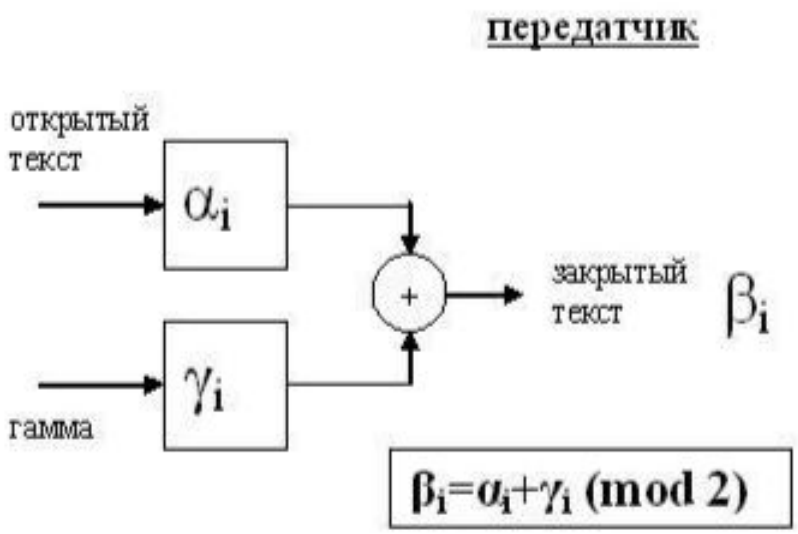
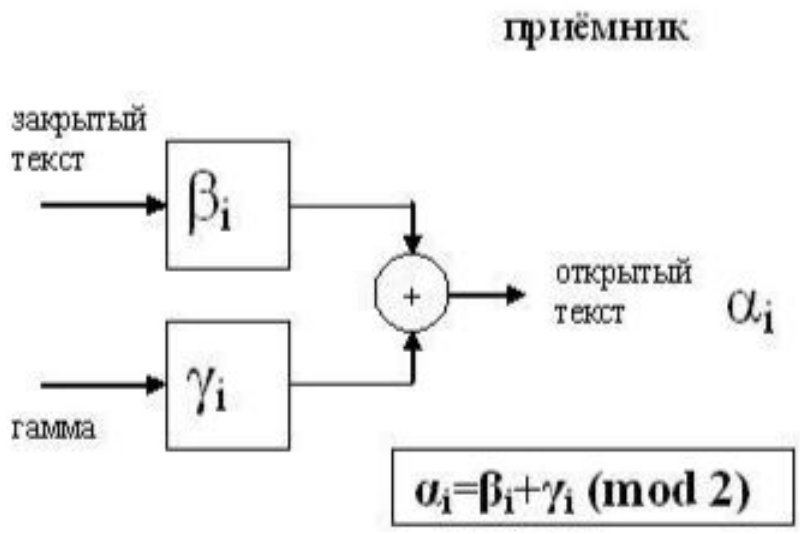
Лекция 2

Потоковые шифры

- Гаммирование
- Генераторы ПСЧ
- Потоковые шифры

- **Потоковый шифр** — это симметричный шифр, который выполняет преобразования над битами, реже байтами и словами Шифрование и дешифрование в потоковых шифрах осуществляется путем наложения на исходный или зашифрованный текст гаммы шифра





Процедуры шифрования и дешифрования выполняются побитово по следующим формулам

$C_i = P_i \ \text{XOR} \ K_i$,

$P_i = C_i \ \text{XOR} \ K_i$.

C_i – бит исходного текста, K_i – бит гаммы, P_i – бит защифрованного текста

- Иногда используется сложение и вычитание гаммы:

$C = (P + K) \bmod N$

$P = (C + N - K) \bmod N$

N – число символов в алфавите

- Гамма шифра - случайная или псевдослучайной последовательность
- Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, когда гамма шифра не содержит повторяющихся битовых последовательностей.
- Обычно наложение гаммы заключается в использовании операции "исключающее ИЛИ", называемое также сложением по модулю 2.
- Каждый бит зашифрованного текста зависит от ключа и номера шифруемого бита исходного текста.

Достоинства и недостатки

- В потоковых шифрах каждый символ открытого текста шифруется, передается и дешифруется независимо от других символов. В некоторых случаях символ открытого текста может шифроваться с учетом ограниченного числа предшествующих ему символов.

- Важным достоинством поточного шифрования является высокая скорость преобразования данных, соизмеримая со скоростью поступления открытого текста, что обеспечивает шифрование и расшифрование передаваемой информации больших объемов практически в реальном масштабе времени.

- *Сфера применения потоковых шифров* - военные, сетевые, телефонные и другие системы, где необходимо преобразование речевой информации в цифровую форму и надежное шифрование данных. Причина популярности – высокое быстродействие, простота реализации и конструирования генераторов, надежность шифрования, отсутствие ошибок в потоковом шифре.

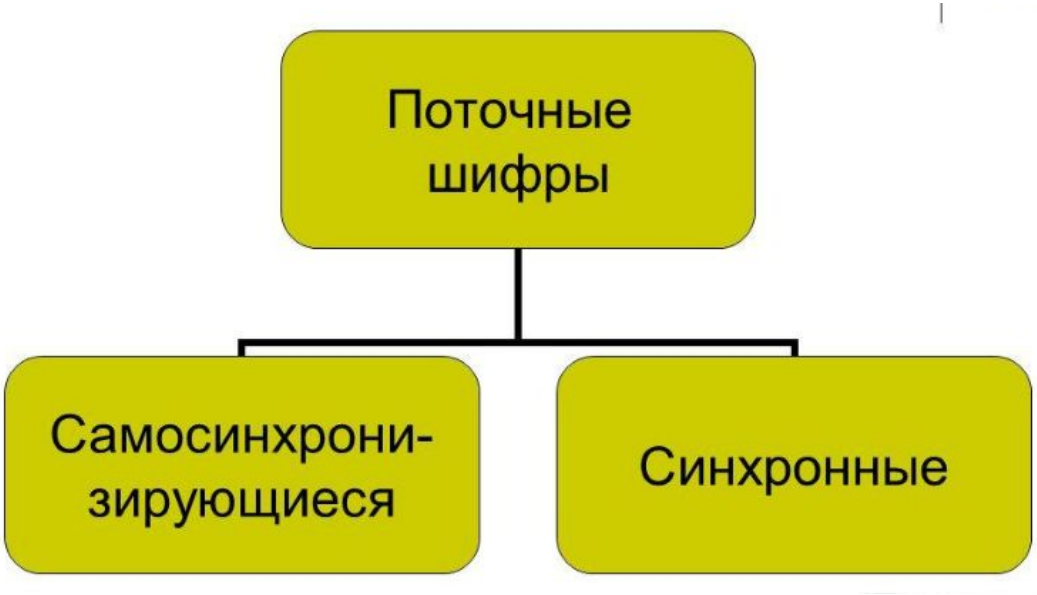
- Классический пример потоковых шифров – шифр Вернама, или одноразовый блокнот. Если для гаммы последовательность битов выбирается случайно и длина гаммы равна длине сообщения, то взломать шифр невозможно. Но у данного режима шифрования есть и отрицательная особенность – проблемы с передачей и хранением ключей, ведь ключи, сравнимые по длине с передаваемыми сообщениями, трудно использовать на практике.

- Поэтому основная идея современных потоковых шифров – реализовать концепцию одноразового блокнота, используя секретный ключ меньшей длины, из которого для гаммы генерируется псевдослучайная числовая последовательность, похожая на случайную.

- Гамму получают с помощью детерминированного генератора псевдослучайных чисел. Последовательность гаммы зависит только от **параметров инициализации**. Запущенный дважды с одними и теми же параметрами инициализации, генератор должен выдать одинаковые последовательности.
- Последовательность, называемая гаммой или потоком ключей, в этом случае не является ключом. Ключом являются параметры инициализации генератора ключевой последовательности.

Схема потокового шифра

- Стойкость системы целиком зависит от внутренней структуры генератора ключевой последовательности.



Синхронные потоковые шифры

- *Синхронные поточные шифры (СПШ)* — шифры, в которых поток ключей генерируется независимо от открытого текста и шифротекста и зависит только от исходного секретного ключа шифра $k_i = f(K)$
- При шифровании генератор потока ключей выдаёт биты потока ключей, которые идентичны битам потока ключей при дешифровании. Потеря знака шифротекста приведёт к нарушению синхронизации между этими двумя генераторами и невозможности расшифрования оставшейся части сообщения. Очевидно, что в этой ситуации отправитель и получатель должны повторно синхронизоваться для продолжения работы.
- Обычно синхронизация производится вставкой в передаваемое сообщение специальных маркеров. В результате этого пропущенный при передаче знак приводит к неверному расшифрованию лишь до тех пор, пока не будет принят один из маркеров.

Плюсы СПШ:

отсутствие эффекта распространения ошибок (только искажённый бит будет расшифрован неверно);

предохраняют от любых вставок и удалений шифротекста, так как они приведут к потере синхронизации и будут обнаружены.

Минусы СПШ:

уязвимы к изменению отдельных бит шифрованного текста. Если злоумышленнику известен открытый текст, он может изменить эти биты так, чтобы они расшифровывались, как ему надо.

Самосинхронизирующиеся поточные шифры

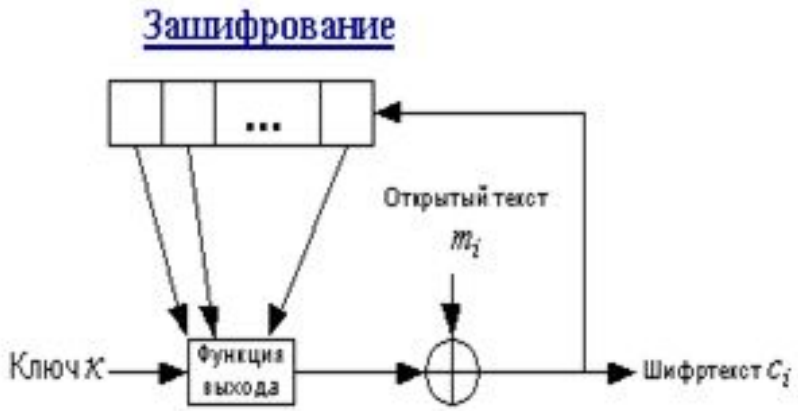
- *Самосинхронизирующиеся поточные шифры (асинхронные поточные шифры (АПШ))* – шифры, в которых поток ключей создаётся функцией ключа и фиксированного числа знаков шифротекста.

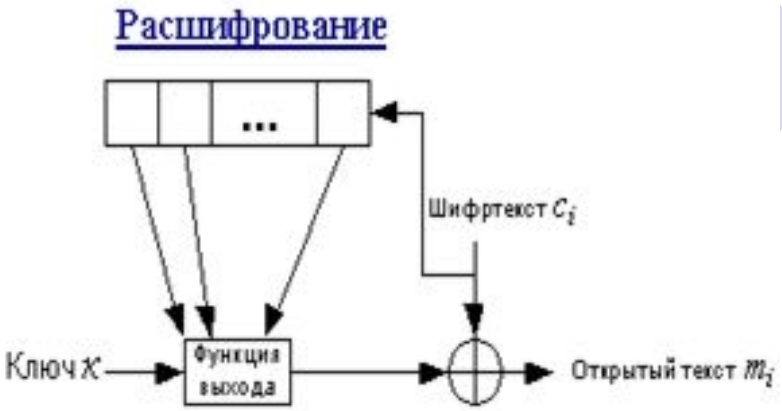
$$k_i = f(K, y_1, y_{2,\ldots,} y_N)$$

В этом случае на стороне получателя генератор начнет синхронно работать с передающей стороной после получения N битов

- Расшифрующий генератор потока ключей, приняв N битов, автоматически синхронизируется с шифрующим генератором.

- Реализация этого режима происходит следующим образом: каждое сообщение начинается случайным заголовком длиной N битов; заголовок шифруется, передаётся и расшифровывается; расшифровка является неправильной, зато после этих N бит оба генератора будут синхронизированы
- Недостаток этих потоковых шифров – распространение ошибок, так как искажение одного бита в процессе передачи шифротекста приведет к искажению N битов гаммы.





Плюсы АПШ:

Размешивание статистики открытого текста. Так как каждый знак открытого текста влияет на следующий шифротекст, статистические свойства открытого текста распространяются на весь шифротекст. Следовательно, АПШ может быть более устойчивым к атакам на основе избыточности открытого текста, чем СПШ.

Минусы АПШ:

распространение ошибки (каждому неправильному биту шифротекста соответствуют N ошибок в открытом тексте);

чувствительны к вскрытию повторной передачей.

Виды гаммирования

- Побитовое шифрование потока данных.
- 2. Побитовое шифрование потока данных с обратной связью (ОС) по шифртексту.
- 3. Побитовое шифрование потока данных с ОС по исходному тексту.
- 4. Побитовое шифрование потока данных с ОС по шифртексту и по исходному тексту.

Генераторы ПСЧ

- Поскольку потоковый шифр максимально должен имитировать одноразовый блокнот, то шифрующая гамма должна по своим свойствам походить на истинно случайную числовую последовательность. Устройство, с помощью которого реализуют истинно случайную последовательность, называют ее генератором, а члены последовательности – случайными числами.
- На сегодняшний день создано огромное количество генераторов псевдослучайных чисел. Все они являются периодическими. Но их периоды могут значительно превышать размеры данных, которые будут когда-либо зашифрованы такими генераторами, либо возможности компьютеров сгенерировать последовательность такой длины. Период последовательности в 2^{64} считается достаточным для использования в самых серьезных криптографических приложениях.
- Стойкость гаммирования однозначно определяется длиной периода гаммы

1. физические генераторы случайных чисел – устройства, генерирующие последовательность случайных чисел на основе измеряемых параметров определенных физических процессов

- на основе теплового шума (обусловлен тепловым движением носителей заряда в проводнике),
- интервалы времени между последовательными нажатиями на клавиатуру;
- счетчик тактов процессора;
- объем свободной памяти

2. табличные генераторы – таблицы случайных чисел, полученные экспериментально как выборки из равномерного распределения Недостатки: ограниченный объем таблиц, большой объем памяти компьютера при их хранении;

3. алгоритмические генераторы – программно-аппаратные средства, которые генерируют псевдослучайные последовательности, похожие по свойствам на случайные числа. Преимущества – быстродействие, компактность. Сгенерированные псевдослучайные последовательности периодичны, их члены почти независимы друг от друга и обычно подчиняются равномерному распределению вероятностей.

Псевдослучайный числовой генератор

- должен генерировать последовательности, статистические свойства которых не должны отличаться от свойств истинно случайной последовательности;
- быть непредсказуемым, т.е. невозможно предсказать значение каких-либо членов последовательности, зная какие либо другие ее члены:
- быть невоспроизводимым, т.е. разные начальные состояния генератора должны порождать разные числовые последовательности.
- Период выходных последовательностей криптографически безопасного генератора должен быть очень большим. Надо соизмерять длину периода гаммы и шифруемого сообщения. К примеру, если период гаммы 2^{32} , то гамма начнет повторяться после ~ 8,5 минут шифрования при скорости 1МВ/с).

Если генератор криптографически безопасный, то три нижеперечисленные задачи для криптоаналитика оказываются вычислительно неразрешимыми:

- определение следующего члена последовательности на основе ее известных членов (атака из прошлого);
- по известным членам последовательности невозможно восстановить предыдущие члены (атака из будущего);
- нахождение ключа по известным фрагментам гаммы конечной длины.

Характеристики ГПСЧ

- **Длиною периода гаммы** называется минимальное количество символов, после которого последовательность начинает повторяться.
- **Случайность распределения символов** по периоду означает отсутствие закономерностей между появлением различных символов в пределах периода.
- **Предсказуемость** означает, что для восстановления всей гаммы необходимо относительно небольшое количество символов гаммы

- Если гамма имеет период 2^{256} бит, но при этом для восстановления всей гаммы требуется только 2000 символов открытого текста, то такой алгоритм не может быть использован в криптографических целях. И если, зная 2000 символов невозможно вскрыть всю последовательность, но при этом период составляет всего 4000 символов, то такой алгоритм также не может быть использован в криптографических целях. Если гамма имеет достаточный период и непредсказуема, но при этом 75 % битов равно «1»”, то злоумышленник может восстановить ~50 % бит, не взламывая генератор. Гамма должна иметь большой период, достаточный для шифрования всех сообщений, в течение всего срока службы, быть статистически случайной и непредсказуемой

- **линейный конгруэнтный генератор псевдослучайных чисел**

$X_{i+1} = (a^* X_i + b) \bmod m$,

где a, b и m – некоторые коэффициенты.

Период линейной конгруэнтной последовательности будет максимальным и равен m, если: НОД (b;m) =1; число a -1 делится нацело на каждый простой делитель p модуля m; число a -1 делится нацело на 4, если модуль m кратный 4. При этих условиях получаемая последовательность называется линейным генератором максимального периода.

Например, при b = 0; a = 75; m= 212 147 483 647 получим

так называемый генератор Парка – Миллера $X_{i+1} = 75 X_i \bmod 2^{31} -1$ с максимальным периодом $2^{31} -1$

Два линейных конгруэнтных генератора могут быть объединены. Генерируется две последовательности с различными константами и из первой вычитается вторая. В этом случае для модулей порядка 2^{31}, период объединенной последовательности может достигать 2^{60}.

- квадратичные генераторы

$X_{i+1} = (a * X_i^2 + b * X_i + c) \bmod m$

кубические генераторы

$X_{i+1} = (a * X_i^3 + b * X_i^2 + c * X_i + d) \bmod m.$

- Криптографически стойким ГПСЧ является **алгоритм Блюм - Блюм - Шуба** (BBS).

$X_{i+1} = X_i^2 \bmod m$,

где m = p * q – является произведением двух больших простых p и q, сравнимых с 3 по модулю 4.

Пример с использованием двух малых простых чисел.

p = 7 (7 mod 4 = 3) и q = 19 (19 mod 4 = 3).

m = 7 * 19 = 133.

$X_0 = 53$.

Метод Фибоначчи с запаздываниями

$X_i = X_{i-a} — X_{i-b}$,

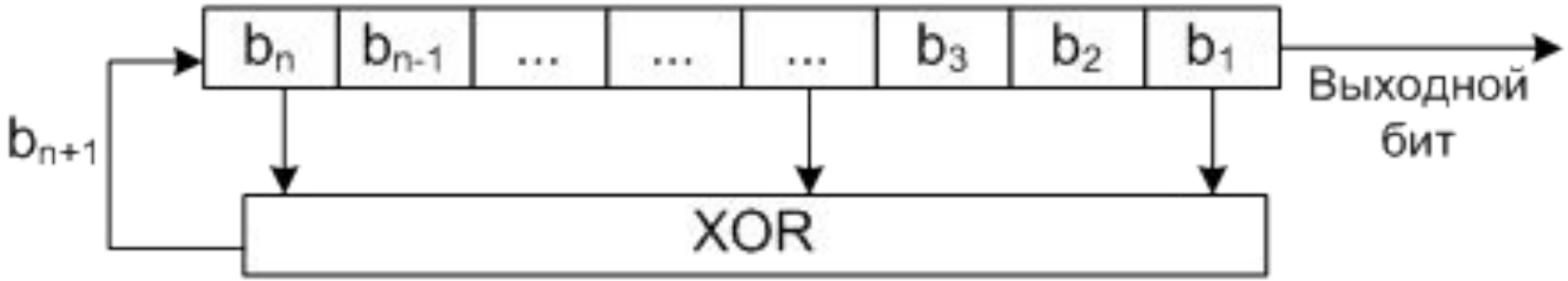
где переменная состояния *X* — беззнаковое целое. Величины запаздываний *a* и *b* берутся не какие угодно, а строго определенные, для достижения максимального качества рекомендуются пары (17,5), (55,24) или (97,33). Чем больше запаздывание, тем больше период и лучше спектральные свойства последовательности.

Сдвиговые регистры

- Сдвиговые регистры с обратной связью могут применяться для получения потока псевдослучайных *бит*. *Сдвиговый регистр* с обратной связью состоит из двух частей: собственно n-битного сдвигового регистра и устройства обратной связи
- Регистр представляет собой последовательность бит.

Извлекать биты из сдвигового регистра можно только по одному. Если необходимо извлечь следующий *бит*, все биты *регистра сдвигаются* влево на 1 разряд. в старший записывается выход функции обратной связи, а младший становится элементом ключевой последовательности либо проходит дополнительные преобразования.

Если функция обратной связи представляет собой сложение по модулю 2 некоторых битов регистра, то обратная связь называется линейной. Перечень битов, участвующих в сложении называется отводной последовательностью, или отводами



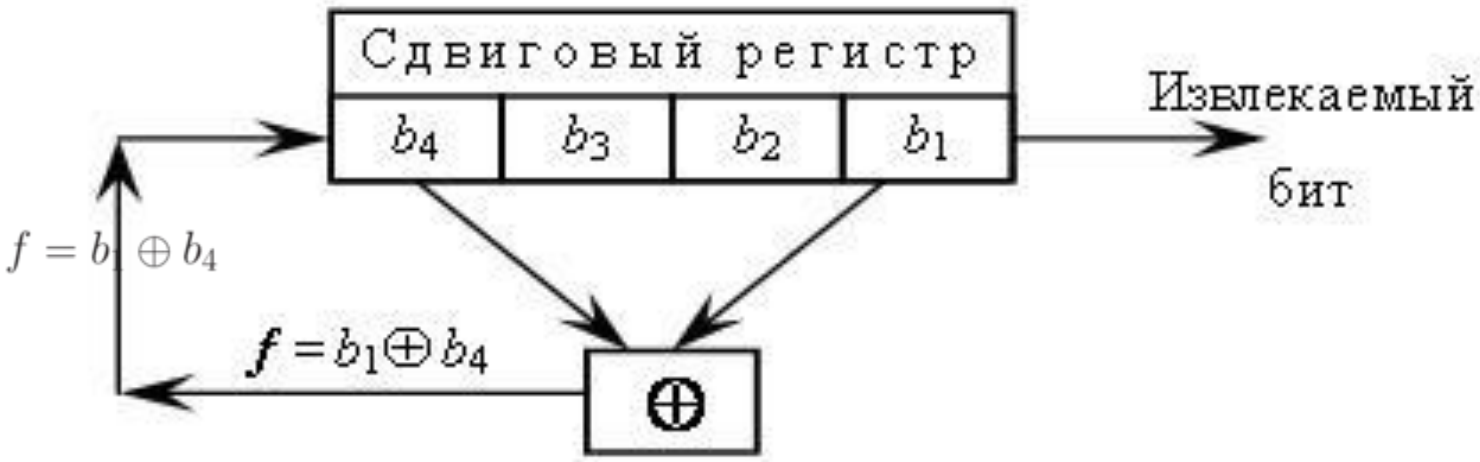
- Через некоторое количество тактов работы регистра последовательность битов начнет повторяться.
- *Длина* получаемой последовательности до начала ее повторения называется *периодом* сдвигового регистра.
- Простейшим видом сдвигового регистра с обратной связью является *линейный сдвиговый регистр с обратной связью* (*linearfeedback shift register – LFSR*) *или* РСЛОС. *Обратная связь* в этом устройстве реализуется просто как сумма по модулю 2 всех (или некоторых) битов регистра. Биты, которые участвуют в обратной связи, образуют *отводную последовательность*. Линейные сдвиговые регистры с обратной связью или их модификации часто применяется в криптографии.

- Если длина равна n битам, то регистр называют n-битовым сдвиговым регистром.
- Максимальное возможное число состояний сдвигового регистра равно 2^n–1
- Ключом РСЛОС является начальное состояние регистра и отводная последовательность. Так как не все последовательности позволяют генерировать последовательности максимальной длины, то перед использованием они должны проверяться. Для того чтобы проверить, является ли РСЛОС максимальным, необходимо из отводной последовательности и 1 образовать многочлен и проверить его на примитивность.
- Многочлен является примитивным, если он является неприводимым и является делителем $X^{2^n-1}+1$
- Например, примитивным является многочлен . X^4+X+1

- Обратные связи соответствуют коэффициентам полинома
- $C(x)= c_L x^L \oplus c_{L-1}x^{L-1} \oplus \ldots \oplus c_1 x+1$,

$c \in \{0,1\}$, $i = 1,..., L$, $c = 0,1$,

0 соответствует отсутствию связи, а 1 – ее наличию. Все операции ыполняются по модулю 2.



Номер состояния	Внутреннее состояние регистра b_4, b_3, b_2, b_1	Результат вычисления функции обратной связи	Извлекаемый бит (b_1)
0	1 0 1 1	0	1
1	0 1 0 1	1	1
2	1 0 1 0	1	0
3	1 1 0 1	0	1
4	0 1 1 0	0	0
5	0 0 1 1	1	1
6	1 0 0 1	0	1
7	0 1 0 0	0	0
8	0 0 1 0	0	0

- Для того чтобы генераторы на основе РСЛОС можно было использовать в криптографических приложениях, выходы нескольких независимо работающих регистров пропускают через некоторую нелинейную функцию

- В качестве нелинейной комбинирующей, выбирают булеву функцию равную сумме различных произведений выходов РСЛОС. Например:

$$f(x_1, x_2, x_3, x_4) = \bar{x}_2 x_3 x_4 \oplus x_1 \bar{x}_2 x_4 \oplus \bar{x}_1 x_2 x_4 \oplus x_1 x_3 \oplus x_2 x_3 \oplus \bar{x}_3 \oplus x_4$$

- Желательно, чтобы таблица истинности такой функции содержала примерно равное число нулей и единиц.

ПОСТУЛАТЫ ГОЛОМБА

Это необходимые, но не достаточные условия, позволяющие принять псевдослучайную последовательность за истинно случайную

На периоде $x_0, x_1, \ldots, x_{i+T-1}$ **должны выполняться:**

I ПОСТУЛАТ: |число «1» – число «0»| < 1;

II ПОСТУЛАТ: половина отрезков может иметь длину 1, четверть отрезков – длину 2; восьмая часть отрезков – длину 4 и т.д. На периоде должна быть одинаковое количество блоков и лакун

Типы отрезков длины s: блок $\underbrace{011\ldots10}_{s}$ **и лакуна** $\underbrace{100\ldots01}_{s}$

III ПОСТУЛАТ: функция автокорреляции должна быть

бути двузначной

ФУНКЦИЯ АВТОКОРРЕЛЯЦИИ ПОСЛЕДОВАТЕЛЬНОСТИ

$\{x_i \oplus x_{i+d}\};\ i = 0,1,2,...;\ d = 0,1,2,...,T-1$ **– результат «*XOR*-вания» исходной последовательности и ее копии, сдвину-той на величину** d ;

$n_1(d)$ и $n_0(d)$ **– число блоков 0 и 1 в последовательности** $\{x_i \oplus x_{i+d}\}$;

$$AC(d) = \frac{n_1(d) - n_0(d)}{T}$$ ***– функция автокорреляции последовательности*** .

ВЫХОДНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ LFSR УДОВЛЕТВОРЯЮТ ПОСТУЛАТАМ ГОЛОМБА

Тестирование псевдослучайных последовательностей

Псевдослучайные последовательности, порождаемые любым генератором для криптографических целей, подлежат обязательному тестированию.

Тестирование псевдослучайных последовательностей — совокупность методов определения меры близости заданной псевдослучайной последовательности к случайной. В качестве такой меры обычно выступает наличие равномерного распределения, большого периода, равной частоты появления одинаковых подстрок и т. п.

Существует несколько методов тестирования ПСП:

Ø Графический тест

Ø Статистический

Графические тесты

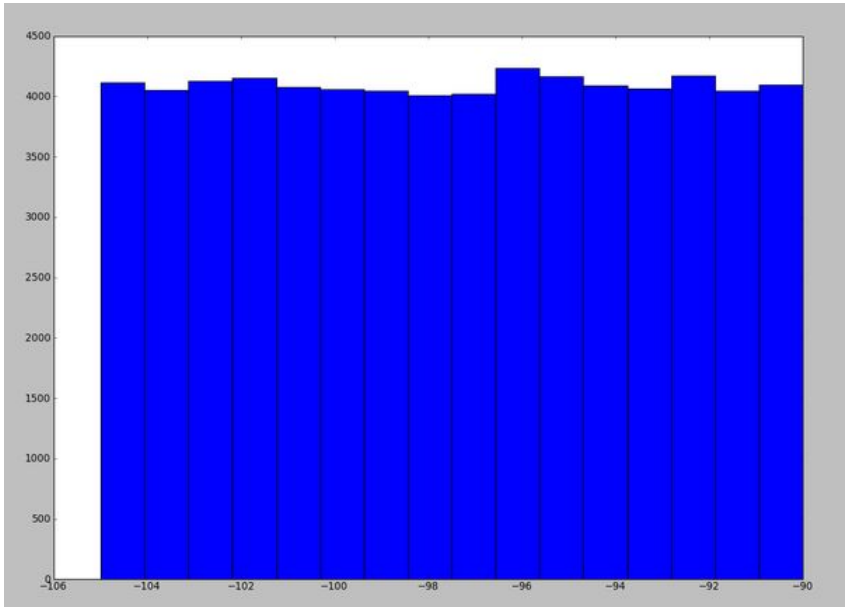
К этой категории относятся тесты, результаты которых отображаются в виде графиков, характеризующих свойства исследуемой последовательности. Среди них:

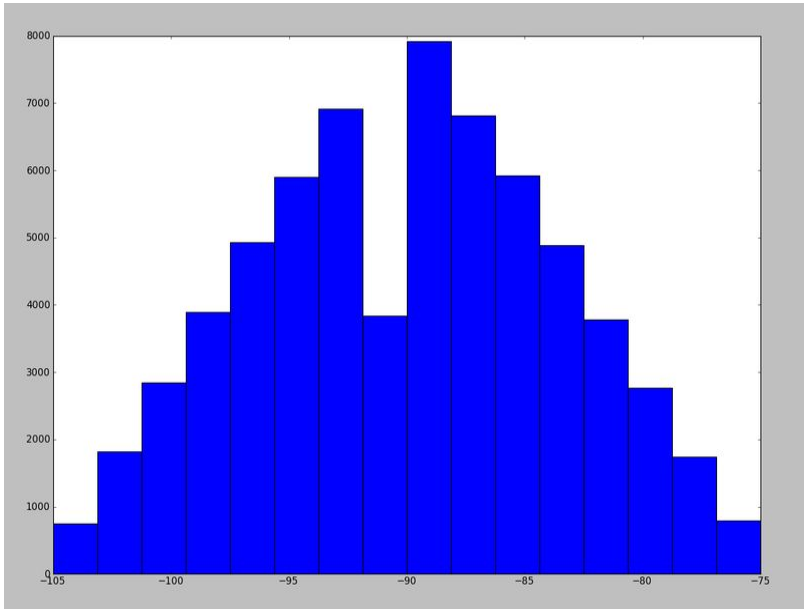
1. *гистограмма распределения элементов последовательности* (позволяет оценить равномерность распределения символов в последовательности и определить частоту повторения каждого символа);
2. *распределение на плоскости* (предназначено для определения зависимости между элементами последовательности);
3. *проверка серий* (позволяет определить равномерность отдельных символов в последовательности, а так же равномерность распределения серий из k бит);

Результаты графических тестов интерпретируются человеком, поэтому на их основе выводы могут быть неоднозначными.

ГРАФИЧЕСКИЕ ТЕСТЫ ДЛЯ СТАТИСТИЧЕСКОЙ ОЦЕНКИ КАЧЕСТВА ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

- ***Гистограмма распределения элементов последовательности***

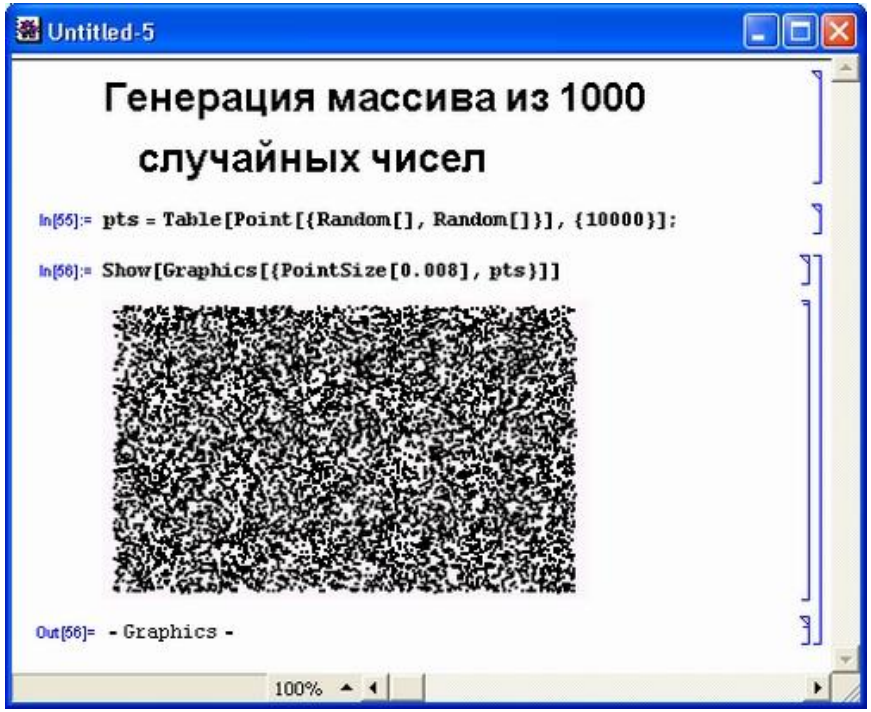


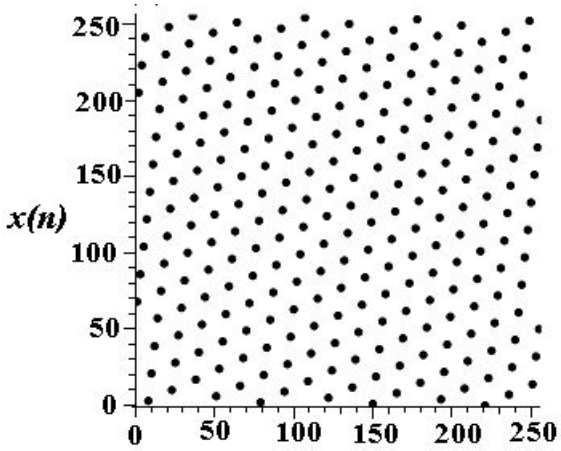


ГРАФИЧЕСКИЕ ТЕСТЫ ДЛЯ СТАТИСТИЧЕСКОЙ ОЦЕНКИ КАЧЕСТВА ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

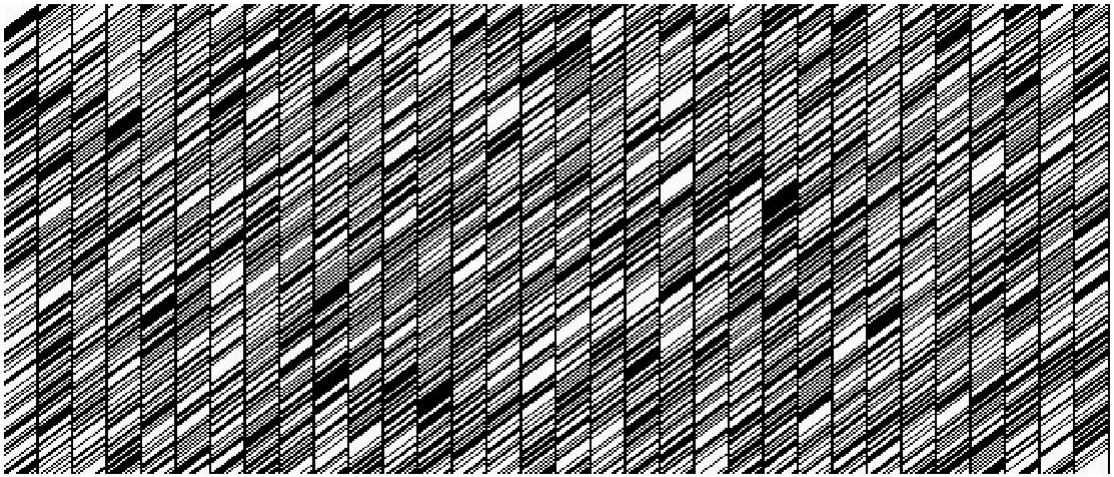
- ***Распределение на плоскости***

Поле размером $(2^R - 1) \times (2^R + 1)$ ***где*** R ***– разрядность чисел последовательности, строят точки с координатами*** $(x_i; x_{i+1})$

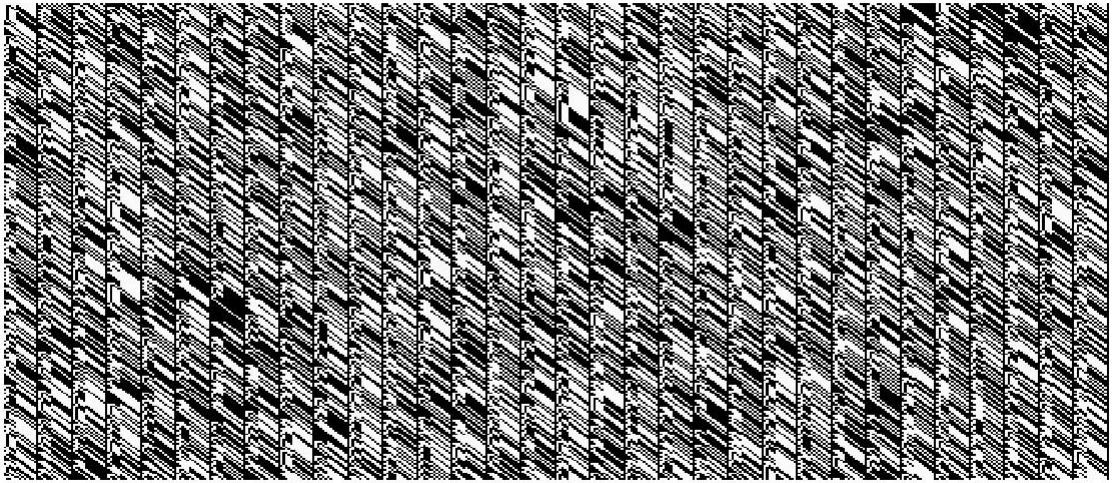




ная



генератора
Фибоначчи



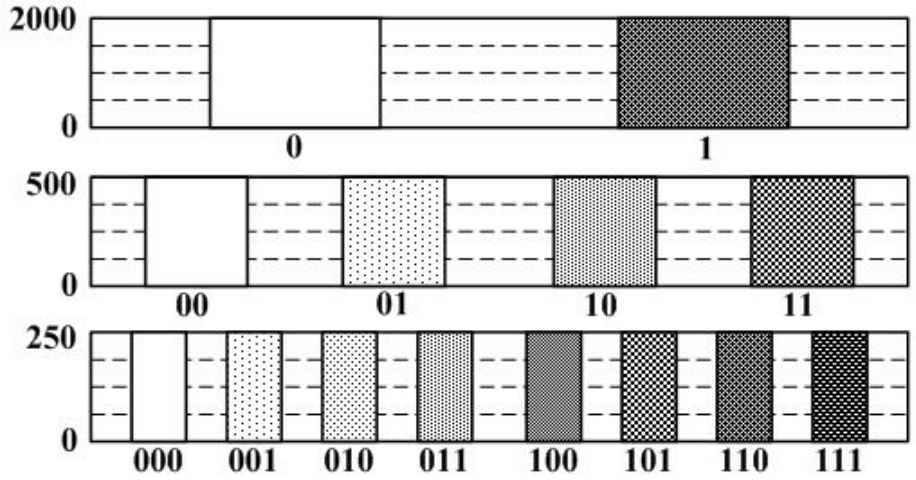
Выходная
последовательность
генератора
Галуа

ГРАФИЧЕСКИЕ ТЕСТЫ ДЛЯ СТАТИСТИЧЕСКОЙ ОЦЕНКИ КАЧЕСТВА ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

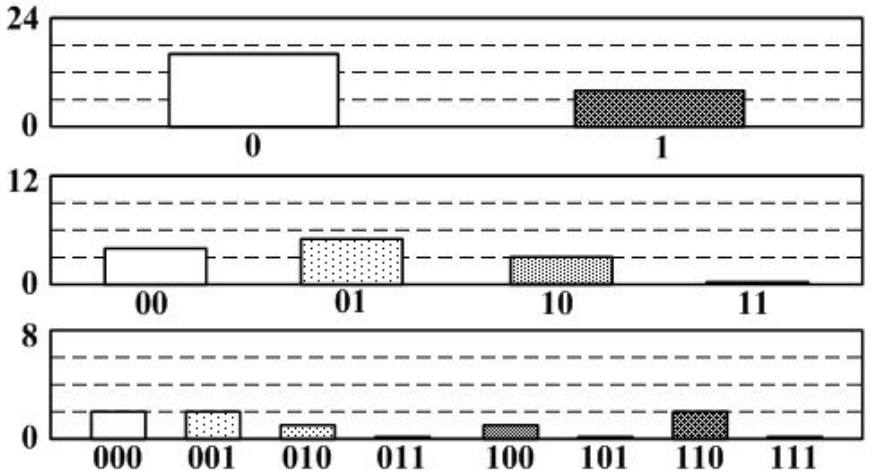
- ***Графическая проверка серий***

Цель – определение равномерности распределения символов. Анализ частоты появления 0 и 1, и разных s-грамм (без перекрытия). (в последовательности подсчитывают число 0, 1, биграмм (00, 01, 10, 11), триграмм (000, 001, 010, 011, 100, 101, 110, 111) и т.д.).

Тест пройден



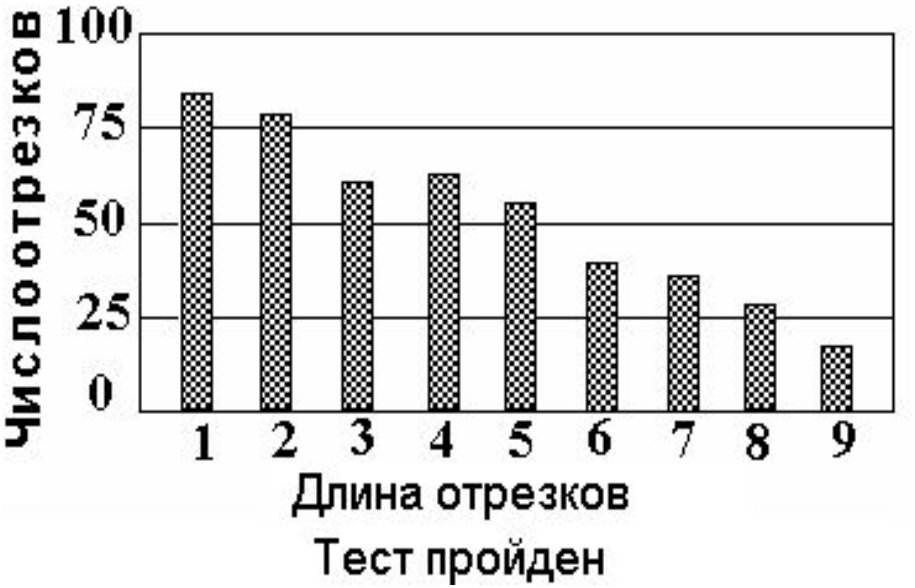
Тест не пройден



ГРАФИЧЕСКИЕ ТЕСТЫ ДЛЯ СТАТИСТИЧЕСКОЙ ОЦЕНКИ КАЧЕСТВА ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

- ***Графическая проверка монотонности***

оценки равномерности распределения символов сравнением длин отрезков невозростания и неубывания членов.



Тест пройден

Тест не пройден

Статистические тесты

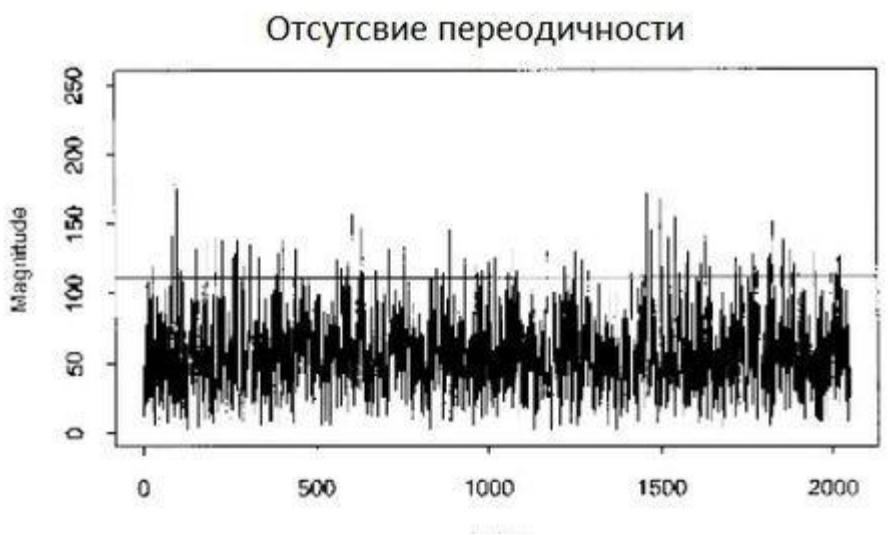
В отличие от графических тестов, статистические тесты выдают численную характеристику последовательности и позволяют однозначно сказать, пройден ли тест. Наиболее известные тесты:

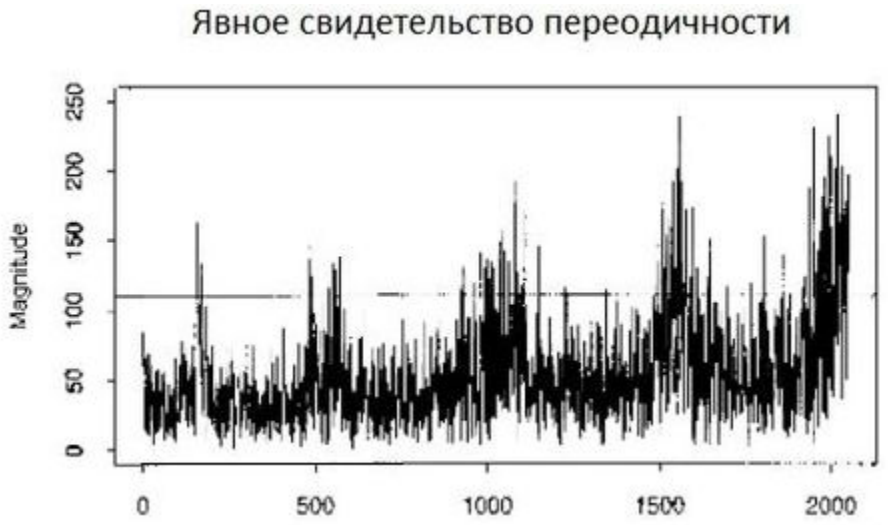
- Подборка статистических тестов Д. Кнута;
- DIEHARD;
- CRYPT-X;
- NIST STS;

- Пакет NIST STS включает в себя 16 статистических тестов, которые разработаны для проверки гипотезы о случайности двоичных последовательностей произвольной длины, порождаемых ГСЧ или ГПСЧ. Все тесты направлены на выявление различных дефектов случайности. Основным принципом тестирования является проверка нулевой гипотезы H_0, заключающейся в том, что тестируемая последовательность является случайной. Альтернативной гипотезой H_a является гипотеза о том, что тестируемая последовательность не случайна. По результатам применения каждого теста нулевая гипотеза либо принимается, либо отвергается. Решение о том, что будет ли заданная последовательность нулей и единиц случайной или нет принимается по совокупности результатов всех тестов.

- Частотный тест. Состоит в подсчете количества нулей и единиц в последовательности битов. Единиц и нулей должно быть примерно поровну.
- Тест на последовательность одинаковых битов. Ищутся ряды одинаковых битов, вида 000...0 или 111...1. Распределение частот, с которыми встречаются ряды, в зависимости от их длины, должно соответствовать такому распределению для истинно случайного сигнала.
- Автокорреляционный тест. Подсчитывается значение корреляции между копиями последовательности, сдвинутыми друг относительно друга. Тест позволяет найти повторяющиеся участки в последовательности.

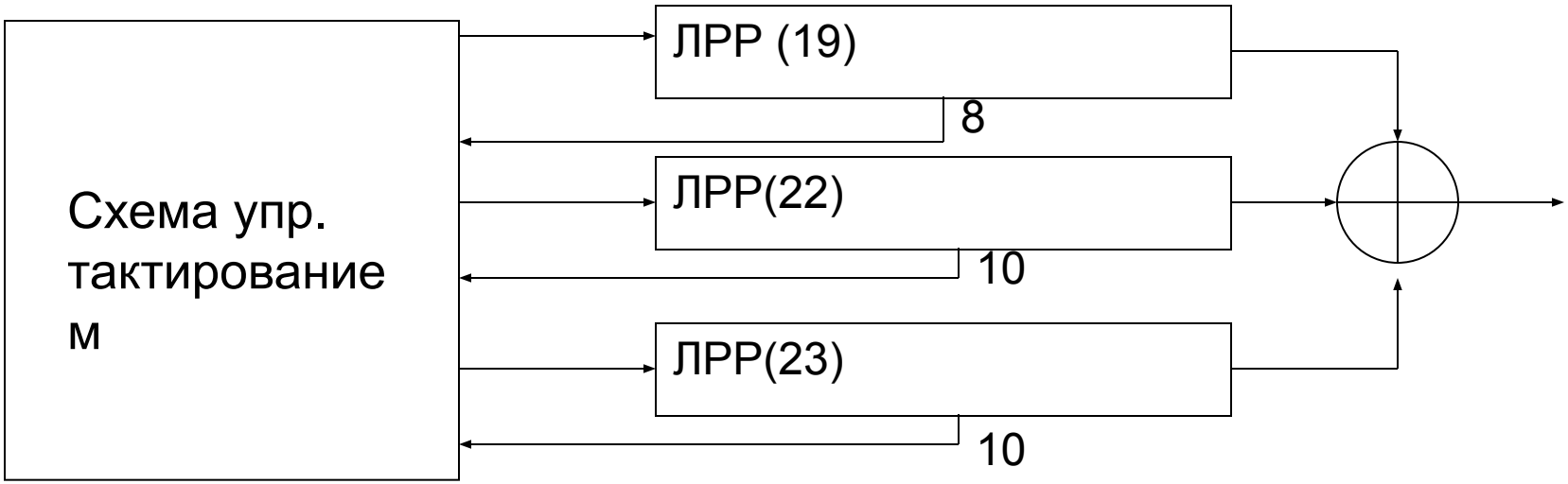
- ***Проверка рангов матриц*** оценивается равномерность распределения 0 и1, для чего из членов последовательности образуются матрицы и ищутся их ранги;
- ***Спектральный тест*** К исходной последовательности применяется дискретное преобразование Фурье. Полученный спектр не должен иметь значительных пиков, которые говорили бы о наличии периодических свойств последовательности

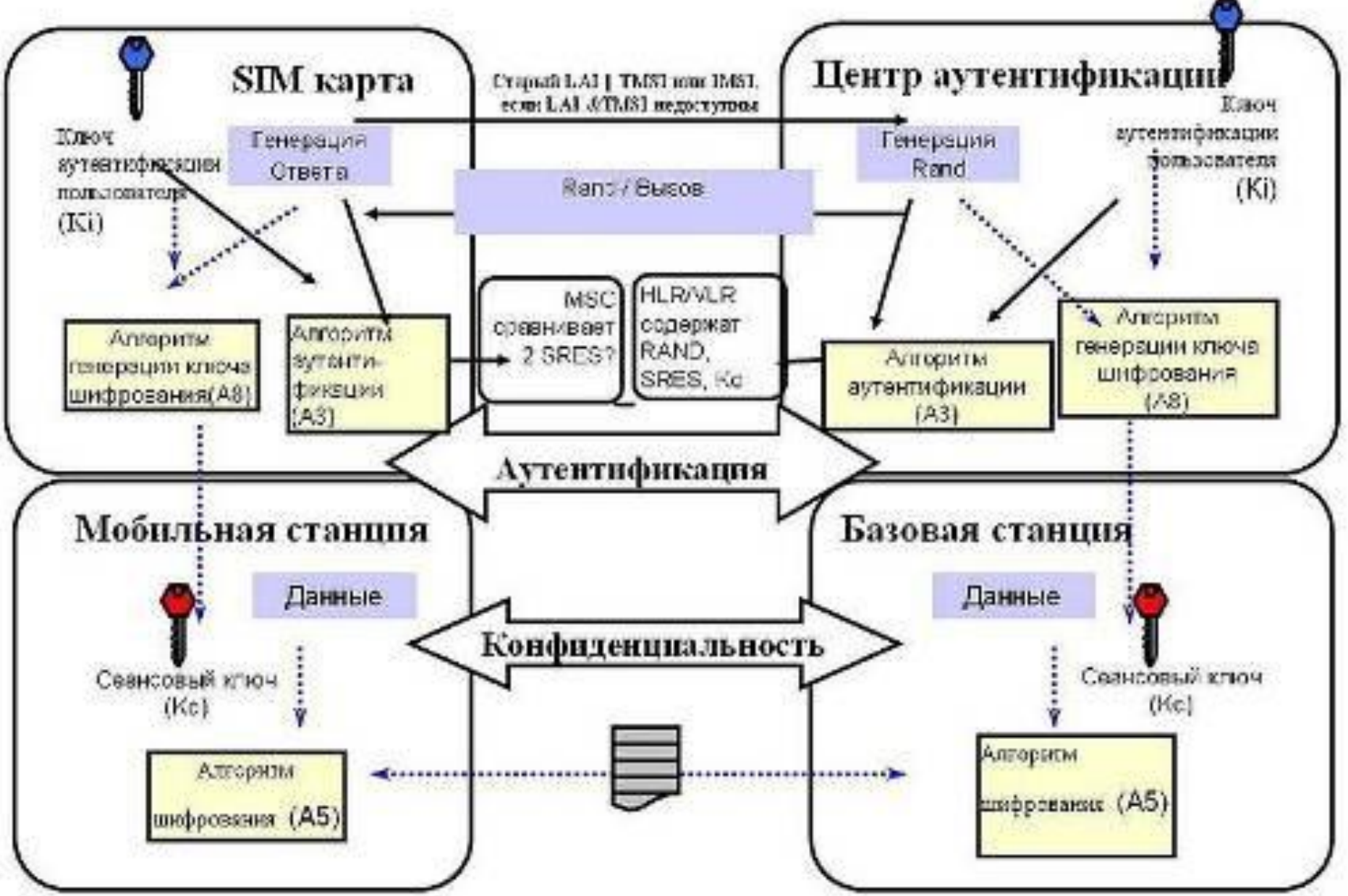




Примеры потоковых шифров

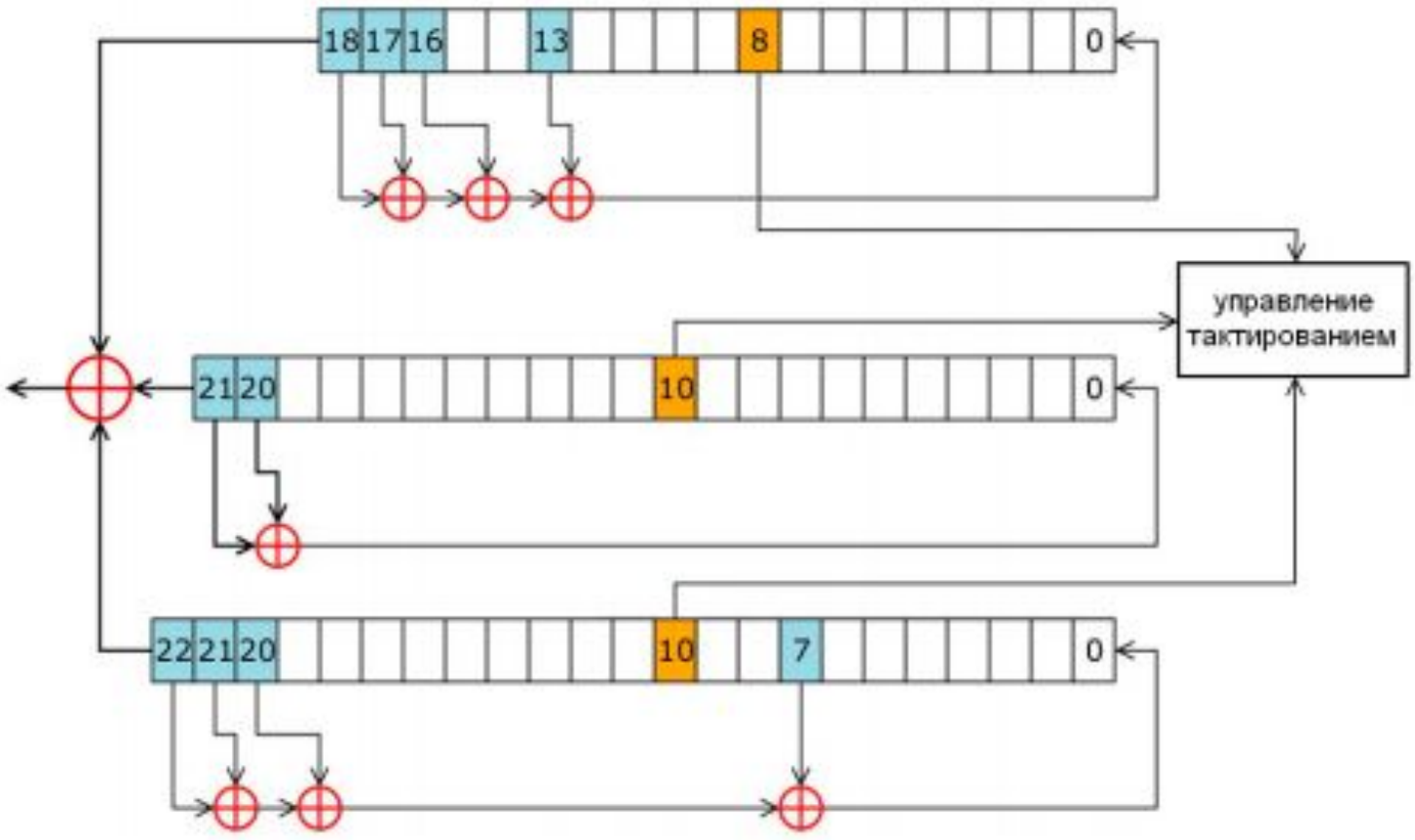
- A5 (шифрование в GSM)




SIM карта
Старый LAI [TMSI или IMSI,
если LAI /TMSI недоступны
Центр аутентификации
Ключ аутентификации пользователя (Ki)
Генерация Ответа
Rand / Ответ
Генерация Rand
Ключ аутентификации пользователя (Ki)
Алгоритм генерации ключа шифрования(A8)
Алгоритм аутентификации (A3)
MSC сравнивает 2 SRES?
HLR/VLR содержат RAND, SRES, Kc
Алгоритм аутентификации (A3)
Алгоритм генерации ключа шифрования (A8)
Аутентификация
Мобильная станция
Базовая станция
Данные
Данные
Конфиденциальность
Сеансовый ключ (Kc)
Сеансовый ключ (Kc)
Алгоритм шифрования (A5)
Алгоритм шифрования (A5)

Шифр A5.

используется в современной сотовой связи в Европе и США. Шифр A5 состоит из трёх регистров сдвига.



- Генератор A5 состоит из трёх регистров сдвига с обратной связью, причём разной длины. Предположительно, длина периода равна перемножению периодов каждого из трёх генераторов, причём здесь используется нетривиальная схема тактирования. Выделяется 3 бита, а затем вычисляется мажоритарная функция, и дальше сдвигаются те регистры, у которых данный бит равен значению мажоритарной функции (мажоритарная функция даёт на выходе 0, если в качестве её аргументов выступают нули; если это не так, то на выходе 1). То есть сдвигаются те регистры, у которых в данных ячейках совпадающие биты. Оказалось, что у такой сложной структуры реально в среднем число периодов составляет 2^{23} (не очень большой период для такой сложной схемы). Более того, к данному шрифту возникали претензии: по поводу генерации пароля, по поводу инициализации и т. д. Сложность атаки для данного шифра составляет 2^{40}. Это означает, что сейчас взломать данный шифр может любой персональный компьютер.

RC4

- RC – сокращение от Rivest's Cipher. Шифр не был запатентован, но находился в частной собственности и оставался коммерческой тайной. RC4 может использовать различные длины слов и различные длины ключей. Но кроме требований к лицензированию, он также подпадает под экспортные ограничения законодательства США, поэтому в России он не может быть законно использован с длиной ключа более 40 бит.
- Основным определяющим параметром является размер слова *n* (шифр работает не с битами, а со словами). В большинстве примеров это 8 бит, но на сегодняшний день может использоваться и 16. Количество ячеек внутреннего состояния равно 2^n, каждая по *n* бит. Необходимо, что бы все возможные слова были записаны в ячейках внутреннего состояния. На первом этапе инициализации ячейки (S-блоки) заполняются последовательно значениями от 0 до 2^n -1.
- For i = 0 to 2^n -1
- S[i] = i

Далее выполняется перемешивание блоков в соответствии с ключом.

$j = 0$

For $i = 0$ to $2^n - 1$

$j = (j + S[i] + Key[i \bmod l]) \bmod 2^n$

Перестановка (S[i], S[j])

Здесь Key – ключ (инициализирующая последовательность) длины l бит. Для длины ключа здесь нет никаких ограничений (кроме законодательных).

Устанавливаются значения внутренних переменных – индексов

$i = 0$

$j = 0$

После завершения этих подготовительных этапов может выполняться непосредственно шифрование.

Для каждого цикла шифрования устанавливаются новые значения индексов:

$i = (i + 1) \bmod 2^n$

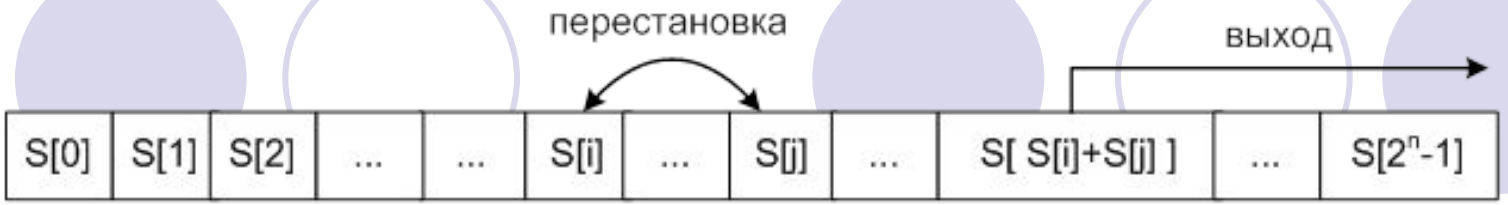
$j = (j + S[i]) \bmod 2^n$

Выполняется перестановка блоков с индексами i и j:

Перестановка (S[i], S[j])

Результатом является:

$K = S[(S[i] + S[j]) \bmod 2^n]$.



Если длина слова составляет 8 бит, то количество различных внутренних состояний составляет 256! $256^2 \approx 2^{1700}$. Для 16 бит – 2^{954069}. Перебор такого числа состояний даже для 8 битовых слов невозможен и наверное никогда не станет возможным. Так же ничего не известно об успешных криптографических атаках на этот алгоритм. В результате самым узким местом является непосредственно ключ. Его длина должна выбираться из соображений невозможности полного перебора. Длина ключа в 40 бит явно недостаточна для обеспечения безопасности.

Алгоритм шифрования.

1. Функция генерирует последовательность битов .
2. Затем последовательность битов K_i посредством операции (xor) объединяется с открытым текстом (m_i). В результате получается шифрограмма (c_i)