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.

$$\begin{aligned} s_{k+1} &= s_k m + c \mod n \\ c &= s_{k+1} - s_k m \mod n \end{aligned}$$

Składowa m nieznana

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

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

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

$$\begin{aligned} ax + my = gcd(a, m) &= 1 \\ ax - 1 &= (-y)m \\ ax &\equiv 1 \mod m \end{aligned}$$

Składowa n nieznana

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

Skorzystamy z dwóch własności

$$X = 0 \mod n \tag{1}$$

$$X = kn \mod n \tag{2}$$

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 $T(n) = S(n+1) - S(n)$

$$\begin{aligned} t_0 &= s_1 - s_0 \\ t_1 = s_2 - s_1 = (s_1 m + c) - (s_0 m + c) &= m(s_1 - s_0) = mt_0 \mod n \\ t_2 = s_3 - s_2 = (s_2 m + c) - (s_1 m + c) &= m(s_2 - s_1) = mt_1 \mod n \\ t_3 = s_4 - s_3 = (s_3 m + c) - (s_2 m + c) &= m(s_3 - s_2) = mt_2 \mod n \end{aligned}$$

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