BSEA-1. “A Stream cipher Backdooring Technique”

Eric Filiol
filiol@esiea.fr

ESIEA
Operational Cryptology and Virology Lab $(C+V)^O$

Summary of the talk

Summary of the talk

Introduction

- Encryption systems have always been under export controls (ITAR, Wassenaar...) and still are. Considered as weapons and dual-use means.

Introduction

- Encryption systems have always been under export controls (ITAR, Wassenaar...) and still are. Considered as weapons and dual-use means.
- Implementation backdoors
  - Key escrowing, key management and key distribution protocols weaknesses (refer to recent NSA/CIA leaks since 2013)
  - Hackers are likely to find and use them as well

Introduction

- Encryption systems have always been under export controls (ITAR, Wassenaar...) and still are. Considered as weapons and dual-use means.
- Implementation backdoors
  - Key escrowing, key management and key distribution protocols weaknesses (refer to recent NSA/CIA leaks since 2013)
  - Hackers are likely to find and use them as well
- Mathematical backdoors
  - Historic cases: Crypto AG and Buehler's case (1995) mostly in stream ciphers
  - Extremely few open and public research in this area
  - Known existence of NSA and GCHQ research programs

Introduction

- Encryption systems have always been under export controls (ITAR, Wassenaar...) and still are. Considered as weapons and dual-use means.
- Implementation backdoors
  - Key escrowing, key management and key distribution protocols weaknesses (refer to recent NSA/CIA leaks since 2013)
  - Hackers are likely to find and use them as well
- Mathematical backdoors
  - Historic cases: Crypto AG and Buehler's case (1995) mostly in stream ciphers
  - Extremely few open and public research in this area
  - Known existence of NSA and GCHQ research programs
- Sovereignty issue: can we trust foreign encryption algorithms?

Aim of the Present Research

- Try to answer to the key question
  - *"How easy and feasible is it to design and to insert backdoors (at the mathematical level) in encryption algorithms?"*

Aim of the Present Research

- Try to answer to the key question
  - *"How easy and feasible is it to design and to insert backdoors (at the mathematical level) in encryption algorithms?"*
- Explore the different possible approaches
  - We consider a particular case of backdoors in stream ciphers.

Summary of the talk

State-of-the-Art

- No public study has been ever published on stream ciphers backdooring techniques.
    - Until the early 2000s, still a lot of encryption systems sold to many governments were stream ciphers
    - The mathematical design is generally not public (the client may have a partial technical description only)
    - But the client may perform statistical analysis. So the system must appear as "good for service" and trustworthy.

State-of-the-Art

- No public study has been ever published on stream ciphers backdooring techniques.
  - Until the early 2000s, still a lot of encryption systems sold to many governments were stream ciphers
  - The mathematical design is generally not public (the client may have a partial technical description only)
  - But the client may perform statistical analysis. So the system must appear as “good for service” and trustworthy.
- BSEA-1 design is derived from
  - Research at the laboratory
  - A few information derived from the rare available technical documentation on the topic
  - One possible class of backdoors (among many other possible classes yet more complex to describe)

Backdoored Stream Ciphers vs Backdoored Block Ciphers

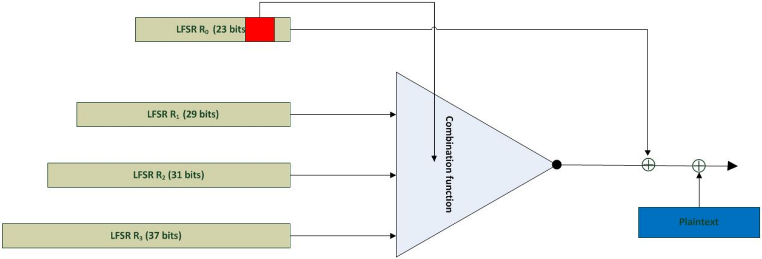
- Cryptographic primitives and properties for stream ciphers are simple and relatively easy to master
  - Large corpus of theoretical (academic level) and operational (industry and governmental levels) knowledge
  - Poor combinatorial complexity. It is therefore difficult to imagine and design non trivial backdoors. Is “less trivial” possible and how?
  - The use of trivial backdoors requires to make the algorithm non public.

Backdoored Stream Ciphers vs Backdoored Block Ciphers

- Cryptographic primitives and properties for stream ciphers are simple and relatively easy to master
  - Large corpus of theoretical (academic level) and operational (industry and governmental levels) knowledge
  - Poor combinatorial complexity. It is therefore difficult to imagine and design non trivial backdoors. Is "less trivial" possible and how?
  - The use of trivial backdoors requires to make the algorithm non public.
- Block ciphers are more complex and offers a better environment to hide backdoors
  - Huge combinatorial complexity
  - The corpus of theoretical knowledge is still limited (compared to that for stream ciphers)
  - Algorithms can be made public without necessarily revealing the backdoor (Bannier et al., 2017)

BSEA-1 Key Features

- Parameters
  - Variable Boolean function combining non-linearly 3 LFSRs (R_1, R_2, R_3).
  - 120-bit master key
  - One LFSR R_0 (irregularly clocked) produces output that modifies the Boolean function at each time instant t



Cryptographic Primitives

- Feedback polynomials

$$P_0(x) = x^{23} \oplus x^{22} \oplus x^{20} \oplus x^{18} \oplus x^{17} \oplus x^{13} \oplus x^{11} \oplus x^{10} \oplus x^9 \oplus x^8 \oplus x^4 \oplus x^3 + \oplus x^2 \oplus x \oplus 1$$

$$P_1(x) = x^{29} \oplus x^{28} \oplus x^{27} \oplus x^{25} \oplus x^{24} \oplus x^{23} \oplus x^{22} \oplus x^{21} \oplus x^{18} \oplus x^{17} \oplus x^{13} \oplus x^{11} \oplus x^{10} \oplus x^6 \oplus x^5 \oplus x^3 \oplus x^2 \oplus x \oplus 1$$

$$P_2(x) = x^{31} \oplus x^{30} \oplus x^{27} \oplus x^{25} \oplus x^{24} \oplus x^{23} \oplus x^{22} \oplus x^{21} \oplus x^{20} \oplus x^{16} \oplus x^{15} \oplus x^{13} \oplus x^{12} \oplus x^{11} \oplus x^{10} \oplus x^9 \oplus x^8 \oplus x^4 \oplus x^3 \oplus x \oplus 1$$

$$P_3(x) = x^{37} \oplus x^{34} \oplus x^{33} \oplus x^{32} \oplus x^{30} \oplus x^{29} \oplus x^{26} \oplus x^{24} \oplus x^{20} \oplus x^{19} \oplus x^{18} \oplus x^{17} \oplus x^{16} \oplus x^{13} \oplus x^{11} \oplus x^8 \oplus x^7 \oplus x^6 \oplus x^4 \oplus x^2 \oplus 1$$

Cryptographic Primitives

- Feedback polynomials

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

- Initial value of combining Boolean function

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

BSEA-1 Algorithm (encryption and decryption)

- Generation of a pseudo-random sequence $(\sigma_t)_{t \geq 0}$ that is xored to the plaintext (encryption) or to the ciphertext (decryption).

Algorithm 1 Pseudo-random sequence generation (base version)

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

Key setup $(R_0, R_1, R_2, R_3) \leftarrow K$
Combining function $f \leftarrow 0x6B$ $\{(1,1,0,1,0,1,1,0)\}$
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**
        Clock register R_0 once
    end for
    $X_0^t \leftarrow R_0 \ \& \ 1$ {R_0 output x_0^t}
    $\tau \leftarrow (R_0 >> 3) \ \& \ 0x7$
    $f \leftarrow f \oplus (R_0 >> \tau) \ \& \ 0xFF$ {modification pattern for f}
    Clock registers R_1, R_2, R_3 once and output x_1^t, x_2^t, x_3^t
    $\sigma_t = f(x_1^t + (x_2^t << 1) + (x_3^t << 2)) \oplus x_0^t$
end for
return $(c_t = \sigma_t \oplus p_t)_{1 \leq t \leq N}$

BSEA-1 Variants

- Feedback polynomials and Boolean function initial value can be changed.

BSEA-1 Variants

- Feedback polynomials and Boolean function initial value can be changed.
- Registers R_1, R_2, R_3 can be irregularly clocked by register R_0.

BSEA-1 Variants

- Feedback polynomials and Boolean function initial value can be changed.
- Registers R_1, R_2, R_3 can be irregularly clocked by register R_0.
- Values S (step value, line 4) and τ can vary
    - A lot of variants are then possible

BSEA-1 Security Analysis

- All feedback polynomials are primitive.

BSEA-1 Security Analysis

- All feedback polynomials are primitive.
- Feedback polynomial degrees are co-prime $(23, 29, 31, 37)$.

BSEA-1 Security Analysis

- All feedback polynomials are primitive.
- Feedback polynomial degrees are co-prime $(23, 29, 31, 37)$.
- Each feedback polynomial is prime (prevent the decimation attack).

BSEA-1 Security Analysis

- All feedback polynomials are primitive.
- Feedback polynomial degrees are co-prime $(23, 29, 31, 37)$.
- Each feedback polynomial is prime (prevent the decimation attack).
- Combination Boolean function has relatively good cryptographic properties. Its variability over the time **seems** to increase security and make existing attacks impossible to build:
  - Correlation attacks and fast correlation attacks
  - Algebraic attacks.

BSEA-1 Security Analysis

- All feedback polynomials are primitive.
- Feedback polynomial degrees are co-prime $(23, 29, 31, 37)$.
- Each feedback polynomial is prime (prevent the decimation attack).
- Combination Boolean function has relatively good cryptographic properties. Its variability over the time **seems** to increase security and make existing attacks impossible to build:
  - Correlation attacks and fast correlation attacks
  - Algebraic attacks.
- BSEA-1 is statistically compliant with FIPS 140-2/STS (US NIST standard). Aslo with respect to TestUI and DieHarder.

BSEA-1 Security Analysis

- All feedback polynomials are primitive.
- Feedback polynomial degrees are co-prime $(23, 29, 31, 37)$.
- Each feedback polynomial is prime (prevent the decimation attack).
- Combination Boolean function has relatively good cryptographic properties. Its variability over the time **seems** to increase security and make existing attacks impossible to build:
  - Correlation attacks and fast correlation attacks
  - Algebraic attacks.
- BSEA-1 is statistically compliant with FIPS 140-2/STS (US NIST standard). Aslo with respect to TestUI and DieHarder.
- BSEA-1 would then pass all classical cryptographic validation and analysis tests!

Summary of the talk

Boolean Function Analysis - Walsh Transform

- 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

$$\widehat{\chi_f}(u) = \sum_{x \in \mathbb{F}_2^n} -1^{f(x) \oplus <x,u>} \text{ and } P[f(x) =< x, u >] = \frac{1}{2}(1 + \frac{\widehat{\chi_f}(u)}{2^n})$$

  - The Walsh spectrum $\mathcal{S}$ gives the correlation for all the possible linear combination of the function inputs $u = (u_1, u_2, \ldots, u_n)$:

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

Analysis of the Variable Boolean Function

- Whenever the Boolean function takes particular values, the Walsh spectrum takes strong correlation values
  - For instance when $f = 0x69$ then $S = (0, 0, 0, 0, 0, 0, 0, -8)$

Analysis of the Variable Boolean Function

- Whenever the Boolean function takes particular values, the Walsh spectrum takes strong correlation values
  - For instance when $f = 0x69$ then $S = (0, 0, 0, 0, 0, 0, 0, -8)$
- For these particular values, it means that the linear combination of the inputs and the output are equal with probability $p = 1.0$. You then can 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.

Cryptanalysis Principle (Known Plaintext Attack)

- Exhaustive search on register R_0.

Cryptanalysis Principle (Known Plaintext Attack)

- 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.
- 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 case, side (implementation) backdoors enable to have a few kbits of known plaintext very easily (e.g. encryption of synchronization frames).

Cryptanalysis Principle (Known Plaintext Attack)

- 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.
- 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 case, side (implementation) backdoors enable to have a few kbits of known plaintext very easily (e.g. encryption of synchronization frames).
- As for the ciphertext only attack, the principle remains the same yet with a bit more tricks to consider. Homework for the audience:
  - Cryptanalysis requires about 50 kbits of ciphertext. The complexity is at most $\mathcal{O}(2^{54})$).

BSEA-1 Cryptanalysis Algorithm (Known Plaintext Attack)

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

Require: Pseudo-random sequence $(\sigma_t)_{0 \leq t \leq N}$

for I_0 from 0 to $2^{L_0} - 1$ **do**
    $R_0 \leftarrow I_0$
    for t from 1 to N **do**
        Compute Boolean function value $f_{I_0^t}$
        if $(f_{I_0^t} == 0x00$ and $\sigma_t \neq 0)$ or $(f_{I_0^t} == 0xFF$ and $\sigma_t \neq 1)$ **then**
            Discard I_0 and continue
        end if
        if $f_{I_0^t} \in \mathcal{B} \setminus \{0x00, 0xFF\}$ **then**
            Write Equation and add it to the system S_{I_0}
        end if
    end for
    Solve equation system S_{I_0}
    Test the solution K_{I_0} against $(\sigma_t)_{0 \leq t \leq N}$
    if K_{I_0} is correct **then**
        $K = K_{I_0}$ and break
    end if
end for
return K

Summary of the talk

Conclusion

- 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 design are bound to remain secret mostly.
  - New primitives are to consider (NLFSRs)

Conclusion

- 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 design are bound to remain secret mostly.
  - New primitives are to consider (NLFSRs)

- Future work
  - First step in a larger research work
  - The next step is to slightly modify the Boolean function modification. 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:

$$f \leftarrow f \oplus ((\pi << (n-1))|\pi)$$

  - The cryptanalysis method becomes less obvious than for BSEA. To be continued...

Conclusion

Thank you for your attention
Questions & Answers