation between the Boolean function inputs and its output value [10]

$$\widehat{\chi_f}(u) = \sum_{x \in \mathbb{F}_n^n} -1^{f(x) \oplus \langle x, u \rangle} \text{ and } P[f(x) = \langle x, u \rangle] = \frac{1}{2} (1 + \frac{\widehat{\chi_f}(u)}{2^n})$$

– The Walsh spectrum S gives the correlation for all the possible linear combination of the function inputs $u = (u_1, u_2, \dots, u_n)$:

$$S = (\widehat{\chi_f}(00\cdots 00), \widehat{\chi_f}(00\cdots 01), \dots, \widehat{\chi_f}(11\cdots 11))$$

Let us now see how to apply this on a Boolean function which is changing at each time instant

- Whenever the Boolean function takes particular values, the Walsh spectrum takes strong correlation values (backdoor values) For instance when f = 0x69 then S = (0,0,0,0,0,0,0,0,-8)
- For these particular values, it means that the linear combination of the inputs and the output are equal with probability p = 1.0. It is then possible to write a linear equation whose unknowns are the R_1 , R_2 and R_3 key bits.
- Exactly 16 values over 256 possibles have a similar Walsh spectrum.
 - $\mathcal{B} = \{0x69, 0x5A, 0x55, 0x3C, 0x33, 0xF, 0xF0, 0xCC, 0xC3, 0xAA, 0xA5, 0x99, 0x99, 0x96, 0x66, 0x00, 0xFF\}$
- Values 0x00 and 0xFF enables to speed up the cryptanalysis by keeping or discarding key candidates quickly.

It is worth mentioning that the time indices for the backdoor values are not the same from key to key. They are strongly dependent from the secret key.

4.2 Known Plaintext Attack

In this case we know N bits of ciphertext $((c_t)_{1 \le t \le N})$ and of corresponding plaintext $((p_t)_{1 \le t \le N})$. Hence we can obtain the pseudo-random sequence in a straightforward way: $\sigma_t = c_t \oplus p_t$ for each time instant t. The cryptanalysis considers an exhaustive search with respect to register R_0 . Here are the main steps:

- Exhaustive search on register R_0 .
- For each initial value I_0 of R_0
 - Boolean function values 0x00 and 0xFF enables to discard I_0 .
 - We build a system of 97 equations of 97 unknowns and solve it.
 - We test the final K (23 + 97 bits) against the known plaintext.

The pseudo-code of the attack is given in Algorithm 2.

Algorithm 2 BSEA-1 Cryptanalysis Algorithm (Known Plaintext Attack)

```
Require: Pseudo-random sequence (\sigma_t)_{0 \le t \le N}
 1: for I_0 from 0 to 2^{L_0} - 1 do
 2:
        R_0 \leftarrow I_0
 3:
        for t from 1 to N do
           Compute Boolean function value f_{I_0^t}
 4:
 5:
           if (f_{I_0^t} == 0x00 \text{ and } \sigma_t \neq 0) or (f_{I_0^t} == 0xFF \text{ and } \sigma_t \neq 1) then
 6:
              Discard I_0 and continue
 7:
 8:
           if f_{I_{\underline{t}}} \in \mathcal{B} \setminus \{0x00, 0xFF\} then
 9:
              Write Equation and add it to the system S_{I_0}
10:
           end if
11:
        end for
        Solve equation system S_{I_0}
12:
13:
        Test the solution K_{I_0} against (\sigma_t)_{0 \le t \le N}
14:
        if K_{I_0} is correct then
15:
           K = K_{I_0} and break
16:
17: end for
18: return K
```

We need only N=1,800 bits of known plaintext to break the whole key K with a complexity of $\mathcal{O}(2^{L_0})$ where L_0 is the length of register R_0 . In most real-life cases, side (implementation) backdoors enable to have a few kbits of known plaintext very easily. For instance, in strategic telegraph systems (export version), synchronizing information is encrypted. In synchronous systems (high data rate) an initial, agreed patterns of bits (generally defined in the technical specifications) is sent but encrypted. In asynchronous system (low data rate) synchronizing bits are inserted (called stop-bits and start-bits) and encrypted.

When evaluating a cryptographic system not only the algorithm but also its implementation and use must be carefully analysed in order to evaluate the risk of side implementation backdoors. Cryptographic algorithms must be armoured door installed on a cardboard wall.

4.3 Ciphertext-only Attack

The principle remains the same but requires a little bit more effort. Whenever the Boolean function takes particular values (different from those in \mathcal{B}), the Walsh spectrum takes strong correlation values

- 256 possibles have a similar Walsh spectrum).
- Then we have Equation $1 + x_3^t = f(x_3^t, x_2^t, x_1^t)$ holding with probability p = 0.875. If we consider that $p[m_t = 0] = 0.6$ (probability of a plaintext bit to be equal to 0, which is a rather commonly observed for many languages and for most encodings) then Equation $1 + x_3^t = c_t$ holds with probability p = 0.575

We get then a system of noisy equations to solve for each register [16]. We need about 50 kb of ciphertext bits to recover the whole key with a complexity of at most $\mathcal{O}(2^{54})$. However many optimizations are possible to reduce this complexity significantly.

Conclusion and Future Work

In this paper, we have proposed one possible technique of stream cipher backdooring at the design level. It is illustrated by a 120-bit algorithm, named BSEA-1 which exhibits many of the desirable properties that any secure stream cipher algorithm should. When exploiting the backdoor, we manage to break it with a very limited amount of resources successfully.

The next BSEA variant consists in slightly changing the way the Boolean function is updated over the time. Instead of modifying the whole function, it is better to modify it "by half". The modification pattern π of size 2^{n-1} is then applied as follows:

$$f \leftarrow f \oplus ((\pi << 2^{(n-1)})|\pi)$$

The cryptanalysis method becomes less obvious than for BSEA-1 and requires to consider far different cryptanalysis approaches and methods.

It is worth stressing on the fact that backdooring stream ciphers requires to consider working at the combination module mostly.

- Secure primitives of the random engine part (LFSRs) are very much well known (primitive, dense polynomials of prime and coprime length...)
- However due to lack of combinatorial complexity, backdoored designs are bound to remain secret mostly.

The next step in this research work about cryptographic backdooring techniques (design and detection) will be to consider more sophisticated designs and primitives such as Nonlinear Feedback Shift Registers (NLFSR), design with memory...

References

- Arnaud Bannier and Eric Filiol. "Partition-Based Trapdoor Ciphers". In: Partition-Based Trapdoor Ciphers. Ed. by Arnaud Bannier and Eric Filiol. Rijeka: IntechOpen, 2017. Chap. 1. DOI: 10.5772/intechopen.70420. URL: https://doi.org/10.5772/intechopen. 70420.
- Gregory V. Bard. Algebraic Cryptanalysis. 1st. Springer Publishing Company, Incorporated, 2009. ISBN: 0387887563, 9780387887562.
- Daniel J. Bernstein, Tanja Lange, and Ruben Niederhagen. "Dual EC: A Standardized Back Door". In: The New Codebreakers: Essays Dedicated to David Kahn on the Occasion of His 85th Birthday. Ed. by Peter Y. A. Ryan, David Naccache, and Jean-Jacques Quisquater. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 256-281. ISBN: 978-3-662-49301-4. DOI: 10.1007/978-3-662-49301-4_17. URL: https://doi.org/10.1007/978-3-662-49301-4_17.

- [4] Robert G. Brown, Dirk Eddelbuettel, and David Bauer. *Dieharder: A Random Number Test Suite Version 3.31.1s.* 2017. URL: https://webhome.phy.duke.edu/~rgb/General/dieharder.php.
- [5] Joan Daemen and Vincent Rijmen. The design of Rijndael. Springer Verlag, 2002.
- [6] Eric Filiol. "Decimation Attack of Stream Ciphers". In: *Progress in Cryptology INDOCRYPT* 2000. Ed. by Bimal Roy and Eiji Okamoto. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 31–42. ISBN: 978-3-540-44495-4.
- [7] Eric Filiol. How to Detect and Break Operationally the Misuse of Weak Stream Ciphers (and Even Block Ciphers Sometimes) Application to the Office Encryption Cryptanalysis. Black Hat Europe 2010. 2010. URL: https://www.blackhat.com/html/bh-eu-10/bh-eu-10-archives.html#Filiol.
- [8] Pierre L'Ecuyer and Richard Simard. "TestU01: A C Library for Empirical Testing of Random Number Generators". In: ACM Trans. Math. Softw. 33.4 (Aug. 2007), 22:1-22:40.
 ISSN: 0098-3500. DOI: 10.1145/1268776.1268777. URL: http://doi.acm.org/10.1145/ 1268776.1268777.
- Willi Meier and Othmar Staffelbach. "Fast correlation attacks on certain stream ciphers".
 In: Journal of Cryptology 1.3 (Oct. 1989), pp. 159-176. ISSN: 1432-1378. DOI: 10.1007/BF02252874. URL: https://doi.org/10.1007/BF02252874.
- [10] Willi Meier and Othmar Staffelbach. "Nonlinearity Criteria for Cryptographic Functions". In: Advances in Cryptology — EUROCRYPT '89. Ed. by Jean-Jacques Quisquater and Joos Vandewalle. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 549–562. ISBN: 978-3-540-46885-1.
- [11] BBC News. Australia data encryption laws explained. https://www.bbc.com/news/world-australia-46463029. 7 December 2018.
- [12] Karsten Nohl. Cryptanalysis of Crypto-1. 2008. URL: https://www.cs.virginia.edu/~kn5f/Mifare.Cryptanalysis.htm.
- [13] Karsten Nohl. Lost Mifare obscurity raises concerns over security of OV-Chipkaart. 2008. URL: https://www.cs.virginia.edu/~kn5f/.
- [14] Karsten Nohl and Henryk Ploetz. *Mifare Little security despite Obscurity*. Talk at 24C3. 2007. URL: https://events.ccc.de/congress/2007/Fahrplan/events/2378.en.html.
- [15] Rainer A. Rueppel. *Analysis and Design of Stream Ciphers*. Berlin, Heidelberg: Springer-Verlag, 1986. ISBN: 0-387-16870-2.
- [16] Thomas Siegenthaler. "Decrypting a Class of Stream Ciphers Using Ciphertext Only". In: *IEEE Trans. Computers* 34.1 (1985), pp. 81–85. DOI: 10.1109/TC.1985.1676518. URL: https://doi.org/10.1109/TC.1985.1676518.
- [17] Res Strehle. Verschlüsselt: der Fall Hans Bühler. Werd, 1994.