Attacks on PRNGs

Gaspare FERRARO
CyberSecNatLab
Matteo ROSSI
Politecnico di Torino

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it

License & Disclaimer

License Information

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Goal

- Learn how to break LCG with different knowledges
- Learn how to break the Mersenne Twister
- Learn different ways to break a LFSR

Prerequisites

- Lectures:
  - *CR_0.1 – Number Theory and modular arithmetic*
  - *CR_0.2 – Random Number Generation*
  - *HW_0.2.3 – Linear Feedback Shift Registers*

Outline

- Introduction
- Attacks on LCGs
- Attacks on Mersenne Twister
- Attacks on LFSR
- Rand in practice

Outline

- Introduction
- Attacks on LCGs
- Attacks on Mersenne Twister
- Attacks on LFSR

(Pseudo-)Random Number Generators

- A Random Number Generator (RNG) is a utility or device of some type that produces a sequence of numbers within an interval [min, max] while guaranteeing that values appear unpredictable
- A Pseudo-Random Number Generators (PRNG) is an algorithm or a hardware device that generates a sequence of random bits or numbers

PRNGs in cryptography

- Not every random number generator needs to be *secure* in the cryptographic sense
- Most of them are designed to be good for simulations
- PRNGs are not considered *cryptographically secure*
- However, PRNG researchers have worked to solve this problem by creating what are known as *Cryptographically Secure PRNGs* (CSPRNGs)

Attacks on PRNGs

- In the next slides some attacks on PRNGs are presented:
  - Linear congruential generator (LCG)
  - Mersenne Twister
  - Linear-feedback shift register (LFSR)
- Attacks are mainly based on observation of the generated output numbers
- Knowledge of some information inside the algorithm can help with the attacks

Outline

- Introduction
- Attacks on LCGs
- Attacks on Mersenne Twister
- Attacks on LFSR

Recall: Linear Congruential Generator

- The simplest known PRNG is the Linear Congruential Generator (LCG)
- We fix three integers n, a, b respectively called modulus, multiplier and increment
- We fix a starting point x_0, the seed of the generator
- Next values are produced as $x_{i+1} = ax_i + b \ (mod\ n)$

Issues of LCGs

- LCGs can be easily broken if we have some observations from them
- We can attack LCGs in 4 different ways:
  1. n, a, b are known
  2. n, a are known
  3. only n is known
  4. nothing is known

Break LCGs: n, a and b known

- We know some observations $x_1, x_2, \dots, x_{k-1}, x_k$
- We know n, a, b
- How can we compute x_{k+1}?
  - Simply $x_{k+1} = ax_k + b \ (mod\ n)$

Break LCGs: n and a known

- We know some observations $x_1, x_2, \dots, x_{k-1}, x_k$
- We know n, a
- How can we find b?
  - $x_k = ax_{k-1} + b \ (mod\ n) \rightarrow b = x_k - ax_{k-1} (mod\ n)$
- Solve as previous scenario

Break LCGs: only n is known

- We know some observations $x_1, x_2, \dots, x_{k-1}, x_k$
- We know only n
- How can we compute a?
  - $x_k = ax_{k-1} + b \ (mod\ n), x_{k-1} = ax_{k-2} + b \ (mod\ n)$
  - $x_k - x_{k-1} = a(x_{k-1} - x_{k-2})$
  - $a = (x_k - x_{k-1})(x_{k-1} - x_{k-2})^{-1} (mod\ n)$
- Solve as previous scenario

Break LCGs: nothing is known

- We only know some observations $x_1, x_2, \dots, x_{k-1}, x_k$
- How can we compute n?
  - Write $x_k = ax_{k-1} + b \ (mod\ n), x_{k-1} = ax_{k-2} + b$ and so on
  - We known that $n \mid x_h - (ax_{h-1} + b) = s_h$ for every h
  - So $\gcd(s_1, \dots, s_k) = n$ with high probability… but we can't make the s_h sequence as we miss a and b
  - We can define a new sequence $t_i = x_i - x_{i-1}$ s.t $t_i = at_{i-1} (mod\ n)$
  - Note that $t_{i+1}t_{i-1} - t_i^2 = 0 \ (mod\ n)$ then recover n by applying the gcd
- Solve as previous scenario

Outline

- Introduction
- Attacks on LCGs
- Attacks on Mersenne Twister
- Attacks on LFSR

Mersenne Twister

- Another famous PRNG is the Mersenne Twister (MT)
- It is by far the most widely used PRNG in practice
- Its name derives from the Mersenne prime numbers, primes in the form of $2^n - 1$, used in the algorithm
- It is usually used in its MT19937 version, where 19937 means that you need 2^{19937} calls of the function to obtain a duplicate number from the PRNG

Issues of Mersenne Twister

- Even if widely used, is not cryptographically secure
- By observing enough iterations, 624 word of 32-bit in the MT19937 version, it is possible to recover the internal state vector from which future iterations are produced and allows one to predict all future iterations

Break *(also)* the Mersenne Twister

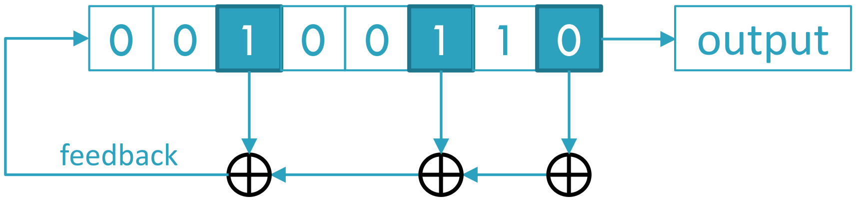
- *Untwister: a seed recovery tool for common PRNGs*
  - https://github.com/altf4/untwister
- *Supported PRNGs:*
  - *Mersenne Twister (MT19937 version)*
  - *Glibc's rand() PHP's MT-variant (php_mt_rand)*
  - *Ruby's MT-variant DEFAULT::rand()*
  - *Java's Random() class*

Outline

- Introduction
- Attacks on LCGs
- Attacks on Mersenne Twister
- Attacks on LFSR

Linear Feedback Shift Register

- LFSRs are used as PRNG with application for example in stream ciphers
- A LFSR is defined by:
  - A bit size
  - A characteristic polynomial
  - An initial state



An 8-bit LFSR with initial state of 00100110 and characteristic polynomial of $x^8 + x^6 + x^3 + 1$

Linear Feedback Shift Register

- LFSRs are used as PRNG with application for example in stream ciphers
- A LFSR is defined b[y]
  - A bit size
  - A characteristic polynom[ial]
  - An initial state

Further details can be found in the lectures:
- *HW_S_0.2.3 – Linear Feedback Shift Registers - LFRSs*

An 8-bit LFSR with initial state of 00100110 and characteristic polynomial of $x^8 + x^6 + x^3 + 1$

Recovering internal state

- Assume that the characteristic polynomial of a n-bit LFSR is known
- Is it possible to completely recover the internal state given a binary output sequence of length n
- With the internal state is it possible to go backward and forward and recover all the output sequence

Berlekamp Massey algorithm

- Given some binary observation of a LFSR is it possible to recover its characteristic polynomial
- Berlekamp–Massey algorithm is an algorithm that will find the shortest linear feedback shift register (LFSR) for a given binary output sequence
  - References: Weisstein, Eric W. "Berlekamp-Massey Algorithm." From *MathWorld* - A Wolfram Web Resource
  - https://mathworld.wolfram.com/Berlekamp-MasseyAlgorithm.html
- An Online Calculator of Berlekamp-Massey Algorithm:
  - http://bma.bozhu.me/

Attacks on PRNGs

Gaspare FERRARO
CyberSecNatLab
Matteo ROSSI
Politecnico di Torino

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it