шкалу оценок и коэффициенты для каждого из параметров, и при анализе ресурса делать оценку по шкале.

5. Вывод

Очевидно, что проблема интернет-прогнозирования может быть сведена к разработке адекватной математической модели и процессу рандомизации. Однако, перед разработкой модели, необходимо тщательно изучить параметры (регрессоры) и найти оптимальный метод оценки этих параметров. То есть, провести анализ каждого из параметров и построить необходимые шкалы оценок. При этом, необходимо опираться на текущую статистику посещаемости, которая может показать некоторые тенденции.

Если все параметры, влияющие на посещаемость, будут выделены и их удастся описать математическим языком, то вполне вероятно, что проблема прогнозирования в сети Интернет будет решена.

Литература

1. *Дрейнер С.* Прикладной регрессионный анализ. М.: Наука и техника,1986. 128 с.

О ПЕРИОДИЧЕСКИХ СВОЙСТВАХ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Т. А. Валуева

1. Введение

Псевдослучайные последовательности широко используются при решении задач методом имитационного моделирования, а также для защиты информации в компьютерных сетях.

Одно из требований, предъявляемых к генераторам псевдослучайных последовательностей, является требование к периоду, который должен быть достаточно большим.

В данной работе рассмотрены суммирующий генератор и криптосхема RC4. Для суммирующего генератора найден период выходной последовательности, в случае, когда периоды входных последовательностей совпадают. Для криптосхемы RC4 найдены слабые состояния.

2. Суммирующий генератор

2.1. Описание суммирующего генератора

Будем рассматривать генератор, предложенный в [1], который является двоичным нелинейным комбинирующим генератором с памятью.

Переменная внутреннего состояния суммирующего генератора (перенос) принимает целые значения от 0 до $n - 1$, где n – количество входов. Таким образом, размер памяти равен $m = \lceil \log_2 n \rceil$ бит.

Для описания суммирующего генератора будем рассматривать модель инициального конечного автомата. Пусть $X_t = (x_{1,t}, ..., x_{n,t})$ – n входных бит в момент времени t, $\{y_t\}$ – выходной бит в момент времени $t \geq 0$. Внутреннее состояние суммирующего генератора записывается как $S_t = \sum_{i=0}^{m-1} s_{i,t} 2^i$ и двоичное представление переноса имеет вид $S_t = (s_{0,t}, ..., s_{m-1,t})$. Начальное состояние $S_0 = 0$. Функции выходов и переходов в следующее состояние определяются следующим образом:

$$y_t = \sum_{i=1}^{n} x_{i,t} + s_{0,t} \pmod 2, \qquad (1)$$

$$S_{t+1} = \left\lfloor \frac{\left(\sum_{i=1}^{n} x_{i,t} + S_t \right)}{2} \right\rfloor.$$

2.2 Период выходной последовательности суммирующего генератора

Предложение 2.1. Пусть входные последовательности суммирующего генератора $x_i = \{x_{i,t}\}$, $1 \leq i \leq n$ – периодические последовательности с периодом T. Тогда период выходной последовательности $\{y_t\}$ суммирующего генератора равен T.

Доказательство. Рассмотрим последовательность $\{S_t\}$, где S_t значение переноса в момент времени t. Для доказательства леммы будем использовать следующие свойства $\{S_t\}$:

1. Последовательность $\{S_t\}$ ограничена, т. к. $0 \leq S_t \leq n - 1$.
2. $S_t \leq S_{t+T}$ для любого $t \geq 0$.

Докажем неравенство $S_t \leq S_{t+T}$, $t \geq 0$ методом математической индукции.

База индукции: $t = 0$: $S_0 = 0$, $S_T \geq 0 \Rightarrow S_0 \leq S_T$.

Предположим, что утверждение верно для всех $t <$ k. Найдем значения:

S_{k+1} и S_{k+1+T}: $S_{k+1+T} = \left\lfloor \frac{(\sum_{i=1}^{n} x_{i,k+1+T} + S_{k+T})}{2} \right\rfloor = \left\lfloor \frac{(\sum_{i=1}^{n} x_{i,k+1} + S_{k+T})}{2} \right\rfloor$,

$S_{k+1} = \left\lfloor \frac{(\sum_{i=1}^{n} x_{i,k+1} + S_k)}{2} \right\rfloor$. В силу предположения индукции $S_k \le S_{k+T} \Rightarrow$ $S_{k+1} \le S_{k+T+1}$.

Вернемся к доказательству леммы. После T шагов генератора возможны два случая.

Если $S_0 = S_{0+T}$, тогда из (1) следует, что T – период.

Если $S_0 < S_{0+T}$, то $S_{0+T} \ge 1$.

После следующих T шагов генератора получаем, что $S_0 = S_{0+2T}$, либо $S_{0+2T} \ge 2$. Теперь рассмотрим S_{0+iT}. Рассуждая аналогично случаям $i = 1$ и $I = 2$ получаем, что на шаге iT либо $S_{0+(i-1)T} = S_{0+iT}$, либо $S_{0+iT} \ge i$. Т. е., если за $(n-1)T$ не выполнилось равенство $S_{0+(i-1)T} = S_{0+iT}$ (из которого следует, что период выходной последовательности равен T), то $S_{0+(n-1)T} = n-1$. Тогда с одной стороны $S_{0+nT} \ge S_{0+(n-1)T}$, но с другой стороны $S_{0+nT} \le n-1 \Rightarrow$ единственно возможный случай, когда $S_{0+(n-1)T} = S_{0+nT} = S_{0+(n-1)T+T}$, что завершает доказательство предложения.

3. Генератор RC4

3.1. Описание генератора RC4

Криптосистема RC4 – семейство алгоритмов, зависящих от параметра (модуля) n, который рекомендуется брать равным 256. Состоянием криптосхемы являются подстановка $S = \begin{bmatrix} 0 & . & . & . & n-1 \\ s[0] & & & & s[n-1] \end{bmatrix}$ из n элементов, где *s[k]* – элемент, находящийся на k-том месте и два счетчика i, j. Будем обозначать с индексом t состояние (S_t, i_t, j_t) криптосхемы на в момент времени t. Начальные значения счетчиков $i_0 = 0, j_0 = 0$. Пусть – ключ длины L байт .

Формирование начального состояния S_0.

Пусть S_{-L} – тождественная подстановка.

Для каждого t, $1 \le t \le L$ вычисляем:

Функция перехода состояний F.

1. $i_t = i_t + 1 (\mathrm{mod}\, n)$,
2. $j_t = j_{t-1} + s_{t-1}[i_t] (\mathrm{mod}\, n)$,
3. переставляем $s_t[i_t]$ и $s_t[j_t]$.

Функция выхода f.

$z_t = s_t[(s_t[i_t] + s_t[j_t]) \bmod n]$.

3.2. Свойства выходной последовательности генератора RC4

Предложение 3.1. Если начальное состояние криптосхемы $S_0, i_0 = k (\mathrm{mod}\, n), j_0 = k + 1 (\mathrm{mod}\, n)$, причем $S_0[(k+1) \bmod n] = 1$, тогда криптосхема вернется в то же состояние через n $(n - 1)$ шагов.

Доказательство. Вычислим следующее состояние криптосхемы: $i_1 = i_0 + 1 = k + 1 (\mathrm{mod}\, n)$,

$j_1 = (j_0 + S_0[i_1]) \bmod n = (k + 1 + S_0[(k+1)]) \bmod n) \bmod n = (k+2) \bmod n$,

$S_1[j_1] = S_0[i_1] = S_0[(k+1) \bmod n] = 1$, $S_1[i_1] = S_0[j_1] = S_0[(k+2) \bmod n]$. Т. е. получили состояние, которое отличается от (S_0, i_0, j_0) тем, что $k' = k + 1$ и $S_1[k' \bmod n] = S_0[(k'+1) \bmod n] = 1$. Заметим, что через n шагов криптосхема вернется в состояние (S_n, i_0, j_0), где S_n – такая подстановка, что $S_n[(k+1) \bmod n] = 1$, $S_n[k \bmod n] = S_0[(k+2) \bmod n]$ и для $\forall l \in \mathrm{Z}$, $l \ne k, k + 1 (\mathrm{mod}\, n)$ $S_n[l \bmod n] = S_0[(l-1) \bmod n]$, т. е. за n шагов k-ый элемент сдвигается на две позиции влево, а остальные элементы подстановки сдвигаются на одну позицию вправо. Таким образом, для того, чтобы все элементы подстановки вернулись на свои места, потребуется $n - 1$ серия из n шагов. Т. е. получили, что через n $(n - 1)$ шагов криптосхема вернется в исходное состояние.

Литература

1. *Jovan Dj. Golic, Mahmoud Salmasizadeh, Ed Dawson* Fast Correlation Attacks on the Summation Generator // Journal of Cryptology, vol 13, pp. 245–262, 2000.

ИССЛЕДОВАНИЕ СВОЙСТВ УСТОЙЧИВЫХ ЗАКОНОВ РАСПРЕДЕЛЕНИЯ И ИХ КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

Л. Э. Гаджиева

1. Введение

В настоящее время для исследования различных экономических и финансовых моделей достаточно часто применяется вероятностный под-