ЛЕКЦИЯ 16.
Методы генерации случайных чисел

16.1. Требования к случайным числовым последовательностям. Физические источники случайных чисел.

16.2. Генераторы псевдослучайных последовательностей.

16.3. Криптографически генерируемые псевдослучайные числа.

Алгоритмы защиты сети, предполагающие использование случайных чисел:

- **Схемы взаимной идентификации**, рассмотренные в лекции 6. Сценарии распределения ключей в процессе установления соединения используют *оказии* для того, чтобы исключить возможность атаки на основе воспроизведения сообщений. Использование случайных чисел для оказий не дает шанса оппоненту определить или угадать значение оказии.
- **Генерирование сеансовых ключей**, выполняемое либо центром распределения ключей, либо одним из участников соединения.
- **Генерирование ключей** для алгоритма RSA − шифрования с открытым ключом.

Требования к используемой последовательности случайных чисел:
случайность и непредсказуемость.

Критерии для проверки последовательности на случайность:

- **однородность распределения**: распределение чисел в последовательности должно быть однородным, т.е. частота появления в последовательности конкретного значения должна быть ***примерно одинаковой*** для всех значений;
- **независимость**: ни одно из значений последовательности не должно *логически выводиться* из других значений.

Физические источники случайных чисел:

- импульсные детекторы ионизирующего излучения,
- газоразрядные лампы,
- конденсаторы с утечкой тока и пр.

Генераторы псевдослучайных последовательностей

Основные требования

к криптографически стойким

генераторам псевдослучайных последовательностей (или ***гаммы*****:**

1. Период гаммы должен быть **достаточно большим** для шифрования сообщений различной длины.
2. Гамма должна быть **трудно предсказуемой**.
3. Генерирование гаммы не должно быть связано с **большими техническими** и организационными трудностями.

Формирование значений *дробной части* многочлена первой степени:

$$Y(n) = Ent\,(a \times n+b), \qquad a, b = const.$$

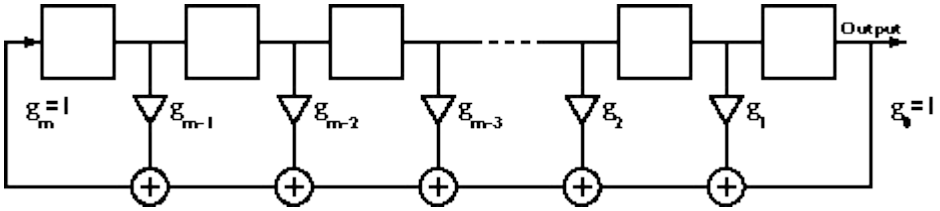
Способ Джона фон Неймана –

каждое последующее случайное число образуется **возведением предыдущего в квадрат** с последующим отбрасыванием цифр с обоих концов.

Генератор последовательности Фибоначчи

Последовательность Фибоначчи – в данной последовательности, за исключением первых двух ее членов, каждый последующий член равен сумме двух предыдущих:

{0,1,1,2,3,5,8,13,21,34 …}.



Генератор последовательности ***Фибоначчи.***

Квадратами обозначены ***разряды*** генератора,

треугольниками обозначено ***умножение на коэффициенты*** (на практике в зависимости от коэффициента там либо есть соединение с последующей логикой, либо его нет).

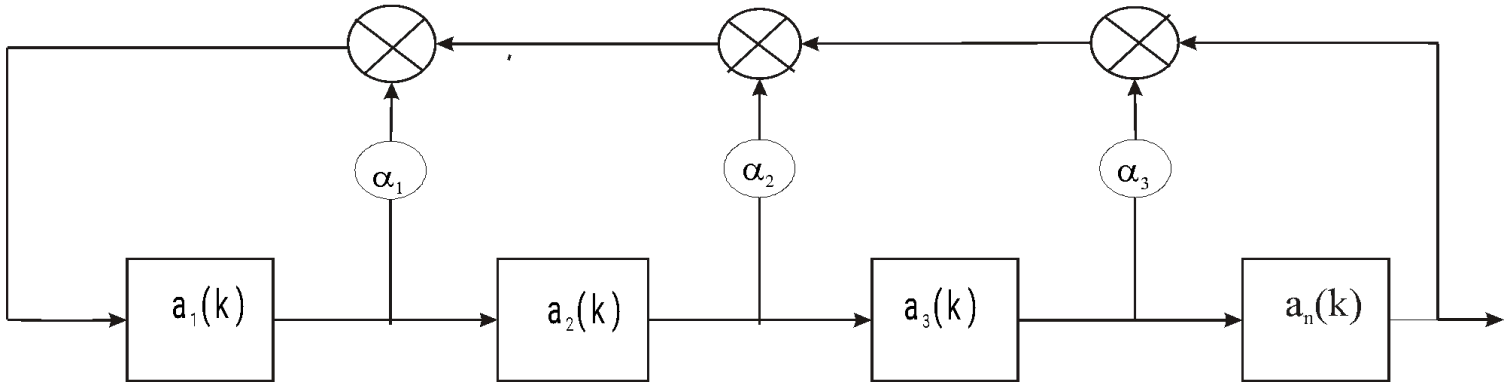
Плюсы в кружках − операция **XOR**: **0+0=0, 0+1=1, 1+1=0**

Рекуррентные двоичные последовательности

База для построения генератора псевдослучайных последовательностей - **регистр**.

При этом необходимо выполнение следующих условий:

- должно быть задано правило подключения сумматоров (α_0, α_2, .., α_N);
- α_0 и α_N всегда равны 1 (поэтому на схеме их можно не указывать);
- из всех α_i , $i \in \{1,2,..,N-1\}$, хотя бы одно должно иметь значение ‘1’.



Фильтр ***Хаффмана***

Циклические свойства генератора последовательности определяется т.н. ***характеристическим полиномом***:

$$\varphi(x) = \sum_{k=0}^{N} \alpha_k x^k$$

где $\alpha_0=\alpha_N=1$, $\alpha_j \in \{0,1\}$, j=1,2,..N-1.

Период последовательности будет **максимальным** в том случае, когда многочлен $\phi(x)$ удовлетворяет условиям **примитивности** и **неприводимости**.

Метод линейного сравнения *(конгруэнтный способ)*

Алгоритм имеет **четыре** параметра:

m	модуль сравнения	$m > 0$,
a	множитель	$0 \le a < m$,
c	приращение	$0 \le c < m$,
X_0	начальное или порождающее число	$0 \le X_0 < m$.

Последовательность случайных чисел $\{X\}$ получается с помощью итераций:

$$X_{n+1} = (aX_n + c) \bmod m.$$

1. Если m, a, c и X_0 являются ***целыми***, то будет получена последовательность целых чисел из диапазона $0 \le X_n < m$.
2. При $c = 0$, m – простое, наибольший период составит $m/4$ при $a = 3+8j$ или $a = 5+8j$ и нечетном начальном числе.
3. Если c нечетно, а $a = 1+4j$, то период можно увеличить до числа $m = 2^n$

Используются значения $a = 69069$ и $a = 71365$.

Значение m выбирается равным или почти равным значению 2^{31} - максимально допустимому для компьютера неотрицательному целому числу.

При реализации псевдослучайных последовательностей в ***компьютере*** предлагаются **три критерия** *качества* генератора двоичных псевдослучайных чисел:

- Генерирующая функция должна быть функцией **полного периода**, т.е. функция должна порождать все числа от **0** до **m** прежде, чем числа начнут повторяться.
- Генерируемая последовательность должна вести себя **как случайная**.
- Генерирующая функция должна эффективно реализовываться в рамках **32-битовой арифметики**.

В 32-битовой арифметике удобным простым значением ***m*** является значение $2^{31} - 1$.

В этом случае генерирующая функция принимает вид

$$X_{n+1} = (aX_n) \bmod (2^{31} - 1).$$

Знания небольшой части последовательности уже достаточно для того, чтобы определить все параметры алгоритма.

Например, противник сможет определить значения X_0, X_1, X_2, и X_3.

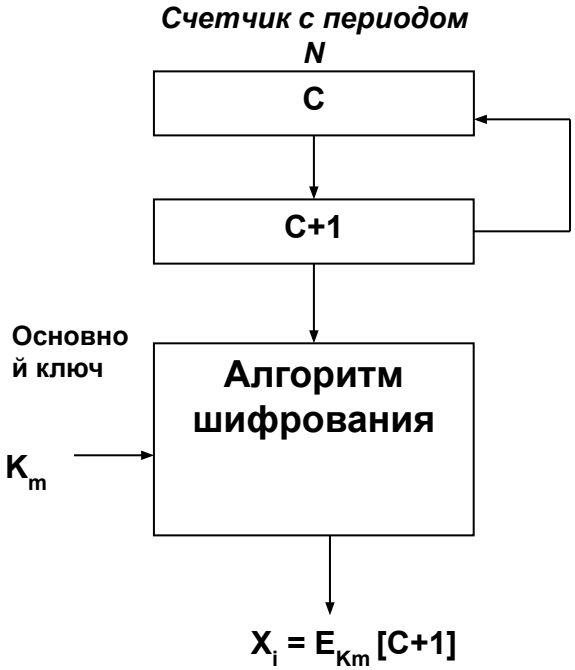
Тогда

$$X_1 = (aX_0 + c) \bmod m,$$
$$X_2 = (aX_1 + c) \bmod m,$$
$$X_3 = (aX_2 + c) \bmod m.$$

Эти уравнения могут быть решены относительно ***a, c*** и ***m***.

Криптографически генерируемые псевдослучайные числа

Циклическое шифрование



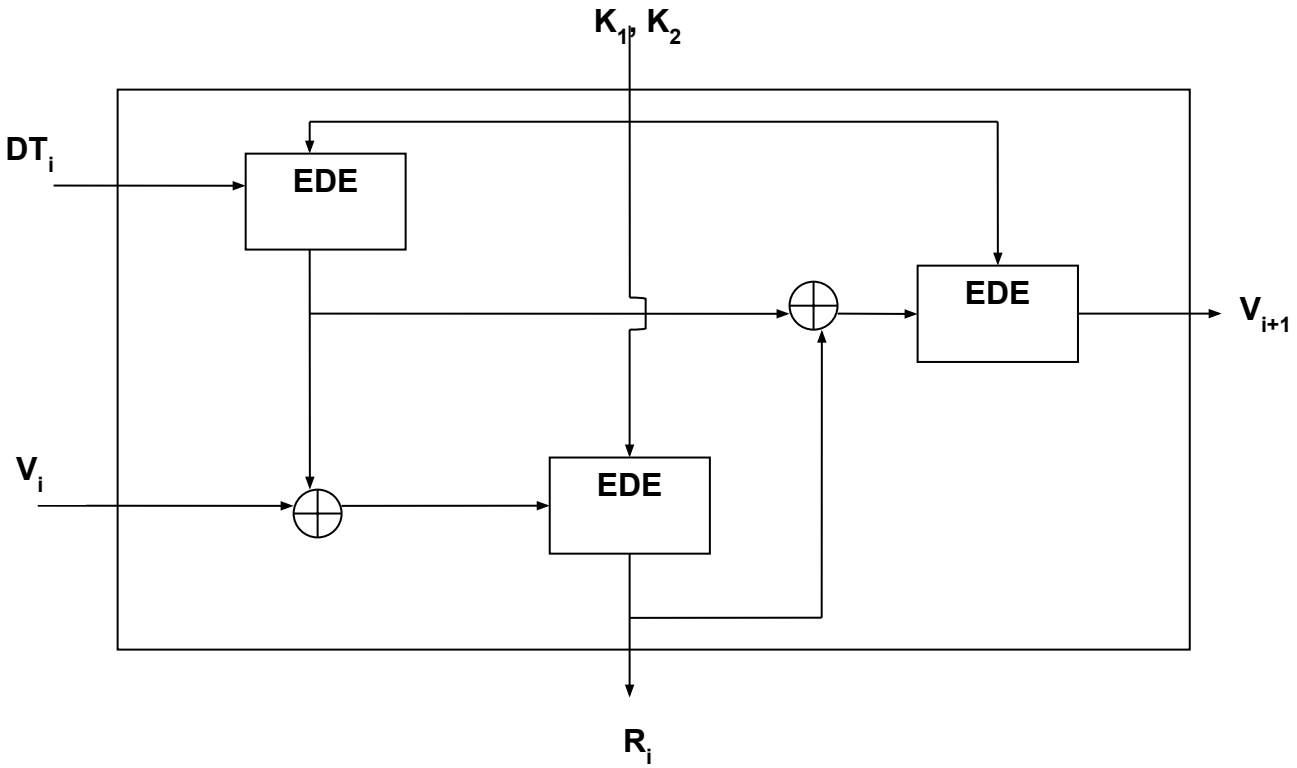
Генерирование псевдослучайных чисел **с использованием счетчика**

Чтобы сделать алгоритм еще более защищенным, вместо значений простого счетчика в качестве вводимых значений можно использовать выходные значения некоторого полнопериодического генератора псевдослучайных чисел.

Генератор псевдослучайных чисел ANSI X9.17.

Алгоритм имеет составляющие:

- **Ввод.** На вход генератора подается два псевдослучайных значения. Одно из них является 64-битовым представлением ***текущих даты и времени*** и меняется для каждого нового генерируемого числа. Другое представляет собой 64-битовое *начальное произвольное значение*, которое обновляется в процессе вычислений.
- **Ключи.** Генератор использует три модуля шифрования «тройного» DES. Каждый из этих трёх модулей использует одну и ту же пару 56-битовых ключей, которые должны сохраняться в секрете и использоваться *только для генерирования* псевдослучайных чисел.
- **Вывод.** Выводимыми значениями являются **64-битовое псевдослучайное число** и ***64-битовое начальное значение*.**



Генератор псевдослучайных чисел ANSI X9.17.

Обозначим:

DT_i : значение ***даты/времени*** в начале i -го шага генерирования;
V_i : начальное значение для i - го шага генерирования;
R_i : псевдослучайное число, получаемое в результате i - го шага генерирования;
K_1, K_2 : ключи DES, используемые на каждом шаге генерирования.
Тогда

$$R_i = EDE_{K1,K2}\,[V_i \oplus EDE_{K1,K2}\,[DT_i]],$$
$$V_{i+1} = EDE_{K1,K2}\,[R_i \oplus EDE_{K1,K2}\,[DT_i]],$$

где $EDE_{K1,K2}$ означает последовательность *«шифрования-дешифрования-шифрования»* с использованием алгоритма ***«тройного» DES с двумя ключами***.

Криптографическая надежность метода определяется факторами:

1. используются 112-битовый ключ и три блока шифрования **EDE**, в сумме дающие **девятикратное** шифрование ***DES***.

2. Схема управляется двумя вводимыми псевдослучайными значениями: значением **даты и времени** и **начальным** значением, производимым генератором, но *отличным* от производимого генератором псевдослучайного значения.

Генератор BBS
Блюма-Блюма-Шуба (Blum, Blum, Shub - BBS*)*

Процедура имеет вид:

1. Выбираются два **больших** простых числа ***p*** и ***q***, дающих при делении на 4 в остатке 3, т.е.

$$p \equiv q \equiv 3 \pmod 4.$$

Это означает, что (p mod 4) ≡ (q mod 4) ≡ 3.
Например, для простых чисел 7 и 11 как раз имеем 7 ≡ 11 ≡ 3 (mod 4).

2. Пусть $n = p \times q$.

Выбирается случайное число ***s***, взаимно простое с ***n*** —
ни ***p***, ни ***q*** *не являются делителями* ***s***.

3. Генератор ***BBS*** порождает последовательность битов B_i в соответствии с алгоритмом:

$$X_0 = s^2 \pmod n$$
$$\text{for } i = 1 \text{ to } \infty$$
$$X_i = (X_{i-1})^2 \pmod n$$
$$B_i = X_i \pmod 2$$

На каждой итерации выбирается *младший* бит.

Применение генератора BBS

для $n = 192649 = 383 \times 503$ и начального значения $s = 101355$.

i	X_i	B_i
0	20749	
1	143135	1
2	177671	1
3	97048	0
4	89992	0
5	174051	1
6	80649	1
7	45663	1
8	69442	0
9	186894	0
10	177046	0
11	137922	0
12	123175	1
13	8630	0
14	114386	0
15	14863	1
16	133015	1
17	106065	1
18	45870	0
19	137171	1
20	48060	0

Генератор ***BBS*** называют **криптографически защищенным генератором псевдослучайных битов.**

Этот генератор удовлетворяет *критерию следующего бита (next-bit test)*:

«Генератор псевдослучайных битов удовлетворяет критерию следующего бита, если не существует алгоритма с полиномиальной оценкой времени его выполнения, который по первым *k* битам выходной последовательности может предсказать ее (*k*+1)-й бит с вероятностью, существенно большей, чем ½».