Kryptografia, Lista 1

Adrian Mucha, Politechnika Wrocławska, WPPT 13/05/2020

Zad 1

W zadaniu użyto własny liniowy generator kongruencyjny.

$$lcg_{k+1} = (m \cdot lcg_k + c) \mod n = 15623 \cdot lcg_n + 1836716 \mod 21373737$$

Distinguisher *oracle* potrafi przewidywać wartości generowane przez lcg na podstawie zaobserwowanych wcześniej liczb. Distinguisher potrafi obliczyć składowe *c*, *m* oraz *n*.

Wszystkie znane komponenty

Wystarczy obliczyć kolejny stan generatora - przypadek trywialny.

Składowa c nieznana

Znając 2 stany s_k, s_{k+1} możemy obliczyć c.

$$s_{k+1} = s_k m + c \mod n$$
$$c = s_{k+1} - s_k m \mod n$$

Składowa m nieznana

Znając 3 stany s_k, s_{k+1}, s_{k+2} możemy obliczyć m

$$s_1 = s_0 m + c \mod n$$
$$s_2 = s_1 m + c \mod n$$
$$s_2 - s_1 = s_1 m - s_0 * m \mod n$$
$$s_2 - s_1 = m(s_1 - s_0) \mod n$$
$$m = (s_2 - s_1)(s_1 - s_0)^{-1} \mod n$$

Jedyną trudnością jest tutaj obliczanie odwrotności modulo, którą otrzymuje się poprzez zastosowanie rozszerzonego algorytmu Euklidesa.

$$ax + my = gcd(a, m) = 1$$
$$ax - 1 = (-y)m$$
$$ax \equiv 1 \mod m$$

Składowa n nieznana

Znając 6 stanów $s_k, s_{k+1}, \ldots s_{k+5}$ możemy obliczyć n.

Skorzystamy z dwóch własności

$$X = 0 \mod n \quad (1)$$

$$X = kn \mod n \quad (2)$$

oraz faktu, że z dużym prawdopodobieństwem, jeśli mamy kilka losowych liczb będących wielokrotnościami n, to ich największym wspólnym dzielnikiem będzię n.

Następnie możemy wprowadzić pewną sekwencję $T(n) = S(n+1) - S(n)$

$$t_0 = s_1 - s_0$$

$$t_1 = s_2 - s_1 = (s_1 m + c) - (s_0 m + c) = m(s_1 - s_0) = mt_0 \mod n$$

$$t_2 = s_3 - s_2 = (s_2 m + c) - (s_1 m + c) = m(s_2 - s_1) = mt_1 \mod n$$

$$t_3 = s_4 - s_3 = (s_3 m + c) - (s_2 m + c) = m(s_3 - s_2) = mt_2 \mod n$$

oraz wykorzystać wcześniej wspomniany fakt by obliczyć

$$t_2 t_0 - t_1 t_1 = (mmt_0 \cdot t_0) - (mt_0 \cdot mt_0) = 0 \mod n$$

a następnie obliczamy gcd i otrzymujemy moduł.

Na koniec warto wspomnieć, że program sprowadza przypadki od najtrudniejszego (znalezienie modułu), przez znalezienie mnożnika i na końcu do znalezienia stałej. W przypadku gdy tylko jedna z nich jest nieznana to reszta nie jest obliczana.

Zad 2

Opis glibc random

Na początek przybliżymy metodę działania `glibc`. Dla zadanego ziarna s, wektor inicjalizujący $r_0, r_1, \ldots, r_3 3$ jest obliczany według następującego schematu:

- $r_0 = s$
- $\forall_{i \in \{1,\ldots,30\}} \quad r_i = (16807 \cdot r_{i-1}) \mod (2^{31} - 1)$
- $\forall_{i \in \{31,\ldots,33\}} \; r_i = r_{i-31}$

Następnie, sekwencja pseudolosowa $r_{34}, r_{35}, \ldots$ jest generowana dzięki pętli liniowego sprzężenia zwrotnego

- $\forall_{i \geq 34} \quad r_i = (r_{i-3} + r_{i-31}) \mod 2^{32}$

$r_0, \ldots, r_{343}$ dezintegrujemy, a i-tym wyjściem o_i funkcji `myrand()` jest

- $o_i = r_{i+344} >> 1$

$>>$ jest tutaj przesunięciem bitowym w prawo (pozbywamy się najmniejznaczącego bitu).

Liniowość

Pomimo odrzucanego ostatniego bitu, mamy prawie liniową zależność otrzymywanej sekwencji wychodzącej. Mamy więc $\forall_{i \geq 31}$

$$o_i = o_{i-31} + o_{i-3} \mod 2^{31} - 1$$

lub

$$o_i = o_{i-31} + o_{i-3} + 1 \mod 2^{31} - 1$$

Nasz *distinguisher* potrafi więc odgadnąć wynik funkcji `glibc random`.