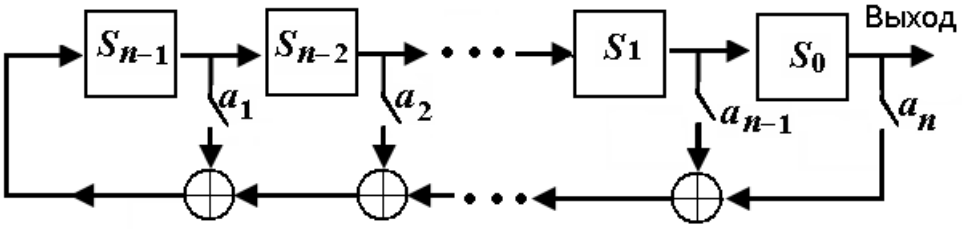
ПРАКТИЧЕСКАЯ КРИПТОЛОГИЯ
ЛЕКЦИЯ 11
Специальность: 6.170101 – Бсіт
Лектор: Сушко С.А.

§7. РЕГИСТРЫ СДВИГА С ЛИНЕЙНОЙ ОБРАТНОЙ СВЯЗЬЮ

В технике один из наиболее распространенных методов генерации битовых последовательностей реализуется с помощью ***регистров сдвига с обратной линейной связью*** (англ. Linear Feedback Shift Register – LFSR).

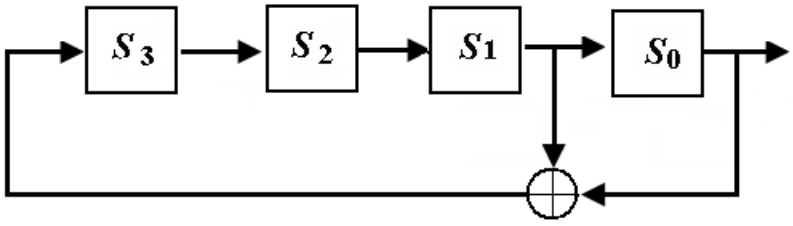


LFSR – микросхема с ячейками памяти, в которые записан один бит информации. Ячейки имеют по одному входу и выходу. Количество ячеек равно длине регистра. Вторая составляющая регистра – функция обратной связи. Для линейного регистра – это операция XOR над некоторыми его битами, которые называют ***отводами*** (***точками съемки***). Принцип работы регистра длины n с обратной линейной связью графически подан на рисунке. Регистр работает в дискретные моменты времени, в каждый из которых выполняются такие операции:

- содержание самой правой ячейки S_0 «выталкивается» из регистра и формирует очередной элемент генерированной последовательности;
- содержание ячейки S_i перемещается в ячейку S_{i-1}, $i = 0,1,...,n-1$;
- новое содержание самой левой ячейки – это бит обратной связи, который равен сумме по модулю 2 битов ячеек $S_0, S_1,\ldots, S_{n-1}$, умноженных на коэффициенты $a_1, a_2,..., a_n$.

Если некоторые из коэффициентов $a_1, a_2,..., a_n$ равны нулю, то соответствующие сумматоры $\oplus$ из цепи обратной связи исключают. В моменты времени $t = 0,1,2,...$ на выходе регистра генерируется последовательность $x_0, x_1, x_2,....$

Пример LFSR длины 4 с коэффициентами $a_1 = a_2 = 0,\ a_3 = a_4 = 1$ и начальным состоянием $(S_0, S_1, S_2, S_3) = (1,0,1,1)$:



	$S_0 S_1 S_2 S_3$	
$t=0$	1 0 1 1;	Генерируется последовательность
$t=1$	0 1 1 1;	1011110001001101
$t=2$	1 1 1 1;	
$t=3$	1 1 1 0;	
$t=4$	1 1 0 0;	
$t=5$	1 0 0 0	

.....

Содержание ячеек называется ***заполнением*** (***состоянием***) ***регистра*** (в начальный момент – это начальное заполнение регистра или вектор начального состояния). При условии, что содержание всех ячеек равно нулю, состояние регистра называют ***нулевым***. Если $x_i(t)$, $x(t)$ – заполнение i-ой ячейки памяти и выход регистра в момент времени t соответственно, то работу регистра описывают уравнения:

$$x_{i-1}(t+1) = x_i(t), \quad i = 0,1,...,n-1;$$
$$x_{n-1}(t+1) = f(x_0(t), x_1(t),..., x_{n-1}(t));$$
$$x(t+1) = x_0(t).$$

Выходная последовательность LFSR называется ***линейной рекуррентной последовательностью n-го порядка*** над полем $GF(2)$. Это последовательность битов, которые при всех $0 \le i < \infty$ удовлетворяют равенству:

$$x_{i+n} = a_n x_i + a_{n-1} x_{i-1} + ... + a_2 x_{i+n-2} + a_1 x_{i+n-1} = \sum_{j=1}^{n} a_j \cdot x_{i+n-j},$$

где все операции выполнены в поле $GF(2)$. Эта формула называется ***законом рекурсии***, который генерирует последовательность, а вектор $(x_0, x_1, ..., x_{n-1})$ – ***начальным вектором*** (***зародышем, состоянием***) последовательности.

Выходная последовательность LFSR единственным образом определяется многочленом обратной связи и начальным заполнением регистра. Для LFSR длины n с коэффициентами $a_1, a_2, ..., a_n$

многочлен обратной связи $\quad P(x) = 1 - \sum_{i=1}^{n} a_i x^i$

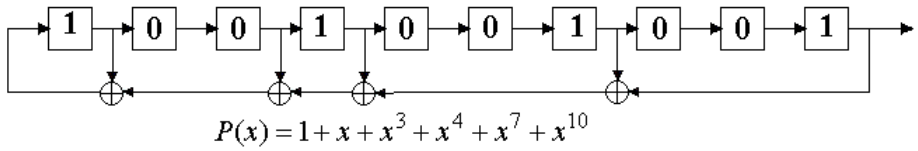
Альтернативно для задания выходной последовательности LFSR

характеристический многочлен $P^*(x) = x^n P\left(\frac{1}{x}\right) = x^n - \sum_{i=1}^{n} a_i x^{n-i}$

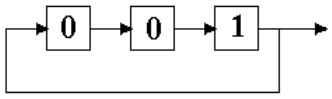
можно использовать

Для LFSR, изображенного на рисунке, многочлен обратной связи – это $P(x) = 1 + x^3 + x^4$, характеристический многочлен $P^*(x) = 1 + x + x^4$, закон рекурсии $x_{i+4} = x_{i+1} + x_i$.

Одну и туже последовательность могут генерировать разные LFSR, например, последовательность 1001001001… генерируют LFSR, представленные на следующем рисунке:



$P(x) = 1 + x + x^3 + x^4 + x^7 + x^{10}$



$P(x) = 1 + x^3$

Поэтому линейная рекуррентная последовательность может иметь разные характеристические многочлены. Характеристический многочлен наименьшей степени называется ***минимальным многочленом последовательности***. Ясно, что в примере для последовательности 1001001001… минимальным многочленом будет $F(\lambda) = 1 + \lambda^3$ (здесь и в дальнейшем минимальный многочлен последовательности обозначаем $F(\lambda)$).

§8. СВОЙСТВА ПОСЛЕДОВАТЕЛЬНОСТЕЙ $\{x_i\}$ НАД ПОЛЕМ $GF(2)$, СГЕНЕРИРОВАННЫХ LFSR

Пусть начальное заполнение регистра ненулевое. Тогда

1^0. Любая последовательность, генерированная LFSR, – периодическая, т.е. $x_{i+T} = x_i$ для некоторого числа T (при этом период полученной битовой последовательности называют ***периодом регистра***).

2^0. LFSR длины n не может иметь более, чем 2^n начальных состояний, поэтому

период последовательности, сгенерированной LFSR , не превышает значения $2^n - 1$.

3^0. Если старший коэффициент многочлена обратной связи $a_n = 0$, то последовательность имеет ***передпериод***, т.е. соотношение $x_{i+T} = x_i$ начинает выполняться с некоторого номера $i \geq \lambda$, где λ – предпериод.

4^0. Если $a_n = 1$, то генерированная последовательность называется ***неособой***. Она начинается сразу со своей периодической части.

5^0. Если минимальный многочлен последовательности неприводимый над полем $GF(2)$ и имеет степень n, то период последовательности является делителем числа $2^n - 1$ (*повторение*: многочлен $F(\lambda)$ – неприводимый над полем если, равенство $F(\lambda) = g(\lambda) \cdot h(\lambda)$ возможно в поле только при условии, что $g(\lambda)$ или $h(\lambda)$ – константа).

6^0. Если минимальный многочлен последовательности имеет степень n и будет примитивным над конечным полем $GF(2)$, то период последовательности максимален и равен $2^n - 1$ (*повторение*: многочлен $F(\lambda)$ – примитивный, если он является делителем многочлена $\lambda^{2^N - 1} + 1$ и не является делителем многочленов вида $\lambda^d + 1$ при всех d, на которые делится число $2^n - 1$).

Справедливо и обратное:

Период последовательности, сгенерированной LFSR длины n над полем $GF(2)$ принимает максимальное значение $2^n - 1$, тогда и только тогда, когда характеристический многочлен, образованный из отводов регистра, – примитивный над полем $GF(2)$.

Последовательности с максимально возможным периодом называют m***-последовательностями*** (***линейными рекуррентными последовательностями максимального периода***, ***максимальными линейными рекуррентными последовательностями***). При этом

- каждая m-последовательность – чисто периодическая;
- для битовых m-последовательностей (с примитивным минимальным многочленом) характерна ***сбалансированность***, т.е. появление одинакового числа 0 и 1 на периоде;

- на каждом периоде m-последовательности любая ненулевая подпоследовательность длины r появится только раз (***свойство де- Брюина***).

Регистры сдвига с обратной линейной связью, которые генерируют m-последовательности, называют ***регистрами максимальной длины***. Важная характеристика регистра максимальной длины – это ***объем ансамбля***. Так называют число различных m-последовательностей для заданного LFSR длины n.

LFSR – базовый элемент многих потоковых шифров. Это обусловлено высоким быстродействием алгоритмов, построенных на их базе; применением в регистрах только простых операций сложения и умножения над битами; способностью регистров генерировать последовательности с большим периодом и хорошими статистическими свойствами.

Заметим, что для потоковых шифров необходимы последовательности не просто с большим, а очень большим периодом, дабы избежать повторного использования гаммы при шифровании одного сообщения. Для генерации такой гаммы с помощью LFSR нужен примитивный многочлен, поэтому его следует уметь находить. Простого способа генерации примитивных многочленов над полем в целом не существует. Простейший способ – выбрать случайно многочлен и проверить его примитивность. Эта задача – сложная и нагадывает тестирование случайных чисел на простоту, но для ее решения разработаны специальные математические программные пакеты. Некоторые примитивные многочлены над полем $GF(2)$ есть в литературе, например у Шнайера, где приведены ненулевые коэффициенты многочленов. Например, запись $(40,5,4,3,0)$, означает, что многочлен $x^{40}+x^5+x^4+x^3+1$ – примитивный по модулю 2, 40-битовый LFSR генерирует новый бит, складывая биты 5ой, 4ой, 3^{ей} и 0ой ячеек. Период последовательности будет максимальным $2^{40}-1$.

Замечание 1. В литературе приведены только ***разряженные*** многочлены – они имеют много нулевых коэффициентов, что ослабляет шифры. Лучше использовать ***плотные*** примитивные многочлены с большим количеством ненулевых коэффициентов. В открытой литературе процедура их построения пока неизвестна не приведена.

Замечание 2: В 1997 р. японцы Макото Мацумото и Такудзи Нисимура разработали новый генератор «Вихрь Марсенна». Этот витковой регистр сдвига с обобщенной отдачей имеет фантастично большой период $2^{19937}-1$, равный числу Марсенна M_{19937}. Его характеристический многочлен насчитывает более 100 членов. Скорость работы в 2-3 раза выше, чем у обчного LFSR, но генератор, к сожалению, генератор оказался некриптостойким.

§9. АТАКИ НА LFSR

LFSR имеют простую линейную структуру, а поэтому шифры, которые их непосредственно используют, уязвимы к атакам.

Выделим два общих типа атак на LFSR длины n:

1. Атаки при известной структуре LFSR, когда криптоаналитик, перехватив отрезок n-битового шифротекста, раскрывает все следующие шифрованные тексты;

2. Атаки на основе выбранного открытого текста длиной $2n$-битов, когда структура LFSR неизвестна. Цель атаки – узнать отводы регистра, т.е. коэффициенты $a_0, a_1, ..., a_{n-1}$ формулы в законе рекурсии:

$$x_{i+n} = a_{n-1}x_{i+n-1} + a_{n-2}x_{i+n-2} + ... + a_1x_{i+1} + a_0x_i .$$

Начальное заполнение регистра $x_0, x_1, \ldots, x_{n-1}$ легко узнать по перехваченной гамме. При $i = 0,1,...,n-1$ запишем в матричной форме систему линейных сравнений относительно $a_0, a_1, ..., a_{n-1}$:

$$\begin{pmatrix} x_{n-1} & x_{n-2} & \ldots & x_1 & x_0 \\ x_n & x_{n-1} & \ldots & x_2 & x_1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{2n-3} & x_{2n-4} & \ldots & x_{n-1} & x_{n-2} \\ x_{2n-2} & x_{2n-3} & \ldots & x_n & x_{n-1} \end{pmatrix} \cdot \begin{pmatrix} a_{n-1} \\ a_{n-1} \\ \ldots \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} x_n \\ x_{n+1} \\ \ldots \\ x_{2n-2} \\ x_{2n-1} \end{pmatrix} (\mathrm{mod}\, 2),$$

Решение системы позволит найти структуру регистра.

Пример. По последовательности битов $1,0,1,1,1,0,1,0,1,0$, созданной 5-битовым LFSR, найти точки съемки регистра.

Р е ш е н и е. Регистр 5-битовый ($n = 5$), закон рекурсии имеет вид

$$x_{5+i} = a_4x_{i+4} + a_3x_{i+3} + a_2x_{i+2} + a_1x_{i+1} + a_0x_i, \qquad i \geq 0,$$

де $a_0, a_1, ..., a_4$ – неизвестные. Нумеруем биты от 0 до 9 ($x_0 = 1$, $x_1 = 0$,…, $x_9 = 0$) и подставляем в матричное уравнение:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix} (\mathrm{mod}\, 2) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

Решаем систему методом исключения в поле $GP(2)$:

$$A^* = \left(\begin{array}{ccccc|c} 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{array}\right) (\mathrm{mod}\, 2) \sim \left(\begin{array}{ccccc|c} 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right) \Rightarrow$$

$$\Rightarrow \begin{cases} a_4x_4 + a_3x_3 + a_2x_2 + \quad + a_0x_0 = 0, \\ \quad a_3x_3 + a_2x_2 + a_1x_1 \quad = 1, \\ \quad a_3x_3 + \quad + a_1x_1 \quad = 0, \\ \quad a_3x_3 \quad + a_1x_1 + a_0x_0 = 1, \\ \quad a_3x_3 \quad = 0. \end{cases} \Rightarrow \begin{cases} a_0 = 1, \\ a_1 = 0, \\ a_2 = 1, \\ a_3 = 0, \\ a_4 = 0. \end{cases}$$

Закон рекурсии $x_{5+i} = x_{i+2} + x_i$, $i \geq 0$, новый бит – это «хорирование» битов 2ой и 0ой ячеек регистра.

$\Rightarrow$ Потоковый шифр, построенный только на одном LFSR беззащитный перед атаками.

§10. Линейная сложность последовательностей

Цель криптоаналитика при вскрытии потокового шифра – разгадать, как найти следующий член шифрующей последовательности, зная ее предыдущие члены. Так как для любой периодической последовательности над конечным полем можно найти РСОЛС, который будет ее генерировать, то задача сводится к подбору регистра, способного формировать гамму. Закон рекурсии зависит от минимального многочлена, поэтому главное мера криптографического качества гаммы – степень этого многочлена, Ее в теории называют линейной сложностью последовательности.

Определение линейной сложности последовательности. Будем говорить, что РСОЛС генерирует бесконечную числовую последовательность $X = \{x_0, x_1, ...\}$, если существует такое его начальное заполнение, при котором последовательность, рожденная генератором, совпадает с X. ***Линейной сложностью (линейным размахом) бесконечной битовой последовательности*** X называется число $L(X)$, определенное условиями:

- $L(X) = 0$ для нулевой последовательности $X = \{0, 0, 0, ...\}$;
- $L(X) = \infty$, когда не существует РСОЛС, который генерировал бы последовательность;

• $L(X)$ – длина кратчайшего регистра, который генерирует последовательность.

$\Rightarrow$ линейная сложность линейной рекуррентной последовательности – это степень ее минимального многочлена.

Аналогично: ***линейная сложность конечной битовой последовательности*** X_N – это длина $L(X_N)$ кратчайшего регистра, который генерирует последовательность, первые члены которой есть з первые члены последовательности X_N.

Свойства линейной сложности последовательностей:

1^0. Для любого $N > 0$ линейная сложность $0 \le L(X_N) \le N$.

2^0. $L(X_N) = 0$ только если X_N – нулевая последовательность длины N.

3^0. $L(X_N) = N$, если $X_N = \{0,0,...,0,1\}$.

4^0. если X – периодическая последовательность с периодом T, то $L(X) \le T$.

5^0. Линейная сложность чисто периодической последовательности равна степени ее минимального многочлена.

6^0. если X и Y – битовые последовательности, то

$$L(X) + L(Y) - 2\text{НОД}(T_X, T_Y) \le L(X \oplus Y) \le L(X) + L(Y),$$

где T_X и T_Y – периоды последовательностей. Если $\text{НОД}(T_X, T_Y) = 1$, то $L(XY)$ достигает максимума.

7^0. Если минимальный многочлен РСОЛС – примитивный многочлен степени L над полем $GF(q)$, то каждое ненулевое начальное заполнение дает на выходе последовательность со сложностью L.

§11. Алгоритм Берлекемпа – Месси

Служит для определения линейной сложности последовательностей, сформированных линейным регистром сдвига минимальной длины. Предложен Берлекемпом в 1968 г., а через год Месси интерпретировал алгоритм для линейных кодов.

На вход алгоритма подается битовая последовательность $X_n = \{x_0, x_1, ..., x_{n-1}\}$ длины n. Алгоритм выполняет n итераций, на каждой итерации, определяя минимальный многочлен и линейную сложность подпоследовательности из первых битов последовательности. На выходе алгоритма – минимальный многочлен всей последовательности

$$F(\lambda) = 1 + a_1\lambda + a_2\lambda^2 + ... + a_L\lambda^L$$

и значение ее сложности $L(X_n)$, $0 \le L(X_n) \le n$.

Пусть конечная битовая последовательность $X_N = \{x_0, x_1, ..., x_{N-1}\}$ сгенерирована РСОЛС, которому отвечает минимальный многочлен

$$F(\lambda) = 1 + a_1\lambda + a_2\lambda^2 + ... + a_L\lambda^L.$$

и пусть $L(X_N)$ – ее линейная сложность.

Алгоритм Берлекемпа - Месси

1. Задать $F(\lambda) = 1;\ \ L = 0;\ \ m = -1;$
   $G(\lambda) = 1;\quad N = 0.$
2. Пока $N < n$, выполнять такие шаги:
   - вычислить разность между двумя последовательными состояниями генератора $d = (x_N + \sum_{i=1}^{L} a_i x_{N-i}) \bmod 2;$
   - если $d = 1$, то
     - $T(\lambda) = F(\lambda);\qquad F(\lambda) = F(\lambda) + G(\lambda)\cdot\lambda^{N-m};$
     - если $L \le \frac{N}{2}$, то $L = N + 1 - L,\ \ m = N,\ G(\lambda) = T(\lambda).$
   - положить $N = N + 1.$

Сложность алгоритма $O(n^2)$, где n – длина последовательности.

Пример. С помощью алгоритма Берлекемпа – Месси найти минимальный многочлен и линейную сложность последовательности $X_{15} = \{1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0\}$, сгенерированной РСОЛС.

Р е ш е н и е.

Задаем $F(\lambda) = 1$ (это означает $a_0 = 1,\ a_1 = a_2 = ... = a_L = 0$); $L = 0;$ $m = -1;\ G(\lambda) = 1;\ N = 0.$

Итерации:

1) $N = 0,\ L = 0.$

$$d = (x_0 + \sum_{i=1}^{0} a_i x_{N-i}) \bmod 2 = x_0 = 1.$$

$$T(\lambda) = F(\lambda) = 1;\ \ F(\lambda) = 1 + 1\cdot\lambda^{0-(-1)} = 1 + \lambda.$$

$$0 \le \frac{0}{2} \Rightarrow L = N + 1 - L = 0 + 1 - 0 = 1.$$

$$m = N = 0;\quad G(\lambda) = T(\lambda) = 1;\quad N = N + 1 = 0 + 1 = 1.$$

2) $N = 1,\ L = 1.$

$$d = (x_1 + \sum_{i=1}^{1} a_i x_{N-i}) \bmod 2 = (x_1 + a_1 x_0) \bmod 2 = (1 + 1\cdot 1) \bmod 2 = 0.$$

Так как $d=0$, то $T(\lambda)$, $F(\lambda)$, L, m и $G(\lambda)$ не меняются.

$N=1+1=2$.

3) $N=2$, $L=1$;

$$d=(x_2+\sum_{i=1}^{1}a_i x_{N-i})\bmod 2=(x_2+a_1x_1)\bmod 2=(0+1\cdot 1)\bmod 2=1$$

;

$T(\lambda)=F(\lambda)=1+\lambda$; $F(\lambda)=1+\lambda+1\cdot\lambda^{2-0}=1+\lambda+\lambda^2$;

$1\le\frac{2}{2}\Rightarrow L=2+1-1=2$;

$m=N=2$; $G(\lambda)=T(\lambda)=1+\lambda$; $N=2+1=3$.

4) $N=3$, $L=2$;

$$d=(x_3+\sum_{i=1}^{2}a_i x_{N-i})\bmod 2=(x_3+a_1x_2+a_2x_1)\bmod 2=$$

$$=(0+1\cdot 0+1\cdot 1)\bmod 2=1;$$

$T(\lambda)=F(\lambda)=1+\lambda+\lambda^2$;

$F(\lambda)=1+\lambda+\lambda^2+(1+\lambda)\cdot\lambda^{3-2}=1+\lambda+\lambda^2+\lambda+\lambda^2=1$.

$2\ge\frac{3}{2}\Rightarrow L=2$. Неравенство $L\le\frac{N}{2}$ не выполняется, поэтому L, m и $G(\lambda)$ не меняется.

$N=3+1=4$ и т.д.

В итоге получим $F(\lambda)=1+\lambda+\lambda^2+\lambda^3+\lambda^8$, а линейная сложность последовательности $L(X_{15})=8$.

Алгоритм Берлекемпа – Месси – это универсальная криптоатака на любой генератор гаммы, так как позволяет заменить его эквивалентным кратчайшим РСОЛС.

Вывод: высокая линейная сложность гаммы – необходимое условие существования стойкого потокового шифру. Но высокая линейная сложность еще не означает пригодности гаммы для шифрования. Например, последовательность $(\underbrace{0,0,0,...,0}_{l-1},1)$ имеет сложность l, но не является шифрующей.

§12. Профиль линейной сложности последовательности.

В криптографию это понятие ввел в 80-х гг. XX ст. швейцарец Райнер Рюппель.

Пусть $X=\{x_0,x_1,...\}$ – бесконечная битовая последовательность; $L(X_N)$ – линейная сложность ее перших N членов $X_N=\{x_0,x_1,...,x_{N-1}\}$.

Последовательность $L(X_1)$, $L(X_2)$, … называется ***профилем линейной сложности бесконечной последовательности*** $X=\{x_0,x_1,...\}$. Аналогично последовательность $L(X_1)$, $L(X_2)$, …, $L(X_N)$ – это ***профиль линейной сложности конечной последовательности*** $X_N=\{x_0,x_1,...,x_{N-1}\}$. Это понятие характеризует сложность алгоритма Берлекемпа – Месси.

Свойства профилю линейной сложности:

1^0. если $j>i$, то $L(X_j)\ge L(X_i)$.

2^0. $L(X_{N+1})>L(X_N)$ тогда и только тогда, когда $L(X_N)\le N/2$.

3^0. если $L(X_{N+1})>L(X_N)$, то $L(X_{N+1})+L(X_N)=N+1$.

4^0. Профиль линейной сложности истинно случайных последовательностей неограниченно приближается к прямой $L=N/2$.

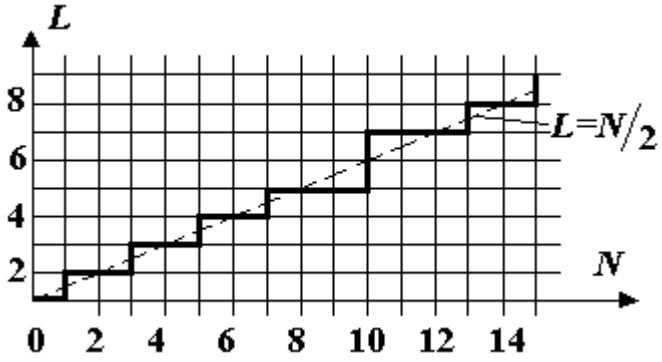


Рис. 6.3

Профиль строят, соединяя точки $(N;\ L(X_N))$ горизонтальными и вертикальными прямыми. на рис. 6.3 график профиля $0,1,1,2,2,3,3,4,4,4,6,6,6,7,7,8$ последовательности из примера..

Для последовательности, близкой к свойствам истинно случайной, профиль приближается к прямой $L=N/2$. Иначе последовательность – неслучайна. Это использовано при тестировании последовательностей с помощью графического теста по исследованию сложности.

Но если последовательность и проходит тест на линейную сложность, этого явно недостаточно, чтобы считать ее случайной.

§13. Классификация криптоатак на потоковые шифры

Методы криптоанализа потоковых шифров делятся на 3 класса.

1) ***Силовые атаки***: методы полного перебора всех возможных ключей (***атака «грубой силы»***). Преимущество отдают тем шифрам, для которых взлом полным перебором эффективнее других атак.

2) ***Статистические атаки***, суть которых в определении статистических свойств гаммы разными статистическими тестами. Эти атаки делят на два подкласса:

▪ I подкласс – ***методы криптоанализа статистических свойств гаммы*** для предсказания следующего бита гаммы по известным предыдущим;

▪ II подкласс – ***методы криптоанализа сложности гаммы***, цель которых исследовать сложность последовательности и установить, сложно ли восстановить закон генерирования битов.

3) ***Корреляционные атаки***, базирующиеся на аналитических принципах взлома шифров. Цель – установить использованный ключ (начальное заполнение регистров).

§14. ПОСТУЛАТЫ ГОЛОМБА

Какими же статистическими свойствами должна обладать периодическая последовательность, чтобы ее можно было считать псевдослучайной? Три таких требования к последовательностям сформулировал математик Агентства Национальной Безопасности США Соломон Голомб, в связи с чем их называют ***постулатами Голомба***.

Обозначим

- $x = \{x_0, x_1, ..., x_{T-1}\}$ – периодическая последовательность битов с периодом T;
- n_1, n_0 – количество 1 и 0 последовательности;
- отрезок длины s – подпоследовательность из s одинаковых символов, ограниченная другими символами:

$$\text{блок } 0\overbrace{11...1}^{s}0 \quad \text{или} \quad \text{лакуна } 1\overbrace{00...0}^{s}1;$$

- $n_1^{(s)}$, $n_0^{(s)}$ – число блоков и лакун на периоде;
- $\{x_i \oplus x_{i+d}\}$; $i = 0,1,2,...$; $d = 0,1,2,...,T-1$ – результат «XOR-вания» исходной последовательности и ее копии, сдвинутой на величину d (индексы вычисляют по модулю T);
- $n_1(d)$, $n_0(d)$ – число блоков и лакун в последовательности $\{x_i \oplus x_{i+d}\}$.

Отношение $AC(d) = \dfrac{n_1(d) - n_0(d)}{T}$ называют ***функцией автокорреляции последовательности*** $\{x_0, x_1, ..., x_{T-1}\}$.

Постулаты Голомба:

1^0. $\left|n_1 - n_0\right| \le 1$ – число 1 и 0 на каждом периоде может отличаться не более, чем на 1.

2^0. На периоде половина отрезков может иметь длину $s=1$, четверть отрезков – длину $s=2$; восьмая часть отрезков – длину $s=3$ и т.д. На периоде должна быть одинаковое количество блоков и лакун.

3^0. Функция автокорреляции должна быть бути двузначной (это означает, что если последовательность на периоде сравнить с этой же последовательностью, но циклически сдвинутой на любое число битов (не равное нулю или периоду), то число несовпадений будет на единицу больше, чем число совпадений). Это необходимое условие независимости битов последовательности: совпадение последовательности и ее сдвинутых копий не дает информации о периоде.

Пример. Проверить постулаты Голомба для m-битовой последовательности 10110011111000110111010100001 00, сгенерированной LFSR, если многочлен, построенный из последовательности отводов регистра $F(\lambda)=\lambda^5+\lambda^2+1$.

Р о з в’ я з а н н я. Минимальный многочлен последовательности примитивный, ее период $2^5-1=31$. Имеем

– $n_1^{(1)}=16$ и $n_0^{(1)}=15 \Rightarrow \left|n_1^{(1)}-n_0^{(1)}\right|\le 1$;

– на периоде последовательности 14 отрезков разной длины.

Тип отрезка, его длина	число отрезков	Частоты
010, $s=1$ 101, $s=1$	3 4	$1/2$
1001, $s=2$ 0110, $s=2$	1 2	$3/14\approx 0,21$
10001, $s=3$ 01110, $s=3$	1 1	$1/7\approx 0,14$
100001, $s=4$	1	$1/14\approx 0,07$
0111110, $s=5$	1	$1/14\approx 0,07$

– строим сдвинутые копии последовательности на $d=0,1,2,...,30$ битов, считаем блоки и лакуны в последовательностях $\{\gamma_i\oplus\gamma_{i+d}\}$:

$$\{x_i\}=10110011111000110111010100001 00$$

$$d=1\Rightarrow\{x_{i+1}\}=01100111110001101110101000010 01$$

$$\{x_i\oplus x_{i+1}\}=11010100001001011001111100011 01.$$

Есть 4 блока 010 и лакуны 101 $\Rightarrow$ $n_1(1)=4$ и $n_0(1)=4$ $\Rightarrow$ $AC(1)=\dfrac{n_1(1)-n_0(1)}{31}=0$.

Аналогично считаем $AC(2)=1/31$ $AC(3)=0$; $AC(4)=0$; $AC(5)=1/31$; $AC(6)=0$; $AC(7)=0$. Функция автокорреляции имеет два значения. Постулаты Голомба выполнены.

Последовательности, удовлетворяющие постулатам Голомба, называют ***псевдошумовыми*** (***ПШ-последовательностями***).

Все линейные рекуррентные последовательности максимального периода – псевдошумовые.

Постулаты Голомба обязательно должны выполняться для гаммы потокового шифра, но этого недостаточно, чтобы она имела высокую криптостойкость.

Соломон Вольф Голомб (англ. Solomon Wolf Golomb, родился в 1932 г.) – математик, инженер, профессор электротехники в университете Южной Калифорнии.

Специализируется на задачах комбинаторного анализа, теории кодирования. Докторскую диссертацию по теории чисел защитил в Гарвардском университете (1957 г.). Обнаружил особенности псевдослучайных m-последовательностей максимальной длины, которые имеют широкое военное, промышленное и криптографическое применение. Разработал так называемое кодирование Голомба – один из видов энтропийного кодирования. Его имя носит линейка Голомба, широко распространенная в радиосвязи, астрономии и теории шифрования. Награжден медалью Агентства Национальной безопасности США, медалью Ломоносова от Российской академии наук, в 2000 р. получил медаль Ричарда Хемминга.

Был одним из первых профессоров, кто прошел высший IQ-тест Хофлика, показав уровень IQ в 176 баллов. Если сравнить этот показатель с результатами других ученых, то коэффициент уникальности Голомба составил 1/1000000.

§15. ГРАФИЧЕСКИЕ ТЕСТЫ ДЛЯ СТАТИСТИЧЕСКОЙ ОЦЕНКИ КАЧЕСТВА ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Сформулированные требования можно заменить следующим: последовательность псевдослучайна, если она успешно пройдет все тесты из статистического набора тестов за конечное время. В криптографии ***набором статистических тестов*** называют совокупность статистических критериев, предназначенных для проверки соответствия анализируемой последовательности гипотезе о независимости и равновероятности ее элементов. Каждый тест состоит

в вычислении по анализируемой последовательности некоторой статистики, имеющей известное распределение для истинной случайной последовательности, и использовании критерия согласия. Стандартными наборами статистических тестов являются набор тестов Д.Кнута, пакет DIEHARD (Дж. Марсальи), набор тестов NIST (Института стандартов США), пакет TestUOT (Л'Эйкуера). Генератор, который проходит все статистические тесты набора, называется ***статистически безопасным***. Это означает, что

- ни один статистический тест не обнаруживает в выходной последовательности генератора каких-либо закономерностей;
- каждый бит сгенерированной им последовательности возникает в результате сложного преобразования над всеми начальними битами (зародышем);
- получив на вход разные начальные «зародыши», генератор выдает на выходе статистически независимые псевдослучайные последовательности.

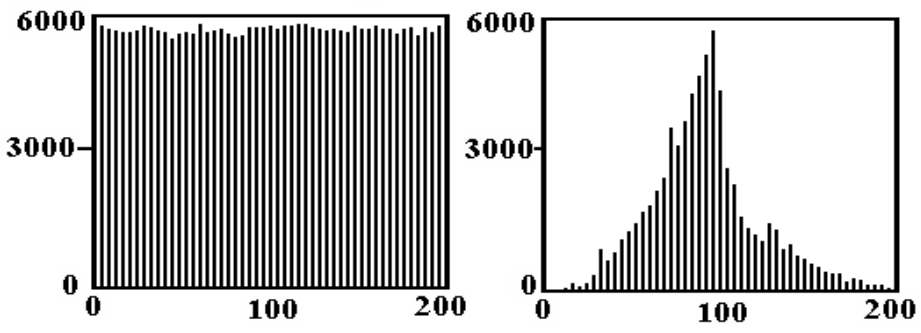
Все статистические тесты делятся на две группы:

1) ***графические тесты***, где статистические свойства последовательностей изображают с помощью различных графических зависимостей.

2) ***оценочные тесты***, анализирующие близость исследуемой и истинно случайной последовательности с помощью оценочных критериев.

Графические тесты

1) ***Гистограмма распределения элементов последовательности*** для оценки его равномерности и определения частоты повторения символов последовательности (по стандартной методике построения гистограмм). Последовательность проходить тест, если в ней есть все возможные члены, а разброс частот символов стремится к нулю, как на рис. 1.а, и не проходит тест в противном случае (на рис. 1.б закон распределения символов последовательности близкий к нормальному, резко отличается от равномерного).



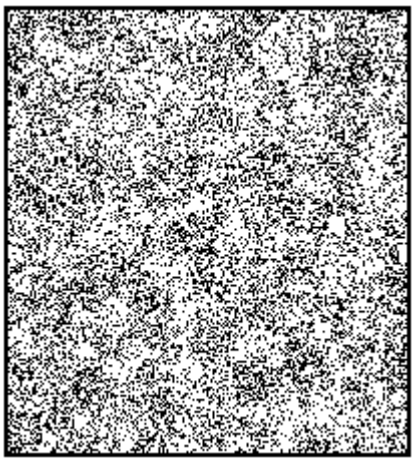
а) б)

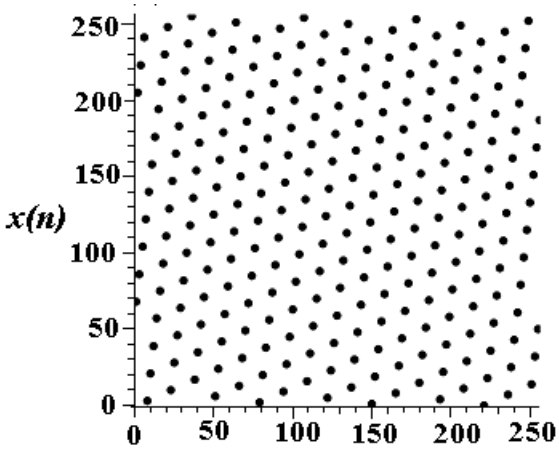
Рис.1

2) ***Распределение на плоскости***. На поле размером $(2^R-1)\times(2^R+1)$, где R – разрядность чисел последовательности, строят точки с координатами $(x_i;\ x_{i+1})$, где x_i – члены последовательности; $i=1,2,...,n$; n – ее длина. Если на поле точки лежат хаотично, то члены последовательности независимы, а если есть какие-то «узоры», то последовательность – неслучайна. Например, решетчастая структура линейного конгруэнтного генератора

$$s_{i+1}\equiv 137s_i+187(\mathrm{mod}\,256)$$

$\{s_i\}=1,\ 68,\ 31,\ 82,...$; точки $(1;\ 68),\ (68;\ 31),\ (31;\ 82),...)$.





решетчастая структура

3) ***Графическая проверка серий*** с целью определения равномерности распределения символов последовательности путем анализа частоты появления 0 и 1, и разных k-грамм (без перекрытия). В последовательности подсчитывают число 0, 1, биграмм (00, 01, 10, 11), триграмм (000, 001, 010, 011, 100, 101, 110, 111).

Например, $(2,\ 5,\ 4,\ 1,\ 8,\ 6)_{10}=00100101010000011000011 0$,

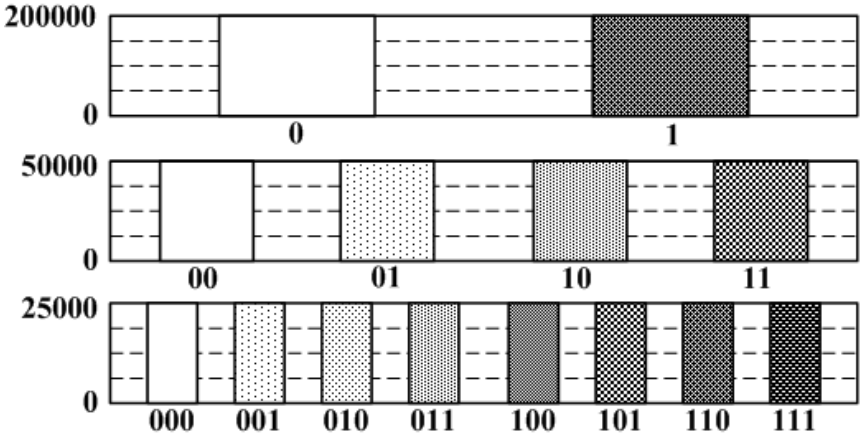
$\Rightarrow\ n_0=16,\ \ n_1=8$;

$00\ 10\ 01\ 01\ 01\ 00\ 00\ 01\ 10\ 00\ 01\ 10 \Rightarrow n_{00}=4,\ n_{01}=5,\ n_{10}=3,\ n_{11}=0$;

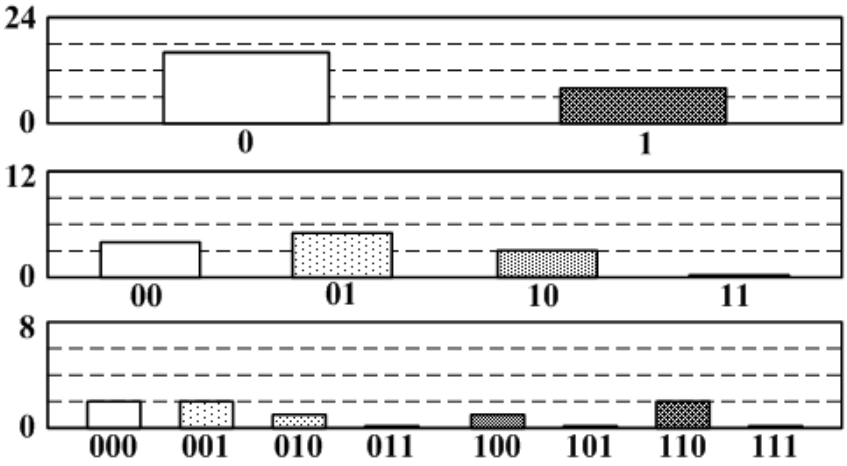
$001\ \ 001\ \ 010\ \ 100\ \ 000\ \ 110\ \ 000\ \ 110\ \ \Rightarrow n_{000}=2,\ \ n_{001}=2,\ \ n_{010}=1,$

$n_{011}=0,\ n_{100}=1,\ n_{101}=0,\ n_{110}=2,\ n_{111}=0.$

Если последовательность близка к случайной, то разброс между числом 0 и 1, числом разных серий-пар и серий-троек стремится к 0.



Тест 3 пройден



Тест 3 не пройден

4) ***Графическая проверка монотонности*** для оценки равномерности распределения символов у последовательности путем сравнения длин отрезков невозростания и неубывания членов. Например, $1,3,4,5,5,3,3,2,1,2,2,2,2,3,4$. Здесь 2 отрезка неубывания длиной 5 и 6, отрезок невозростания длиной 4. В последовательности, близкой к случайной, вероятность появления отрезка монотонности зависит от его длины: чем она больше, тем меньше вероятность.

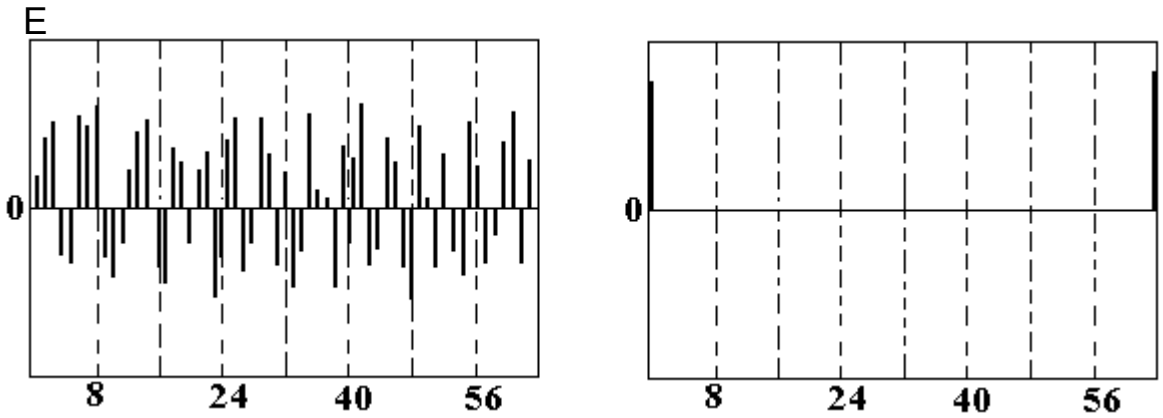
Тест 4 пройден

Тест 4 не пройден

5) ***Автокорреляционная функция*** для обнаружения корреляции между сдвинутыми копиями исследуемой последовательности. Для этого члены битовой заданной последовательности $x_0, x_1, ..., x_{n-1}$ заменяют по правилу: $1 \to 1, \quad 0 \to -1$. Общий член новой последовательности $b_i = (-1)^{1-x_i}, \quad i = 0,1,2,...,n-1$. Далее находят всплески корреляции

$$c_j = \frac{\sum_{i=0}^{n-1} b_i \cdot b_{(i+j) \bmod n}}{\sum_{i=0}^{n-1} b_i^2}, \; j = 0,1,...,n-1.$$

Для последовательности, близкой к случайной, всплески стремятся к 0 во всех точках, кроме кратных длине последовательности. Если есть много таких всплесков, то следует заподозрить зависимость между членами.



Тест 5 не пройден Тест 5 пройден

Еще есть тест ***построение профиля линейной сложности*** и ***графический спектральный тест***.

§16. ПАКЕТ NIST СТАТИСТИЧЕСКИХ ТЕСТОВ ДЛЯ ОЦЕНКИ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Пакет состоит из 16 специальных тестов и разработан Национальным институтом стандартов и технологий (NIST) США. Есть программная реализация пакета для платформы Unix. Если последовательность не проходит хоть один тест сразу означает, что она – неслучайна. В каждом тесте рассчитываются различные параметры, которые затем сравнивают с эталонными аналогичными параметрами, полученными для истинно случайных последовательностей.

1) ***Частотный тест*** устанавливает, какая часть битов составляют 0 и 1.

2) ***Блоковый тест на частоту*** проверяет равномерность появления 0 и 1 в подпоследовательностях. Последовательность разбивают на непересекающиеся блоки по M битов и вычисляют частоту повторения 0 и 1 в блоках.

3) ***Тест на «дырки»*** определяет равномерность распределения 0 и 1, анализируя цепочки из одних 0 или 1 («дырок»). Количество таких цепочек не должно быть слишком большим или малым.

4) ***Тест на максимальный размер серии 1*** анализирует равномерность распределения битов, сравнивая длины наибольших цепочек из 1 в данной и в истинно случайной последовательностях.

5) ***Проверка рангов матриц***. Тест оценивает равномерность распределения 0 и 1, анализируя линейные зависимости между подпоследовательностями. Для этого тест формирует из членов последовательности матрицы и находит их ранги.

6) ***Спектральный тест*** исследует периодические свойства последовательности по высотам пиков дискретного преобразования Фурьє.

7) ***Проверка повторения непересекающихся шаблонов.*** Тест выявляет генераторы, часто порождающие одинаковые непериодические шаблоны определенного типа. Последовательность разрезают на m-граммы, впритык друг к другу, которые затем сравнивают с шаблонами.

8) ***Проверка повторения пересекающихся шаблонов***. Тоже, что и в предыдущем тесте, но теперь m-граммы могут пересекаться.

9) ***Универсальный тест Маурэра*** определяет возможную степень сжатия последовательности на основе вычисления логарифма расстояния между двумя одинаковыми обнаруженными шаблонами, Если последовательность можно сжать без потери информации, то она – неслучайна.

Ули Маурер, профессор компьютерных наук и руководитель группы Cryptography Research в Швейцарском федеральном технологическом институте (ETH) в Цюрихе. Его научные интересы включают информационную безопасность, теоретическую криптографию (новые парадигмы, безопасность доказательств), применение криптографии (например, хэш-функции, симметричное шифрование, цифровые подписи с открытым ключом инфраструктур, цифровые платежные системы, электронное голосование). Маурер получил степень доктора философии (1990) в Цюриха. С 1990 по 1991 он был научным сотрудником факультета компьютерных наук в Принстонском университете, а в 1992 году он вступил в CS отдела ETH Zurich, где является профессором. В настоящее время редактор главного журнала по криптологии, член совета директоров Международной ассоциации по Cryptologic исследований (IACR). Член IEEE, член Германской академии наук, преподаватель кафедры математики в Университете штата Пенсильвания. Консультировал многие компании и государственные организации по вопросам информационной безопасности. , Входит в состав научно-консультативного совета Pricewaterhouse Coopers. Он является соучредителем в Цюрихе компании по безопасности программного обеспечения на основе Seclutions, имеет несколько патентов на криптографические системы.

10) ***Проверка сжатия с помощью алгоритма Лемпела – Зива*** выявляет чрезмерное сжатие – это серьезный дефект.

11) ***Проверка линейной сложности***.

12) ***Серийный тест*** анализирует неравномерность распределения m-битовых слов (блоков). К данной последовательности присоединяют $(m-1)$ первых ее битов. В полученном битовом ряде подсчитывают частоты повторения пересекающихся m-грамм, $(m-1)$-грамм, $(m-2)$-грамм и т.д.

13) ***Проверка аппроксимированной энтропии***. Тест также проверяет неравномерность распределения m-битовых слов, сравнивая частоты перекрытия двух последовательных серий в заданной и истинно

случайной последовательностях. Перед началом работы тест строит битовый ряд, как в серийном тесте.

В тестах 14 – 16 из членов данной последовательности $\{x_i\}$ строят новую последовательность $\{b_i\}$, где $b_i = 2x_i - 1$ (очевидно это соответствует замене $1 \to 1,\ 0 \to -1$).

14) ***Проверка кумулятивных сумм*** выявляет чрезмерное количество 0 и 1 вначале последовательности. Вычисляют максимальные отклонения сумм $z_1 = \max\limits_{1 \le k \le n}\left(\sum\limits_{i=1}^{k} b_i\right)$ и $z_2 = \max\limits_{1 \le k \le n}\left(\sum\limits_{i=n-k+1}^{n} b_i\right)$ от нуля.

15) ***Проверка случайных отклонений-1***. Вычисляют суммы $S_i = \sum\limits_{j=1}^{i} b_j$, образуют ряд $0, S_1, S_2, ..., S_n, 0$, который разбивают на блоки так, чтобы только первый и последний члены блоков были 0. Пусть x – значения ненулевых членов блока. Считают, что $x = -4, -3, -2, -1, 1, 2, , 3, 4$. Далее анализируют с помощью критерия χ^2, будет ли число блоков, в которых x встречается k раз ($k = 0,1,2,3,4,5$), отличаться от подобного числа, вычисленного для истинно случайных чисел.

16) ***Проверка случайных отклонений-2*** Тоже, что и предыдущий блок, но теперь $x = -9, -8, -7, ..., 8, 9$.

Для тестирования генератора исследуют с помощью тестов не одну, а множество рожденных им последовательностей и устанавливают, какая часть из них не проходит испытания. Если уровень неудач близкий к тому, который ожидают для истинно случайных чисел, то генератор – качественный.

Важно: генератор должен пройти <u>все</u> тесты пакета NIST.

Что происходит, если проигнорировать это требование демонстрирует пример печально известного линейного конгруэнтного генератора, используемого в 60-х г. XX ст. – алгоритм RANDU, закон рекурсии которого $x_{i+1} = 65539 x_i (\mathrm{mod}\, 2^{31})$. Генератор на его основе появился в пакете прикладных программ IBM 360 (на Фортране). В 1977 г. выяснили, что генератор не проходит спектральный тест (тройки его последовательных значений ложатся на 15 параллельных плоскостей). Действительно

$$x_{i+2} \equiv 65539 x_{i+1} (\mathrm{mod}\, 2^{31}) \equiv (2^{16}+3) x_{i+1} (\mathrm{mod}\, 2^{31}) \equiv \underset{\uparrow}{(2^{16}+3)^2 x_i} \equiv$$

$$x_{i+1} \equiv (2^{16}+3) x_i (\mathrm{mod}\, 2^{31})$$

$$\equiv (2^{32} + 6 \cdot 2^{16} + 9) x_i (\mathrm{mod}\, 2^{31}) \equiv (0 + 6 \cdot (2^{16}+3) - 9) x_i (\mathrm{mod}\, 2^{31}),$$

$$\Rightarrow \qquad x_{i+2} \equiv 6 x_{i+1} - 9 x_i (\mathrm{mod}\, 2^{31})$$ – *корреляция между членами*.

§17. ДРУГИЕ НАБОРЫ ТЕСТОВ ДЛЯ ОЦЕНКИ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

• Тесты Кнута (Стэнфорд), представленные по 2-м томе «Искусство программирования» (нет рекомендованных параметров тестирования);

• пакет DIEHARD, предложенный Джорджем Марсалья (Флорида), из 18 тестов для исследования последовательностей длиной до более 80 миллионов битов. Тесты жесткие: если последовательность прошла тесты DIEHARD, то обычно она проходит и тесты NIST.

Наиболее интересные «***обезьяньи тесты***» для проверки распределения символов последовательности на основе анализа отсутствующих в ней подпоследовательностей. В битовой строке данных выделяют подпоследовательности определенной длины и интерпретируют их как слова, подсчитывая число пересекающихся слов. Название дал автор, предложив считать генератор обезьяной, печатающей на машине с определенным набором символов. Первый из этих тестов – ***проверка потока битов***. Последовательность символов переводят в биты $b = b_1 b_2 ... b_{32n}$, формируют строку из 20-буквеных пересекающихся слов (одна буква – один бит).

Например, $5\ 3\ 0\ 6 \Rightarrow b = 101\ 011\ 000\ 110 \Rightarrow$ слова $\{101\}$, $\{010\}$, $\{101\}$, $\{011\}$, $\{110\}, \{100\}, \{000\}, \{001\}, \{011\}, \{110\} \Rightarrow$ отсутствует слово – $\{111\}$. Тест повторяют неоднократно, меняя начало отсчета. Если тестируемая последовательность близка к истинно случайной, то число ни разу не появившихся слов описывается нормальным законом.

Последовательности, сгенерированные LFSR, характеризуются длинным периодом и неплохими статистическими свойствами. Но их использование в криптографии затруднено их простым построением. Поэтому гамму усложняют разными способами.

Генераторы ключевых последовательностей характеризуются:

- периодом повторения последовательности;
- статистическими свойствами порождаемых последовательностей;
- криптографической стойкостью генератора ключевой последовательности.

Если используется потоковое шифрование, тогда минимальный период должен удовлетворять условию:

$$T_{\Omega} >> T_{max} * V * n, \qquad (1)$$

где

T_{Ω} - минимальный период повторения последовательности,

T_{max} - максимальное время непрерывной работы КСЗИ,

V - скорость передачи в канале связи,

t - разрядность последовательности на выходе генератора ключевой последовательности.