КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ
ИНСТИТУТ ФИЗИКИ
КАФЕДРА РАДИОФИЗИКИ

КАРПОВ А.В., ТУКТАРОВА И. Р., СМОЛЯКОВ А.Д.

ИМИТАЦИОННАЯ КОМПЬЮТЕРНАЯ МОДЕЛЬ КРИПТОГРАФИЧЕСКОЙ СИСТЕМЫ, ОСНОВАННАЯ НА ГЕНЕРАТОРАХ М-ПОСЛЕДОВАТЕЛЬНОСТИ

Учебно-методическое пособие

Казань - 2015

Принято на заседании кафедры радиофизики
Протокол № 12 от 25 июня 2015 года

Рецензент:
кандидат физико-математических наук,
доцент кафедры Информатики и информационно-управляющих систем
Казанского государственного энергетического университета **Р.А. Ишмуратов**

Карпов А.В., Туктарова И. Р., Смоляков А.Д.,
Имитационная компьютерная модель криптографической системы, основанная на генераторах М-последовательности / А.В. Карпов, И.Р. Туктарова, А.Д. Смоляков – Казань: Казан. ун-т, 2015. – 30 с.

Настоящее пособие включает в себя теоретическую часть и задания, выполняемые в лаборатории по курсу «Криптографические методы защиты информации». Целью представленной работы является получение практических навыков и умений анализа алгоритмов шифрования на примере анализа блочной криптографической системы при помощи тестов NIST и проведения демонстрационной атаки на алгоритм. В описании приводятся сведения, необходимые для выполнения практических заданий.

Учебно-методическое пособие рекомендуется студентам института физики К(П)ФУ, обучающимся по специальностям 090900.62 -«Защита информации» по дисциплине «Криптографические методы защиты информации» и 011800.68 – «Радиофизика» в магистратуре «Информационные процессы и системы» по дисциплине «Основы информационной безопасности».

ОГЛАВЛЕНИЕ

1.СИММЕТРИЧНЫЕ ПОТОЧНЫЕ ШИФРЫ

1.1 Краткая теоретическая справка

Составляющими любой системы шифрования (или криптосистемы) являются сообщение M, ключ K и шифрограмма C. Шифрование и расшифрование осуществляются различными методами, использующими определенными алгоритмы.

Существует два типа шифрования: симметричное и асимметричное.

В *симметричных криптосистемах* (системы с закрытым ключом) для шифрования и расшифрования используется один и тот же ключ. Алгоритм и ключ выбирается заранее и известен обеим сторонам. Сохранение ключа в секретности является важной задачей для установления и поддержки защищённого канала связи. В связи с этим, возникает проблема начальной передачи ключа (синхронизации ключей).

В *асимметричных системах* (с открытым ключом) используются два ключа — открытый и закрытый, математически связанные друг с другом. Открытый ключ передаётся по открытому (то есть незащищённому) каналу и используется для шифрования сообщения. Для расшифровки сообщения используется секретный ключ. Данная схема решает проблему симметричных схем, связанную с начальной передачей ключа другой стороне[1].

Подробнее рассмотрим симметричное шифрование. Оно включает в себябдве большие группы алгоритмов: блочные и поточные шифры.

Блочный шифр–это симметричный шифр, при котором сообщение разбивается на блоки и каждый блок шифруется.

Поточный шифр - это симметричный шифр, в котором каждый символ открытого текста преобразуется в символ шифрованного текста в зависимости от используемого ключа и от его расположения в потоке открытого текста.

Ключевую последовательность для поточного шифра вырабатывает генератор псевдослучайной битовой последовательности. Чаще всего используются

генераторы М-последовательности, основанные на регистрах сдвига. Регистры сдвига строятся по структуре примитивных неприводимых полиномов[2].

Шифрование осуществляется побитовым сложением по модулю два ключевой последовательности и двоичного сообщения⊕МК = С.

Расшифрование– при помощи обратной операции, где складываются шифрованная битовая последовательность и ключ: С ⊕К = М.

Надежность системы зависит только от свойств ключевой последовательности. Чем ближе битовый ключ по свойствам к случайно распределенной последовательности бит, тем меньше вероятность его раскрытия[3]. Оценить, насколько сгенерированный ключ обладает всеми необходимыми свойствами, может его проверка по различным параметрам и характеристикам, называемая тестированием последовательности.

Существует много простых и сложных тестов для битовых последовательностей различной величины. Наиболее полной и проработанной базой тестов является база тестов NIST.

1.2 Имитационная модель работы поточного шифра

Рассмотрим, как работает имитационная модель поточного шифра, основанная на использовании ключа шифрования, создаваемом генератором М-последовательности. Подробнее об имитационном моделировании можно прочитать в пособии [4], об особенностях алгоритмического создания криптосистемы на основе поточных шифров – в учебно-методическом пособии [5].

Блок-схема работы алгоритма поточного шифра приведена на рисунке 1.Пример работы алгоритма поточного шифра представлен на рисунке 2. Для генерации ключевой последовательности используем полином $p(x)=x^4+x^1+1$ (на рисунке представлены блоками 1, 2, 3 и 4).Псевдослучайная последовательность генерируется при помощи регистра сдвига, созданного на основе примитивного неприводимого полинома. Из нее формируется битовый ключ необходимой длины. Бинарный текст сообщения побитово складывается с ключом по модулю два, на выходе образуя зашифрованный текст. Расшифрование проис-

ходит сложением по модулю два зашифрованной битовой последовательности с той же ключевой последовательностью.

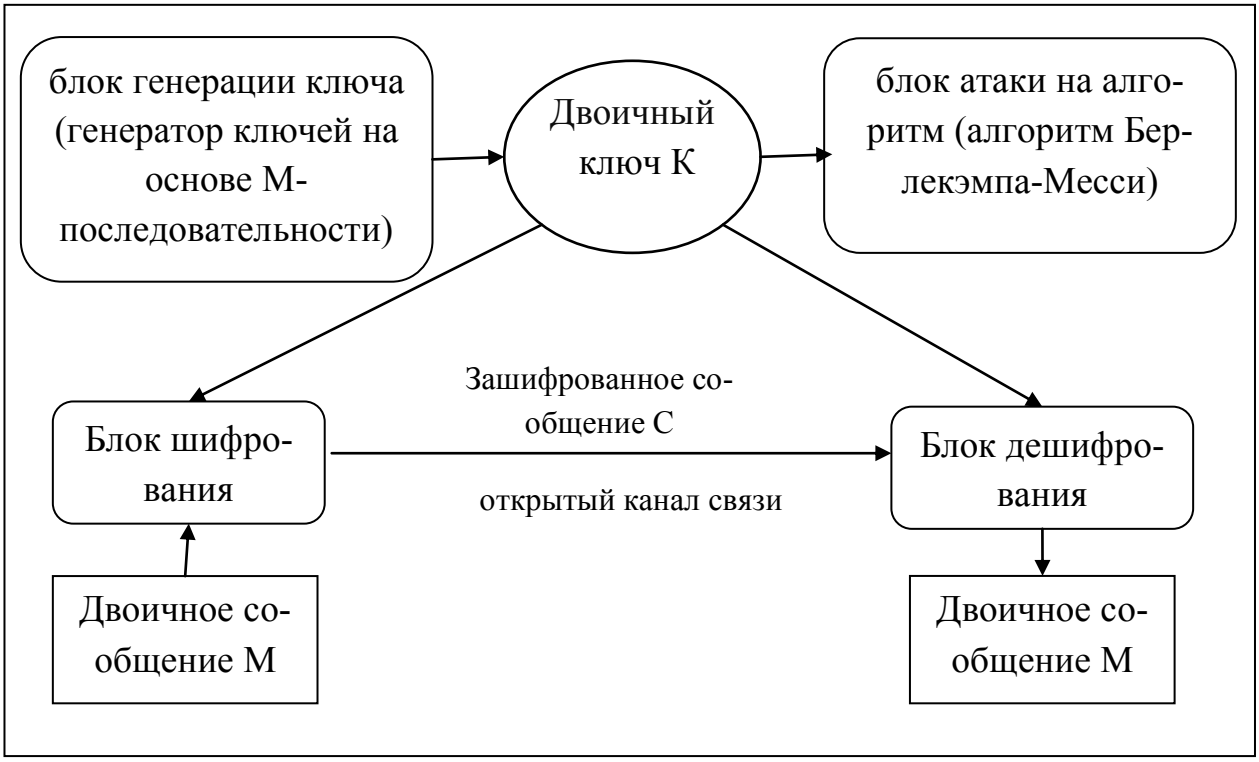

Рис. 1. Блок-схема работы алгоритма поточного шифра

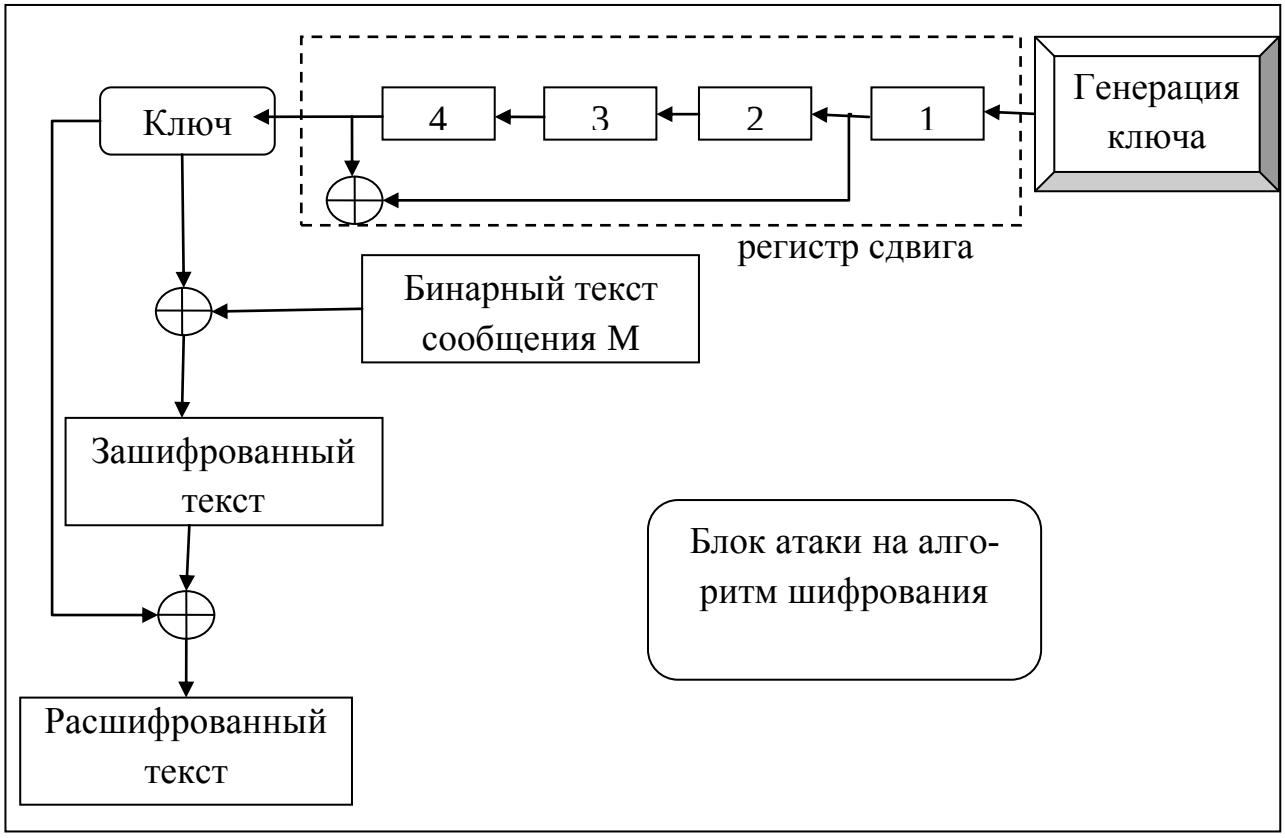

Рис. 2. Пример работы поточного шифра

2. ТЕСТЫ NIST

Тесты NIST – пакет статистических тестов, разработанный Лабораторией информационных технологий (англ. *Information Technology Laboratory*), являющейся главной исследовательской организацией Национального института стандартов и технологий (NIST). В его состав входят 15 статистических тестов, целью которых является определение меры случайности двоичных последовательностей, порождённых либо аппаратными, либо программными генераторами случайных чисел. Эти тесты основаны на различных статистических свойствах, присущих только случайным последовательностям.

Прежде, чем перейти к описанию каждого из тестов, необходимо сделать несколько уточнений по поводу используемого математического аппарата.

Оценка результатов теста происходит статистически. Предполагается гипотеза, что последовательность признается случайной с некоторой вероятностью (P-значение). Уровень статистической значимости для каждого из тестов NIST принимается в 1%, при таком его значении соблюдается баланс между возможными ошибками первого и второго рода. Из этого следует, что:

- Если P-значение ≥ 0.01, то последовательность признается случайной с уровнем доверия 99%
- Если P-значение < 0.01, то последовательность отбраковывается с уровнем доверия 99%

Для каждого из тестов есть свои ограничения по минимальной длине оцениваемой битовой последовательности. Обычно последовательность не должна быть меньше 100-1000 бит, иначе оценить ее вероятностные параметры и асимптотическую близость к нормальному распределению не представляется возможным.

2.1 Частотные тесты

1. Частотный побитовый тест.

Этот тест используется для оценки того, насколько близка доля единиц в полученной последовательности к 0,5. Следовательно, последовательность может быть признана пригодной для использования, если число нулей и единиц в ней примерно одинаково. Все последующие тесты проводятся при условии, что пройден данный тест.

Входные параметры:

Минимальная длина последовательности: 100 бит

Алгоритм тестирования:

Последовательность:1011010101

Ищем сумму всех битов последовательности. Вычисление производится по следующим формулам:

$$S_n = X_1 + X_2 + \dots + X_n, \quad (2.1)$$

где

$$X_i = 2x_i - 1, \quad (2.2)$$

где x_i- бит последовательности.

Для нашего примера получаем

S = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) + 1 + (-1) + 1 = 2

Находим частотные параметры:

$$s_{obs} = \frac{|S|}{\sqrt{n}} = \frac{2}{\sqrt{10}} = 0.632455532 \quad (2.3)$$

Находим p-значение посредством *дополнительной функции ошибок*:

$$P_{value} = erfc(\frac{s_{obs}}{\sqrt{2}}) = erfc(\frac{0.632455532}{\sqrt{2}}) = 0.527089 \quad (2.4)$$

Дополнительная функция ошибок формируется следующим образом:

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty \mathrm{e}^{-t^2} \mathrm{d}t \quad (2.5)$$

Так как конечное решение больше 0,01, исследуемая последовательность полностью прошла тест.

2.Частотный блочный тест

В случае абсолютно случайной последовательности, частота повторения единиц в блоке последовательности длиной *m* бит приблизительно равна *m*/2. Если принять *m* = 1, данный тест переходит в частотный побитовый тест.

Входные параметры:

Минимальная длина последовательности: 100 бит.

Длина блока M >= 20; в одном блоке должно быть больше сотой части бит последовательности (M > 0,01n); количество блоков N < 100.

Алгоритм тестирования:

Дана последовательность 0110011010. Распределим эту последовательность по разным блокам по 3 бита («ненужный» 0 на конце откинут):011 001 101

Вычислим долю единиц для каждого блока: $\pi_1 = 2/3$, $\pi_2 = 1/3$, $\pi_3 = 2/3$. Далее посчитаем статистику, используя для оценки функцию Хи-квадрат с N степенями свободы (по числу блоков):

$$\chi^2_{obs} = 4 \cdot M \cdot \sum_{i=1}^{N} (\pi_i - 1/2)^2 = 4 \cdot 3 \cdot [(\tfrac{2}{3} - \tfrac{1}{2})^2 + (\tfrac{1}{3} - \tfrac{1}{2})^2 + (\tfrac{2}{3} - \tfrac{1}{2})^2] = \quad (2.6)$$

Посчитаем p-значение, используя функцию *Q*:

$$P_{value} = Q(\tfrac{N}{2}, \tfrac{\chi^2_{obs}}{2}) = Q(3/2, 1/2) = 0.801252 \quad (2.7)$$

Q — *неполная верхняя гамма-функция*, которая характеризуемая следующим образом:

$$Q(a, x) = \tfrac{1}{\Gamma(a)} \int_x^\infty \mathrm{e}^{-t} t^{a-1} dt \quad (2.8)$$

А функция Γ - стандартная гамма-функция:

$$\Gamma(z) = \int_0^\infty \mathrm{t}^{z-1} e^{-t} dt \quad (2.9)$$

Так как значение p> 0,01, исследуемая последовательность прошла частотный блочный тест.

3. Тест на одинаковые идущие подряд биты

Проводится анализ рядов длиной *k* бит, состоящих из *k* абсолютно идентичных битов, которые начинаются и заканчиваются с бита, содержащего противоположное значение. В случае абсолютно случайной последовательности количество рядов, состоящих из единиц и нулей с различными длинами, соответствует их количеству в случайной последовательности. В частности, определяется быстро либо медленно чередуются единицы и нули в исходной последовательности. Чем реже происходит замена 0 на 1 (и обратно), тем меньше распределение бит последовательности близко к случайному.

Входные параметры:

Минимальная длина последовательности: 100 бит.

Алгоритм тестирования:

Дана последовательность 1001101011. Подсчитываем, какую часть занимают единицы во всей последовательности:

$$\pi = \frac{\sum_j X_j}{n} = \frac{6}{10} = \frac{3}{5} \quad (2.10)$$

Далее исследуется условие:

$$|\pi - \tfrac{1}{2}| < \frac{2}{\sqrt{n}} \quad (2.11)$$

Если оно не выполняется, весь тест признается неудавшимся. В данном случае 0.63246 > 0.1. Считаем сумму знакоперемен V:

$$V_n = \sum_{k=1}^{n-1} r(k) + 1 \quad (2.12)$$

где $r(k) = 0$, если $X_i = X_{i+1}$, или $r(k) = 1$ в обратном случае.
$V_{10} = (1 + 0 + 1 + 0 + 1 + 1 + 1 + 1 + 0) + 1 = 7$

Считаем p-значение посредством функции ошибок (см. формулу 2.5):

$$P_{value} = erfc\left(\frac{|V_n - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}\right) = erfc\left(\frac{7-(2\cdot 10\cdot\frac{3}{5}\cdot(1-\frac{3}{5}))}{2\cdot\sqrt{2\cdot 10}\cdot\frac{3}{5}\cdot(1-\frac{3}{5})}\right) = 0.147232 \quad (2.13)$$

Тестовая последовательность является случайной.

4. Тест на подпоследовательности

Данный тест заключается в подсчете частоты всех возможных перекрываний шаблонов длины *m* бит на протяжении исходной последовательности битов. Если каждый шаблон длиной *m* бит появляется в последовательности с одинаковой вероятностью, то она считается случайной. При *m*=1 тест на периодичность переходит в частотный побитовый тест.

Входные параметры:

Минимальная длина последовательности: 100 бит

Алгоритм тестирования:

Дана последовательность 0011011101, где n = 10 и m = 3.На ее основе создаются 3 новые последовательности прибавлением к окончанию искомой битовой последовательности m-1 первых битов. Таким образом:

Для m = 3: 0011011101 00 (добавили 2 бита к концу)

Для m-1 = 2: 0011011101 0 (добавили 1 бит к концу)

Для m-2 = 1: 0011011101 (исходная последовательность)

Частоты возникновения каждого блока длиной m, m-1 и m-2: $v_{000} = 0$, $v_{001} = 1$, $v_{010} = 1$, $v_{011} = 2$, $v_{100} = 1$, $v_{101} = 2$, $v_{110} = 2$, $v_{111} = 0$

$v_{00} = 1$, $v_{01} = 3$, $v_{10} = 3$, $v_{11} = 3$

$v_0 = 4$, $v_1 = 6$

Считаем необходимые статистики по формулам:

$$\psi_m^2 = \frac{2^m}{n} \cdot \sum_{i_1 \ldots i_m} (v_{i_1 \ldots i_m}^2) - n \quad (2.14)$$

$$\psi_{m-1}^2 = \frac{2^{m-1}}{n} \cdot \sum_{i_1 \ldots i_{m-1}} (v_{i_1 \ldots i_{m-1}}^2) - n \quad (2.15)$$

$$\psi_{m-2}^2 = \frac{2^{m-2}}{n} \cdot \sum_{i_1 \ldots i_{m-2}} (v_{i_1 \ldots i_{m-2}}^2) - n \quad (2.16)$$

В данном случае:

$$\psi_1^2 = \frac{2}{10}(16 + 36) - 10 = 10.4 - 10 = 0.4$$

$$\psi_2^2 = \frac{2^2}{10}(1 + 9 + 9 + 9) - 10 = 11.2 - 10 = 1.2$$

$$\psi_3^2 = \frac{2^3}{10}(0 + 1 + 1 + 4 + 1 + 4 + 4 + 0) - 10 = 12 - 10 = 2$$

$$\nabla\psi_m^2 = \psi_m^2 - \psi_{m-1}^2 = \psi_3^2 - \psi_2^2 = 2 - 1.2 = 0.8$$

$$\nabla^2\psi_m^2 = \psi_m^2 - 2\psi_{m-1}^2 + \psi_{m-2}^2 = 2 - 2 \cdot 1.2 + 0.4 = 0$$

Конечное решение:

$$P_{value1} = igamc(2^{m-2}, \nabla\psi_m^2) = igamc(2, \frac{0.8}{2}) = 0.93845 \quad (2.17)$$

$$P_{value2} = igamc(2^{m-3}, \nabla^2\psi_m^2) = igamc(1, \frac{0}{2}) = 1 \quad (2.18)$$

Здесь пара P-значений> 0.01, и, соответственно, последовательность является случайной.

5. Тест оценки приблизительной энтропии

Тест считает частоты возникновения различных образцов определенной длины (m) и такие же частоты для образцов длиной m+1. Если частоты перекрывания двух последовательных блоков с длинами *m* и *m+1* совпадают с частотами перекрывания аналогичных блоков в абсолютно случайной последовательности, то исследуемая последовательность сама случайна.

Алгоритм тестирования:

Дана последовательность 0100110101, где n = 10, m = 3

Для начала дополним последовательность первыми m-1 битами: 010011010101.

Вычислим встречаемость всех блоков:

$k_{000} = 0$, $k_{001} = 1$, $k_{010} = 3$, $k_{011} = 1$, $k_{100} = 1$, $k_{101} = 3$, $k_{110} = 1$, $k_{111} = 0$. Вычислим подобные частоты по формуле $C_i^m = k_i / n$: $C_{000}^3 = 0$, $C_{001}^3 = 0.1$, $C_{010}^3 = 0.3$, $C_{011}^3 = 0.1$, $C_{100}^3 = 0.1$, $C_{101}^3 = 0.3$, $C_{110}^3 = 0.1$, $C_{111}^3 = 0$.

Вычислим частоты возникновения подблоков длиной m+1=4:

$C_{0011}^4 = C_{0100}^4 = C_{0110}^4 = C_{1001}^4 = C_{1101}^4 = 0.1$, $C_{0101}^4 = 0.2$, $C_{1010}^4 = 0.3$.

Другие частоты = 0.

Считаем значения для соседних блоков, равные φ^3 и φ^4:

$$\phi^m = \sum_{i=0}^{2^m-1} \pi_i \ln \pi_i = \sum_{i=0}^{2^m-1} C_i^3 \ln C_i^3 = 0 + 0.1(ln0.1) + 0.3(ln0.3) - \quad (2.19)$$
$$0.1(ln0.1) + 0.1(ln0.1) + 0.3(ln0.3) + 0.1(ln0.1) = -1.64341772$$

$$\phi^{m+1} = 0 + 0 + 0 + 0.1(ln0.01) + 0.1(ln0.01) + 0.2(ln0.02) + 0.1(ln0.01) \quad (2.20)$$
$$0 + 0 + .1(ln0.01) + 0.3(ln0.03) + 0 + 0 + 0.1(ln0.01) + 0 + 0 = -1.83437197$$

Считаем Хи - квадрат:

$$\chi^2 = 2n(ln2 - (\phi^3 - \phi^4) = 2 \cdot 10 \cdot (0.693147 + 1.64341772 - 1.83437197) = 10.044 \quad (2.21)$$

P-значение:

$$P_{value} = igamc(2^{m-1}, \frac{\chi^2}{2}) = igamc(4, \frac{10.044}{2}) = 0.261961 \quad (2.22)$$

Конечное решение > 0.01, следовательно. последовательность является случайной.

2.2 Поиск шаблонов

1. Тест на встречающиеся непересекающиеся шаблоны

В данном тесте подсчитывается количество заранее определенных шаблонов – битовых последовательностей конкретного содержания и длины – найденных в исходной последовательности для выявления частоты появления непериодических шаблонов. Термин "*непересекающиеся"* предполагает, что, когда шаблон расположен в последовательности, вытекающее сравнение не фик-

сирует ни одного бита обнаруженного шаблона. Для поиска конкретных шаблонов длиной *m* бит используется окно также длиной *m* бит. Если шаблон не обнаружен, окно смещается на один бит. Если же шаблон найден, окно перемещается на бит, следующий за найденным шаблоном, и поиск продолжается дальше. Чем больше различных шаблонов, тем более последовательность близка к случайной.

Входные параметры:

Длина окна *m* - 9 или 10;

N (количество блоков) < 100;

M (длина блока) > 0.01 * n, где n – длина битовой последовательности.

Алгоритм тестирования:

Последовательность, над которой проводится тест, раскладывается блоками равной длины. Например:1010010010 1110010110. Пусть используется шаблон «001», который нужно отыскать в отдельных блоках. Вследствие исследования всех i-ых блоков обнаружится число W_i совпадений. Касательно взятых блоков $W_1 = 2$ и $W_2 = 1$:

101 **001 001** 0

111 **001** 0110

Найдем математические ожидание и дисперсию, принимая последовательность как подлинно случайную, где N = 2, M = 10, m = 3.

$$\mu = \frac{M-m+1}{2^m} = \frac{10-3+1}{2^3} = 1 \quad (2.23)$$

$$\sigma^2 = M \cdot (\frac{1}{2^m} - \frac{2m-1}{2^{2m}}) = 10 \cdot (\frac{1}{2^3} - \frac{2 \cdot 3-1}{2^{2 \cdot 3}}) \quad (2.24)$$

Посчитаем Хи - квадрат:

$$\chi^2_{obs} = \sum_{j=1}^{N} \frac{(W_j-\sigma)^2}{\sigma^2} = \frac{(2-1)^2+(1-1)^2}{0.46875} = \frac{1+0}{0.46785} = 2.13 \quad (2.25)$$

Посчитаем конечное p-значение посредством неполной гамма - функции:

$$Q(a,x) = Q(\frac{N}{2}, \frac{\chi^2_{obs}}{2}) = Q(\frac{2}{2}, \frac{2.13333}{2}) = 0.344154 \quad (2.26)$$

Исходя из полученного результата, последовательность является случайной.

2. Тест на встречающиеся пересекающиеся шаблоны

В данном случае отличие от предшествовавшего теста заключается в том, что, обнаружив шаблон, «поле» поиска смещается лишь на один бит. Образец исследования последовательности данным тестом идентичен предыдущему.

3. Универсальный тест Мауэра

Тест ориентирован на оценку числа бит между одинаковыми шаблонами в исходной последовательности (мера, имеющая непосредственное отношение к длине сжатой последовательности). Чем меньше число бит между шаблонами, тем лучше данная последовательность может быть значительно сжата без потерь информации. В случае если это возможно сделать, она не является истинно случайной.

4. Тест на самую длинную последовательность из единиц в блоке

В данном тесте определяется самый длинный ряд единиц внутри блока длиной *m* бит. Последовательность считается пригодной, если с вероятностью $p>0.01$ длина такого ряда соответствует ожиданиям длины самого протяжённого ряда единиц в случае абсолютно случайной последовательности.

Входные параметры:

Начальная последовательность из n-го количества битов распределяется на некоторое количество блоков N, во всех по M бит. Используются следующие значения n и M:

Таблица 1

Виды разбиения последовательности на блоки

Общая длина, n	Длина блока, M

128	8
6272	128
750000	10000

Алгоритм тестирования:

Допустим предоставлена последовательность:11001100000101010110110001001100111000000000001001001101010100010001001111010110100000001101011111001100111001101101100010110010

Распределим последовательность блоками по восемь бит в каждом (M=8), далее вычислим наибольшую последовательность из единиц для отдельных блоков: 1 блок -2, 2 блок – 1, 3 блок – 2, 4 блок – 2, 5 блок – 3, 6 блок – 1, 7 блок – 2, 8 блок – 1, 9 блок – 2, 10 блок -2, 11 блок – 1, 12 блок – 3, 13 блок – 2, 14 блок – 3, 15 блок - 2, 16 блок -2.

Следующим шагом является вычисление статистики по различным длинам на базе представленной таблицы 2:

Таблица 2

Вычисление статистики

v_i	M = 8	M = 128	M = 10000
v_0	≤ 1	&le4	&le10
v_1	2	5	11
v_2	3	6	12
v_3	≥ 4	7	13
v_4	-	8	14
v_5	-	≥ 9	15
v_6	-	-	≥ 16

Вычисляем $v_{i,}$ пользуясь данными таблицы 2 для M = 8:

v_0 = { *количество блоков с макс. длиной* $\leq$ *1* } = 4

v_1 = { *количество блоков с макс. длиной* = *2* } = 9

v_2 = { *количество блоков с макс. длиной* = *3* } = 3

v_3 = { *количество блоков с макс. длиной* $\geq$ *4* } = 0

Считаем Хи - квадрат. Переменные K и R приведены в таблице 3.

$$\chi^2 = \sum_{i=0}^{K} \frac{(v_i - R\pi_i)^2}{R\pi_i} \qquad (2.27)$$

Таблица 3

Вычисление констант

M	K	R
8	3	16
128	5	49
10000	6	75

Теоретические вероятности π_i исчисляются постоянными. В частности, если брать K=3 и M=8 используются $\pi_0 = 0.2148$, $\pi_1 = 0.3672$, $\pi_2 = 0.2305$, $\pi_3 = 0.1875$.

$$\chi^2_{(obs)} = \frac{(4-16\cdot 0.2148)^2}{16\cdot 0.2148} + \frac{(9-16\cdot 0.3672)^2}{16\cdot 0.3672} + \frac{(3-16\cdot 0.2305)^2}{16\cdot 0.2305} + \frac{(0-16\cdot 0.1875)^2}{16\cdot 0.1875} = 4.882 \qquad (2.28)$$

Следующим шагом будет расчет P - значение:

$$P_{value} = igamc(\frac{K}{2}, \frac{\chi^2_{(obs)}}{2}) = igamc(\frac{3}{2}, \frac{4.882605}{2}) = 0.180598 \qquad (2.29)$$

Рассматриваемая последовательность является в нужной степени случайной.

2.3 Тест рангов бинарных матриц

Тест основан на расчёте рангов непересекающихся подматриц, построенных из исходной двоичной последовательности. Если подстроки фиксированной длины первоначальной последовательности линейно зависимы, то битовая последовательность с вероятностью p>0.01 считается случайной.

Входные параметры:

Длина последовательности L >= 38MQ

M = Q = 32 (размерность матрицы), длина последовательности n = M^2 * N.

Алгоритм тестирования:

В данном примере M = Q = 3. Используем вероятности P_M, P_{M-1} и P_{M-2}. Упрощаем вычисления, работая с малой частью погрешности. В этом случае вероятности равнозначны:

$$P_M \approx \prod_{j=1}^{\infty}[1 - \tfrac{1}{2^j}] \approx 0.2888 \quad (2.30)$$

$$P_{M-1} \approx 2P_M \approx 0.5776 \quad (2.31)$$

$$P_{M-2} \approx \tfrac{4P_M}{9} \approx 0.1284 \quad (2.32)$$

$$\Rightarrow 1 - 0.2888 - 0.5776 = 0.1336 \quad (2.33)$$

Дана последовательность 01011001001010101101. Последовательность разбивается на 2 матрицы:

$$A_1 = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{vmatrix}$$

$$A_2 = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{vmatrix}$$

Устанавливаем ранг матриц: $R_1 = 2$, $R_2 = 3$. Нам же необходимо 3 числа:

F_M = {количество матриц с рангом M} = {количество матриц с рангом 3} = 1

F_{M-1} = 1 (так же)

N — F_M — F_{M-1} = 2 — 1 — 1 = 0

Считаем Хи - квадрат

$$\chi^2 = \sum \frac{(F_i - NP_i)^2}{NP_i} \quad (2.34)$$

$$\chi^2_{(obs)} = \frac{(F_M - 0.2888N)^2}{0.2888N} + \frac{(F_{M-1} - 0.5776N)^2}{0.5776N} + \frac{(N - F_M - F_{M-1} - 0.1336N)^2}{0.1336N} = 0.596953 \quad (2.35)$$

Считаем p - значение:

$$P_{value} = igamc(1, \frac{\chi^2_{(obs)}}{2}) = e^{\frac{-\chi^2_{(obs)}}{2}} = e^{-0.596953/2} = 0.741948 \quad (2.36)$$

Последовательность является случайной.

2.4 Спектральный тест

Тест оценивает высоту пиков дискретного преобразования Фурье исходной последовательности. Наличие таких пиков свидетельствует о ее периодических свойствах, что является признаком отклонения такой последовательности от случайной. Если доля пиков, превышающих 95%-й барьер, не больше 5%, то последовательность считается истинно случайной.

Входные параметры:

Длина последовательности L> 1000 бит

Алгоритм тестирования:

Дана последовательность 1001010011. Заменяем нули на -1. Здесь x={1,-1,-1,1,-1,1,-1,-1,1,1}.

При разложении подобной последовательности при помощи fft имеем следующие пики: `ans = 0.0000 2.0000 4.4721 2.0000 4.4721 2.0000 4.4721 2.0000 4.4721 2.0000`

Имеем 5 различных значений: 0, 2, 4.4721, 2, 4.4721

Произведем расчет граничного значения:

$$T = \sqrt{log(\frac{1}{0.05}) \cdot n} \approx 5.47 \quad (2.37)$$

Итог разделения: 4 пика не превышают барьера.

Посчитаем теоретически максимальное количество пиков N_0, где $N_0 < T$.

$$N_0 = \frac{0.95n}{2} = 4.75 \quad (2.38)$$

Даем оценку разности:

$$d = \frac{N_1 - N_0}{\sqrt{n \cdot 0.95 \cdot 0.05/4}} = \frac{4 - 4.75}{\sqrt{10 \cdot 0.95 \cdot 0.05/4}} = -2.176429 \quad (2.39)$$

Считаем p - значение:

$$P_{value} = erfc(\frac{|d|}{\sqrt{2}}) = erfc(\frac{2.176429}{\sqrt{2}}) = 0.029523 \quad (2.40)$$

Так как p-значение>0.01, то последовательность является случайной.

2.5 Тест на линейную сложность

В основе теста лежит анализ последовательности, представленной в качестве битов, полученных в результате работы линейного регистра сдвига с обратной связью (*LFSR*). Абсолютно случайные последовательности характеризуются длинными линейными регистрами сдвига с обратной связью. Если же такой регистр слишком короткий, то предполагается, что последовательность не является в полной мере случайной.

Искомая последовательность раскладывается одинаковыми блоками длиной M. Затем по всем отдельным блокам посредством алгоритма Берлекэмпа–Месси определяется их линейная сложность (L_i), то есть длина регистра. Следующим шагом у всех полученных L_iнаходят значение функции Хи - квадрат со 6 степенями свободы.

Входные параметры:

Длина последовательности n >= 10^6;

количество блоков M: 500 – 5000.

Алгоритм тестирования:

Рассмотрим блок 1101011110001 (M=13), для которого алгоритм Берлекэмпа — Месси дает значение L = 4. Для этого блока каждый следующий бит получается как сумма (по модулю 2) 1-го и 2-го бита (нумерация с 1):

$x_5 = x_1 + x_2 = 1+1=0$

$x_6 = x_2 + x_3 = 1+0=1$

$x_7 = x_3 + x_4 = 1+0=1$

и так далее.

Считаем математическое ожидание по формуле

$$\mu = \frac{M}{2} + \frac{9+(-1)^{M+1}}{36} - \frac{\frac{M}{3}+\frac{2}{9}}{2^M} = \frac{13}{2} + \frac{9+(-1)^{13+1}}{36} - \frac{\frac{13}{3}+\frac{2}{9}}{2^{13}} = 6.77722 \quad (2.41)$$

Находим величину T_i по всем блокам:

$$T_i = (-1)^M \cdot (L_i - \mu) + \frac{2}{9} = (-1)^{13} \cdot (4 - 6.777222) + \frac{2}{9} = 2.999444 \quad (2.42)$$

Затем считаем набор $v_0,...,v_6$, и получаем: если $T_i <= -2.5$, то v_0++

если $-2.5 < T_i <= -1.5$, то v_1++

если $-1.5 < T_i <= -0.5$, то v_2++

если $-0.5 < T_i <= 0.5$, то v_3++

если $0.5 < T_i <= 1.5$, то v_4++

если $1.5 < T_i <= 2.5$, то v_5++

если $T_i > 2.5$, то v_6++

У нас есть 7 вариантов. Вычисляем Хи-квадрат с числом степеней свободы 6:

$$\chi^2 = \sum_{i=0}^{K} \frac{(v_i - N\pi_i)^2}{N\pi_i} \quad (2.43)$$

Вероятности π_i в данном способе твердо зафиксированы: 0.010417, 0.03125, 0.125, 0.5, 0.25, 0.0625, 0.020833. Считаем p-значение:

$$P_{value} = igamc(\frac{K}{2}, \frac{\chi^2}{2}) \quad (2.44)$$

Если p> 0.01, последовательность считается случайной

2.6 Поиск частичных сумм битов последовательности

1. Тест кумулятивных сумм

Тест исследует частичные суммы битов последовательности и заключается в поиске максимального отклонения (от нуля) кумулятивной суммы заданных (-1, +1) цифр в последовательности. Если значение отклонения от произвольного обхода будет вблизи нуля, то последовательность считается случайной.

Алгоритм тестирования:

Мы берем все «0» за +1, а все «1» за -1, и складываем сумму:

$S_1 = x_1$

$S_2 = x_1 + x_2$ $S_3 = x_1 + x_2 + x_3 \ldots$

$S_n = x_1 + x_2 + x_3 + \ldots + x_n$

Затем ищется z = max данных сумм. Вычисляется p-значение:

$$P_{value} = 1 - \Sigma_1 + \Sigma_2 \quad (2.45)$$

$$\Sigma_1 = \sum_{k=(\frac{-n}{z}+1)\cdot 4}^{(\frac{n}{z}-1)\cdot 4} \left[\Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k-1)z}{\sqrt{n}}\right)\right] \quad (2.46)$$

$$\Sigma_2 = \sum_{k=(\frac{-n}{z}-3)\cdot 4}^{(\frac{n}{z}-1)\cdot 4} \left[\Phi\left(\frac{(4k+3)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right)\right] \quad (2.47)$$

Здесь Ф — стандартная функция нормального распределения, где математическое ожидание 0 и дисперсия 1.

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} \mathrm{e}^{\frac{-u^2}{2}} du \quad (2.48)$$

Когда конечное P-значение > 0.01, последовательность является случайной.

2. Тест на произвольные отклонения

Суть данного теста заключается в подсчёте числа циклов, имеющих строго *k* посещений при произвольном обходе кумулятивной суммы. Произвольный обход кумулятивной суммы начинается с частичных сумм после последовательности (0,1), переведённой в соответствующую последовательность (-1, +1). Цикл произвольного обхода состоит из серии шагов единичной длины, совершаемых в случайном порядке. Кроме того такой обход начинается и заканчивается на одном и том же элементе. Цель данного теста — определить отличается ли число посещений определенного состояния внутри цикла от аналогичного числа в случае абсолютно случайной входной последовательности. Фактически данный тест есть набор, состоящий из восьми тестов, проводимых для каждого из восьми состояний цикла: −4, −3, −2, −1 и +1, +2, +3, +4. В каждом таком тесте принимается решение о степени случайности исходной последовательности в соответствии со уже известным правилом.

Алгоритм тестирования:

Дана последовательность 0110110101, а S(i) — является частичной суммой с 1 по i - й элемент (рисунок 3). Добавим «0» с двух сторон последовательности S(i) — это необходимо с целью целостности будущих расчётов:

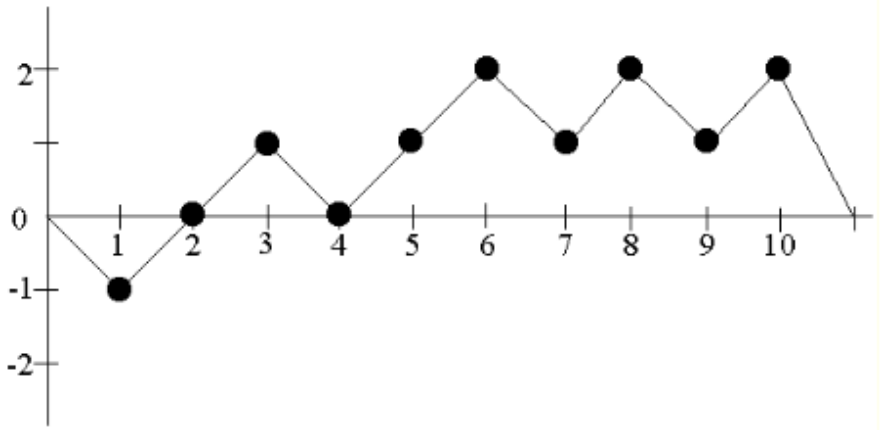

Рис. 3. Частичные суммы

Последовательность разбивается на циклы. В данном случае их получается три: {0, -1, 0}, {0, 1, 0} и {0, 1, 2, 1, 2, 1, 2, 0}. Затем они поочередно приобретают разнообразные *состояния*. В частности, 1-ый цикл дважды приобретает состояние «0» и единожды состояние "-1". В этом тесте значение «хи квадрат» имеют состояния в промежутке [–4;4]. В соответствие с проведенными расчетами имеем таблицу состояний:

Таблица 4

Состояния расчетных циклов

Состояние (x)	Цикл №1	Цикл №2	Цикл №3
-4	0	0	0
-3	0	0	0
-2	0	0	0
-1	1	0	0
1	0	1	3
2	0	0	3
3	0	0	0
4	0	0	0

На базе предыдущей таблицы, создаем иную таблицу числа циклов:

Таблица 5

Новые состояния расчетных циклов

Состояние (x)	Ни разу	1 раз	2 раза	3 раза	4 раза	5 раз
-4	3	0	0	0	0	0
-3	3	0	0	0	0	0
-2	3	0	0	0	0	0
-1	2	1	0	0	0	0
1	1	1	0	1	0	0
2	2	0	0	1	0	0
3	3	0	0	0	0	0
4	3	0	0	0	0	0

Затем считается Хи - квадрат статистики по формуле:

$$\chi^2_{obs} = \sum_{k=0}^{5} \frac{(v_k(x) - J\pi_k(x))^2}{J\pi_k(x)} \quad (2.49)$$

Здесь $v_k(x)$ — значения в таблице для данного состояния, J — число циклов (в нашем случае 3), $\pi_k(x)$ — вероятности того, что состояние «x» появится k раз в подлинно случайном распределении (они известны). В частности, когда x=1:

$$\chi^2 = \frac{(1-3\cdot0.5)^2}{3\cdot0.5} + \frac{(1-3\cdot0.25)^2}{3\cdot0.25} + \frac{(0-3\cdot0.125)^2}{3\cdot0.125} + \frac{(1-3\cdot0.0625)^2}{3\cdot0.0625} + \frac{(0-3\cdot0.0312)^2}{3\cdot0.0312} + \frac{(0-3\cdot0.0312)^2}{3\cdot0.0312} = 4.333 \quad (2.50)$$

Считаем p-значение:

$$P_{value} = igamc(\frac{k}{2}, \frac{\chi^2_{obs}}{2}) = igamc(5/2, 4.333/2) = 0.502529 \quad (2.51)$$

Когда конечное p-значение > 0.01, последовательность является случайной.

Суммарно нужно посчитать восемь p-значений, а заключительное решение о последовательности принимается на базе иных тестов.

3.Разновидность теста на произвольные отклонения

В этом тесте подсчитывается общее число посещений определенного состояния при произвольном обходе кумулятивной суммы. Тест проводится для набора состояний: −9, −8, …, −1 и +1, +2, …, +9. Для каждого случая определяются отклонения от ожидаемого числа посещений различных состояний при произвольном обходе. На каждом этапе делается вывод о случайности входной последовательности[6].

3.АТАКИ НА ПОТОЧНЫЕ ШИФРЫ

Компьютерная атака - целенаправленное несанкционированное воздействие на информацию, на ресурс информационной системы или получение несанкционированного доступа к ним с применением программных или программно-аппаратных средств.

Атаки на алгоритм шифрования с целью получения несанкционированного доступа к сообщению принято считать криптоанализом алгоритма.

Криптоанализ (от др.-греч. κρυπτός — скрытый и анализ) — наука о методах расшифровки зашифрованной информации без предназначенного для такой расшифровки ключа. Термин был введён американским криптографом Уильямом Ф.

3.1. Классификация атак

Все методы криптоанализа подразделяются на силовые, статистические и аналитические.

Силовые атаки:

–атаки, осуществляющие полный перебор всех возможных вариантов ключевой последовательности

Сложность полного перебора зависит от количества всех возможных решений задачи (от размера пространства ключей или открытых текстов).

Статистические атаки:

– метод криптоанализа статистических свойств шифрующей последовательности: направлен на изучение выходной последовательности криптосистемы при помощи статистических тестов.

– метод криптоанализа сложности последовательности направлен на генерацию последовательности, аналогичную используемой.

Оба метода имеют линейную сложность (зависимость количества итераций алгоритма от размера его входных данных).

Аналитические атаки:

– атаки на определение использованного ключа при известных открытых и соответствующих закрытых текстах.

Типы атак:

– *корреляционные*: основаны на поиске корреляции выходной последовательности криптосистемы с выходной последовательностью регистров генератора(базовые, низко-весовая проверка четности, анализ сверточных кодов, турбо-кодов, восстановление линейных полиномов(используемый алгоритм Берлекэмпа-Месси), атаки с минимальной вычислительной сложностью);

–*компромисс «время-память»* восстановление исходного состояния регистра с использованием схемы генератора псевдослучайной последовательности и фрагмента шифрующей последовательности (атака Стива Бэббиджа, атака Бирюкова-Шамира).

– *«предполагай и определяй»*: атака с использованием псевдослучайной последовательности, полинома обратной связи, для ее успеха необходимо знание о количестве сдвигов регистра между выходами схемы и фильтрующей функции.

Далее рассмотрим атаку на поточный шифр на примере одного из видов аналитических атак в виде алгоритма Берлекэмпа-Месси.

3.2. Алгоритм Берлекэмпа-Месси

Рассмотрим, как происходит атака на симметричный поточный шифр на примере алгоритма восстановления примитивного неприводимого полинома – основы для линейного регистр сдвига.

Существует теорема о возможности восстановления полинома по созданной с его помощью псевдослучайной последовательности.

Теорема: Рассмотрим регистр сдвига длины L с линейной обратной связью, порождающий битовую последовательность z длины N, где N может быть и бесконечным. Тогда

- L последовательных состояний регистра линейно независимы;

- L + 1 последовательных состояний регистра линейно зависимы;
- Если $N \geq 2L$ символов последовательности заданы, то полином, задающий обратные связи, однозначно определен.

Если гипотеза о линейной сложности L верна, то существует единственное решение этой системы уравнений, являющееся решением этой задачи.

Тот факт, что на практике сложность L не известна заранее, не представляет проблемы, так как можно по очереди проверять гипотезы L =1,2,.... Решение системы из L уравнений имеет сложность не более L^3, поэтому общая сложность нахождения регистра не превышает L^4, т.е. остается в любом случае полиномиальной.

Алгоритм Берлекэмпа-Месси решает задачи определения линейной сложности и нахождения коэффициентов полинома со сложностью порядка L^2. Алгоритм выполняет поиск регистра сдвига с обратной связью наименьшей длины L, генерирующего заданную последовательность бит при определенном начальном состоянии, где L<n.

В основе алгоритма лежит рекурсия: для каждого элемента последовательности r, начиная с r = 1, строим регистр, генерирующий первые r элементов последовательности. Обозначим длину построенного регистра через L_r, а сам регистр с обратной связью опишем вектором коэффициентов $c(r) = (c(r_1), c(r_2), ..., c(r_{Lr}))$. На r -й итерации вычисляем r -й выход предыдущего $(r-1)$ -го регистра:

$$\hat{z}_r = \sum_{j=1}^{r-1} c_j^{(r-1)} z_{r-j} . \qquad (3.1)$$

На самом деле степень полинома $c(r-1)$ может быть меньше, чем r – 1 но для упрощения записи не учитывается, что многие слагаемые в последней сумме тождественно равны нулю. Истинный элемент последовательности может не совпадать с полученным по формуле. «Невязку» запишем в виде:

$$\Delta_r = z_r + \hat{z}_r = z_r + \sum_{j=1}^{r-1} c_j^{(r-1)} z_{r-j} = \sum_{j=0}^{r-1} c_j^{(r-1)} z_{r-j}\,, \tag{3.2}$$

где подразумевается, что $c_0(r-1) = 1$. Если невязка нулевая, то итерация закончена. Иначе нужно модифицировать вектор, задающий регистр, чтобы сделать ее нулевой. Новый полином будем искать в виде

$$c^{(r)}(x) = c^{(r-1)}(x) + x^l c^{(m-1)}(x)\,, \tag{3.3}$$

где $c(m-1)(x)$ - это один из уже использовавшихся полиномов, а $m < r$ таково, что невязка на m -м шаге была ненулевой. Если к тому же положить $l = r - m$, то после такой модификации имеем:

$$\Delta_r' = \sum_{j=0}^{r-1} c_j^{(r-1)} z_{r-j} + \sum_{j=0}^{r-1} c_j^{(m-1)} z_{r-j-l} = \Delta_r + \Delta_m = 1 + 1 = 0 \tag{3.4}$$

Значение m выбирается так, чтобы минимизировать длину получаемого регистра. Оказывается, что этому условию удовлетворяет выбор последнего m, при котором невязка была равна 1[7].

Алгоритм Берлекэмпа-Месси.

Вход –последовательность z длины n.

1. Инициализация: $r = 0$, $c(x) = 1$, $b(x) = 1$.
2. Полагаем $r \leftarrow r + 1$. Вычисляем невязку

$$\Delta = z_r + \sum_{j=1}^{r-1} c_j z_{r-j} = \sum_{j=0}^{r-1} c_j z_{r-j} \tag{3.5}$$

3. Если $\Delta = 0$, сдвиг: $b(x) = xb(x)$, переходим к шагу 5, в противном случае выполняем шаг 4.

4. Формируем новый полином $t(x) \leftarrow c(x) + xb(x)$. Сохраняем предыдущий полином: $b(x) = c(x)$. Меняем связи регистра $c(x) = t(x)$.

5. Если $r < n$, возвращаемся к шагу 2. В противном случае работа закончена, результаты работы алгоритма – полином $c(x)$ и линейная сложность последовательности z, равная $L = \deg c(x)$.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите особенности симметричных шифров. Каковы их достоинства и недостатки?
2. Что такое имитационная модель работы алгоритма шифрования? Какую роль играет моделирование для успешного функционирования криптографической системы?
3. Как создается генератор М-последовательности?
4. Перечислите основные статистические свойства тестирующей двоичной последовательности.
5. При помощи каких механизмов тесты NIST решают задачу тестирования двоичной последовательности? Каковы основные ограничения на параметры тестируемой последовательности?
6. Чем отличается тест на поиск подпоследовательностей (относящийся к частотным тестам) от тестов на поиск шаблонов?
7. Какие из тестов NIST предоставляют больше информации о тестируемой последовательности?
8. Что такое атака на алгоритм шифрования? Перечислите основные виды атак?
9. К какому типу атак относится алгоритм Берлекэмпа-Месси?
10. На компрометацию какого из элементов системы шифрования направлены действия алгоритма Берлекэмпа-Месси?

ЗАДАНИЕ

Задание посвящено изучению работы алгоритмов симметричных шифров, а также изучению механизмов атак с целью получению доступа к алгоритму генерации ключей.

Исследование проводится при помощи готовой компьютерной имитационной модели, в состав которой входят: 1) блок криптосистемы: генератор М-последовательности и поле вывода полученной псевдослучайной последовательности, поле ввода двоичного сообщения, поле вывода зашифрованного сообщения; 2) блок атаки: поле ввода L перехваченных бит ключа и поле вывода полученного примитивного полинома; 3) блок перехвата: поле ввода зашифрованного сообщения, поле ввода ключа и поле вывода расшифрованного сообщения.

Ознакомившись с теоретическим материалом, необходимо выполнить следующие задания на имитационной модели: (для моделирования работы генератора М-последовательности используются примитивные неприводимые полиномы из списка, представленного на рисунке 4)

$x^{31}+x^{3}+1$	$x^{31}+x^{6}+1$	$x^{31}+x^{7}+1$	$x^{33}+x^{13}+1$
$x^{71}+x^{7}+1$	$x^{93}+x^{2}+1$	$x^{137}+x^{21}+1$	$x^{35}+x^{2}+1$
$x^{145}+x^{52}+1$	$x^{161}+x^{18}+1$	$x^{521}+x^{32}+1$	$x^{47}+x^{5}+1$
$x^{55}+x^{24}+1$	$x^{58}+x^{19}+1$	$x^{57}+x^{7}+1$	$x^{52}+x^{49}+1$

Рис. 4. Список примитивных неприводимых полиномов

1.Пользуясь имитационной моделью, создайте генератор М-последовательности на основе выбранного примитивного неприводимого полинома и сгенерируйте битовый ключ шифрования.

2. Оцените случайность распределения битов ключа при помощи тестов NIST командой Test('text.txt',6,9), набранной в командной строке среды Matlab.

В текстовом файле 'text.txt' сохраняется исследуемая последовательность бит, первый параметр – минимальное количество бит между одинаковыми шаблонами в универсальном тесте Мауэра, второй параметр необходим для оценки тестов на приблизительную энтропию (длина искомых образцов) и ранговых бинарных матриц (длина подпоследовательностей-строк матрицы):

1) Проанализируйте результаты частотных тестов. Соберите статистику. Объясните прохождение или провал того или иного отдельного теста.

2) Проанализируйте результаты тестирования при помощи заданных битовых шаблонов. Соберите статистику. Объясните прохождение или провал того или иного отдельного теста.

3) Проанализируйте результаты тестирования, использующего поиск частичных сумм битов последовательности. Соберите статистику. Объясните прохождение или провал того или иного отдельного теста.

4) Проведите тест рангов бинарных матриц над последовательностью. Объясните и при необходимости улучшите полученный результат.

5) Проведите спектральный тест над последовательностью. Объясните и при необходимости улучшите полученный результат.

6) Проведите тест на линейную сложность над последовательностью. Объясните и при необходимости улучшите полученный результат.

7) Оцените случайность распределения битов передаваемого сообщения (размером не менее 100 бит) при помощи тестов NIST. Объясните полученные результаты.

В случае провала отдельных тестов улучшите статистические свойства битовой последовательности так, чтобы эти тесты были пройдены.

3.Зашифруйте и расшифруйте заданное битовое сообщение при помощи сгенерированного ключа.

4.Произвести атаку на симметричный поточный шифр при помощи алгоритма Берлекэмпа-Месси (раздел 3.2):

1) При помощи L бит перехваченной ключевой последовательности восстановите полином, на основе которого она была сгенерирована.

2) Зная зашифрованную последовательность, восстановите исходное сообщение. Убедитесь в том, что оно совпадает с отправленным сообщением. Почему это возможно?

3) Каким должно быть минимальное число L бит перехваченной ключевой последовательности для восстановления с ее помощью используемого полинома? Укажите необходимое количество бит для Вашей ключевой последовательности.

Результаты выполнения заданий оформите в качестве отчета, выполняя задание по пунктам. Ответы должны быть максимально подробными и сопровождаться примерами с результатами моделирования (скриншотами).

СПИСОК ЛИТЕРАТУРЫ

1. Смарт Н. Криптография./ Н. Смарт - М.: Техносфера, 2005. 528 с.

2. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии./ А.П. Алферов, А.Ю. Зубов, А.С. Кузьмин, А.В. Черемушкин - М.: Гелиос АРВ, 2002. 480 с.

3. Шеннон Р. Имитационное моделирование систем / Р. Шеннон – Искусство и наука: Перевод с англ. – М.: Мир, 1978. – 418с.

4. Карпов А.В. Компьютерное имитационное моделирование. Учебное пособие для магистрантов и студентов старших курсов./ А.В. Карпов - Казань: Казан.ун., 2004. -79 с.

5. Карпов А.В., Любимов Д.В., Сулимов А.И.. Введение в криптографию. Учебно-методическое пособие для выполнения лабораторных работ./ А.В. Карпов, Д.В. Любимов, А.И. Сулимов - Казань, 2013. - 37 с.

6. Статистическая проверка случайности двоичных последовательностей методами NIST. Электрон.ресурс компании Код Безопасности, Криптография. URL: habrahabr.ru/company/securitycode/blog/237695/

7. Блейхут Р. Теория и практика кодов, контролирующих ошибки./ Р. Блейхут – М.: Мир. 1986. -576с.

Учебное издание

Карпов Аркадий Васильевич
Туктарова Ирина Ринатовна
Смоляков Алексей Дмитриевич

ИМИТАЦИОННАЯ КОМПЬЮТЕРНАЯ МОДЕЛЬ КРИПТОГРАФИЧЕСКОЙ СИСТЕМЫ, ОСНОВАННАЯ НА ГЕНЕРАТОРАХ М-ПОСЛЕДОВАТЕЛЬНОСТИ

Дизайн обложки
М.А. Ахметов

Подписано в печать 26.06.2015.
Бумага офсетная. Печать цифровая.
Формат 60x84 1/16. Гарнитура «Times New Roman». Усл. печ. л. 34
Тираж 10 экз. Заказ 2

Отпечатано с готового оригинал-макета
в типографии Издательства Казанского университета

420008, г. Казань, ул. Профессора Нужина, 1/37
тел. (843) 233-73-59, 233-73-28