УДК 681.3.06

Анализ эффективности градиентной статистической атаки на блоковые шифры RC6, MARS, CAST-128, IDEA, Blowfish в системах защиты информации

А.С. Лысяк, Б.Я. Рябко, А.Н. Фионов

Данная работа посвящена экспериментальному исследованию статистических методов в криптоанализе на примере блоковых шифров RC6, MARS, CAST-128, IDEA и Blowfish. Предложенная градиентная атака основана на статистическом тесте «стопка книг». Описанные схемы и модификации атаки позволяют значительно уменьшить трудоёмкость нахождения секретного ключа по сравнению с ранее известными видами атак. В данной работе проведены исследования эффективности градиентной статистической атаки, показаны пределы её современной практической и теоретической применимости (9 раундов шифра RC6, 21 раунд MARS и 4 раунда CAST-128), выведены аналитические зависимости между эффективно атакуемыми раундами и временно́й и аппаратной сложностью атаки. Кроме того, для шифров RC6 и MARS разработаны модификации градиентной атаки, значительно сокращающие временну́ю и операционную сложность. Также в данной работе показан метод подбора оптимальных параметров атаки; исследована операционная и пространственная трудоёмкости и их зависимости от параметров теста и объёма шифротекста; показаны теоретические потребности в вычислительных ресурсах, необходимых для осуществления атаки.

Ключевые слова: градиентная атака, статистическая атака, криптоанализ, блоковые шифры, RC6, MARS, CAST-128, IDEA, Blowfish.

1. Введение

Блоковые шифры с секретным ключом находят широкое применение в системах защиты, передачи и хранения информации, что делает актуальными как задачи построения надёжных блоковых шифров, так и задачи поиска эффективных криптологических атак на эти шифры (т.е. методов определения секретного ключа шифра на основе экспериментов с зашифрованными сообщениями). Многие специалисты занимаются исследованием теоретической и практической устойчивости блоковых шифров к атакам различного рода. Постоянно разрабатываются как новые шифры, так и новые атаки на них, позволяющие оценить реальную устойчивость шифров. Отметим, что для криптографии практический интерес представляют только те атаки, которые позволяют находить ключ за время меньшее, чем метод прямого перебора ключей.

Шифры RC6, MARS – одни из претендентов на звание AES, конкурса, проводимого в 2001 году для выбора нового криптографического стандарта США. Blowfish, IDEA, CAST-128 – алгоритмы, которые используются в целом ряде продуктов криптографической защиты, в частности, в некоторых версиях PGP. Существуют многочисленные работы, посвящённые анализу этих шифров. Большинство из них используют результаты дифференциального и линейного криптоанализов – двух наиболее распространённых видов атак, применимых к блоковым шифрам в несколько урезанном виде: в исследованиях шифров зачастую не используются присущие им операции так называемого «забеливания» («pre-whitening»

и «post-whitening»), а также уменьшено количество этапов шифрования. В последнее время появился новый тип атак, основанный на исследовании статистических свойств шифров. Например, получен результат, согласно которому выходную последовательность шифра RC6 можно отличить от случайной последовательности при наличии подходящего количества выбранного шифротекста (для r-го раунда необходимо 2^{8r+10} бит текста). Таким образом, уже для 4-го раунда сложность будет значительной – 2^{74}, что делает атаку практически нереализуемой.

В данной работе предлагается новая атака, которая позволяет найти секретный ключ блокового шифра за время существенно меньшее, чем в случае линейного и дифференциального криптоанализов, и атаки, основанной на традиционном исследовании статистических свойств шифров. Например, для 5-го раунда RC6 временна́я сложность предлагаемой атаки равна 2^{46}, а в общем случае временна́я сложность, зависящая от количества атакуемых раундов, $T(R)$ определяется следующим образом: $T(R) = 2^{8\cdot\left\lceil\frac{9\cdot r-2}{7}\right\rceil}$.Предложенная атака основана на исследовании статистических свойств шифра RC6 при помощи разработанного Б.Я. Рябко статистического теста «стопка книг», который, по многочисленным исследованиям, является лучшим на текущиймоментстатистическим тестом.

В данной статье описаны исследования эффективности статистического теста «стопка книг» на примере сравнительного анализа его с другими тестами. Проведены исследования эффективности градиентной атаки на примере шифров RC6, Blowfish, MARS, IDEA, CAST-128; показаны пределы её современной практической и теоретической применимости, получены математические зависимости между эффективно взламываемыми раундами и количеством требуемых вычислительных ресурсов. Выявлены условия применимости и расширяемости атаки на произвольное количество раундов, особенности применения атаки и возможные её улучшения на конкретных примерах каждого из шифров. Кроме того, в данной работе показан метод подбора оптимальных параметров для статистического теста, их влияние на атаку; исследована временна́я сложность атаки и её зависимость от параметров теста и объёма шифротекста; показаны теоретические потребности в вычислительных мощностях, необходимых для осуществления атаки.

2. Описание блокового шифрования

У целого класса современных блоковых шифров начальный секретный ключ K длиной в $|K|$ бит преобразуется в последовательность так называемых ключей раундов $K_1, K_1, \dots, K_r$, которые используются последовательно для шифрования на разных этапах, называемых раундами шифра. В разных шифрах данные преобразования разные. Схематично процесс шифрования для такого класса шифров можно представить следующим образом. Берётся блок из входных данных фиксированной длины и подаётся на вход шифрующему преобразованию, которое преобразовывает данный входной блок в выходной. Затем берётся следующий блок той же длины из входа и подаётся на вход шифрующему преобразованию; и так, пока не зашифруем всю последовательность. Шифрующее преобразование представляет собой последовательность этапов шифрования, в которых весь процесс разбивается на несколько раундов, при этом на каждом следующем раунде происходит шифрование блока, полученного на предыдущем раунде.

$$x_1 = E(x_0, K_1), x_2 = E(x_1, K_2), \dots, x_r = E(x_{r-1}, K_r),$$

где x_0 – исходный блок данных, который необходимо зашифровать, E – операция (функция) шифрования на i-ом раунде, K_i – ключ, используемый на i-ом раунде, x_i – блок данных, являющийся «выходом» i-ого раунда и «входом» $(i+1)$-ого. В разных шифрах эти процедуры (от генерации раундовых ключей до шифрования блоков) осуществляются по-разному, причём это зависит не только от шифра, но и от значений длин блока, ключа и числа раундов r, которые для многих шифров являются параметрами. Например, для шифра RC6 длина блока может принимать значения 32, 64 или 128 бит, количество раундов может быть любым це-

лым числом, а длина ключа должна быть кратна 8 и может принимать любое значение, начиная с 8 бит. Дешифрованиепроводитсяпосхеме, обратной к шифрованию:

$$x_{r-1} = D(x_r, K_r), x_{r-2} = D(x_{r-1}, K_{r-1}), \dots, x_0 = D(x_1, K_1),$$

где используются те же ключи раундов, что и при шифровании, а операция D – обратная функция к E.

Одно из основных требований, предъявляемых к современным блоковым шифрам, можно сформулировать следующим образом.

Утверждение 1. *Любое зашифрованное сообщение должно быть «похоже» на реализацию бернуллиевского процесса, т.е. любое зашифрованное сообщение должно быть статистически неотличимо от абсолютно случайной последовательности любым статистическим тестом.*

В частности, все шифры, принимавшие участие в конкурсе AES, проверялись на выполнение этого условия. Для анализа выполнения этого требования будем использовать следующую общепризнанную схему. Определим n-битовое слово α_i, как двоичную запись числа i в виде последовательности фиксированной длины n, где i = 0, 1, 2, …, $2^n - 1$, где, как и ранее, n – длина блока рассматриваемого шифра (т.е. α_0 состоит из n-битовой цепочки нулей, α_1 – из $(n-1)$ нуля и единицы, α_2 – из $(n-2)$ нулей и т.д.). Заданная последовательность, очевидно, имеет явные закономерности и сильно отличается от случайной. От современного блокового шифра требуется, чтобы при любом значении секретного ключа последовательность n-битовых слов $E(\alpha_0)$, $E(\alpha_1)$, $E(\alpha_2)$, ..., рассматриваемая как двоичная последовательность, была статистически неотличима от случайной. Это требование, в частности, позволяет использовать блоковые шифры как генераторы псевдослучайных чисел.

3. Описание градиентной статистической атаки

3.1. Общие принципы атаки

Предлагаемый метод относится к классу атак с выбираемым шифруемым текстом (chosen ciphertext sattack) и описан в [5]. При реализации этой атаки криптоаналитик может подавать на вход шифра любой текст, анализировать полученное зашифрованное сообщение, затем, базируясь на результатах этого анализа, подавать новое сообщение и т.д. Цель атаки – нахождение (секретного) ключа, причём при этом предполагается, что криптоаналитик знает все характеристики шифра, кроме этого ключа.

Для нашей задачи на вход шифров подавались блоки α_i, длина которых в двоичной записи равна длине блока данных шифра данных. Блоки подаются в последовательности $\alpha_1, \alpha_2, \dots, \alpha_m$, которые имеют следующий вид:

$$00\dots001,\ 00..0010,\ 000..0011,\dots \quad (1)$$

Введём понятие «меры случайности» последовательности данных. Мы будем использовать термины «более» и «менее» случайные последовательности, понимая под этим, что некоторая последовательность более случайна, чем другая, если отклонения от случайности у первой достоверно выявляются при большей длине, чем у второй, или, что тоже самое, – величина статистики критерия для первой последовательности меньше, чем для второй (менее случайной). Допустим, задана последовательность α. Применим к ней статистический тест Г, вычисляющий величину статистики критерия (что фактически равно отклонению от случайности), которую обозначим как $\Gamma(\alpha)$. Тогда в качестве «меры случайности» последовательности α возьмём величину, обратную к $\Gamma(\alpha)$.

Если на вход шифра подавать данные вида (1) (что существенно, – с длиной каждого α_i, равной размеру блока шифра), то вероятность обнаружить на выходе последовательность с отклонениями от «случайности» будет значительно выше, т.к. «случайность» этой последовательности очень низкая и одновременно совпадение с размером блока уменьшает «сме-

шиваемость» элементов последовательности и даёт менее запутанную на выходе шифра последовательность. В результате экспериментов, описанных в [1], было показано, что мера случайности последовательности, полученной на выходе после i-го раунда, в любом блоковом шифре возрастает с ростом числа раундов, что полностью соответствует требованию утверждения (1). В то же время, в процессе дешифрования мера случайности (будем называть её просто «случайность») последовательности убывает.

Кратко опишем схему градиентной статистической атаки. Пусть есть i-й раунд шифра: последовательность бит $x_1, x_2, \dots, x_m$ – зашифрованная на i раундов последовательность, которая подаётся на вход операции дешифрования раунда i, и эта же последовательность на выходе дешифратора раунда i – $y_1, y_2, \dots, y_m$. Тогда схему дешифрования можно представить в виде: $y_m = D(x_m, K_i), y_{m-1} = D(x_{m-1}, K_i), \dots, y_1 = D(x_1, K_i)$. Далее, оцениваем меру случайности последовательности $y_1, y_2, \dots, y_m$. При дешифровании с правильным ключом она должна сильно уменьшиться, тогда как дешифрование с неправильным ключом аналогично шифрованию, т.е. оно, наоборот, увеличивает меру случайности последовательности. В итоге, если $\Gamma(y_1, y_2, \dots, y_m) < \Gamma(x_1, x_2, \dots, x_m)$, то ключ раунда K_i подобран правильно, иначе – неправильно. Смысл атаки сводится к нахождению такого ключа K_i, который будет минимизировать $\Gamma(y_1, y_2, \dots, y_m)$. Теоретически для этого надо перебрать все $2^{|K_i|}$ вариантов, но исходя из того что правильный ключ даёт существенно меньшую величину $\Gamma(y_1, y_2, \dots, y_m)$, нежели при неправильном ключе раунда, можно подбирать ключи до нахождения этого «существенного минимума». После нахождения ключа K_i, дешифруем последовательность на i-ом раунде и переходим к подбору ключа K_{i-1} и т.д. Нахождение всех ключей раундов K_i равносильно нахождению секретного ключа K, т.к. позволяет шифровать/дешифровать любую последовательность.

3.2. Исследование эффективности и применимости атаки

Эффективность данной атаки в сравнении с полным перебором ключей заключается в том, что мы можем находить по отдельности ключи каждого раунда. Как правило, размер ключей в раундах является постоянным и равным во всех раундах, при этом он обычно намного меньше по длине, чем исходный ключ K. Используя вышеописанную схему, мы можем подбирать ключи K_i, что даёт следующую операционную сложность атаки:

$$T = \sum_{i=1}^{r} m_i \cdot 2^{|K_i|},$$

где m_i – трудоёмкость операции дешифрования одного раунда с учётом длины входной последовательности, $|K_i|$ – длина раундового ключа на i-ом раунде, r – количество раундов шифра. При равных размерах раундовых ключей можно определить верхнюю оценку следующим образом:

$$T = r \cdot m_{\max} \cdot 2^{|K_0|},$$

где K_0 – размер раундового ключа, Операционная сложность полного переборе равна $2^{|K|}$. Таким образом, градиентная статистическая атака эффективна при выполнении следующего условия:

$$\sum_{i=1}^{m} m_i \cdot 2^{|K_i|-|K|} < 1. \quad (2)$$

В ближайшем приближении оценка (2) выглядит так:

$$r \cdot m_{\max} \cdot 2^{|K_0|-|K|} < 1.$$

Данную оценку можно переписать в сокращённом виде:

$$\log_2(m) + |K_0| < |K|. \quad (3)$$

Количество раундов в (3) не фигурирует в силу того, что, как будет показано ниже, сложность атаки на последний раунд существенно выше (на несколько порядков) сложности атаки на предыдущие раунды шифра.

Таким образом, если существует статистический тест, который эффективно находит отклонения от случайности на $(r-1)$ раунде, при рекомендованных r раундах, то ключ такого шифра можно найти существенно быстрее, чем методом прямого перебора. Это основная идея «градиентной статистической атаки».

В случае когда $|K_0| \ll |K|$, можно подбирать ключи сразу к нескольким последним раундам с соответственной сложностью

$$T = m \cdot 2^{\sum_{j=d+1}^{r}|K_i|} + \sum_{i=1}^{d} m_i \cdot 2^{|K_i|}.$$

Общее число успешно атакуемых раундов при этом равно:

$$d + \left\lfloor \frac{|K|}{|K_0|} \right\rfloor - 1,$$

где d – максимальный номер раунда, на котором статистический тест выявляет отклонения от случайности.

Исходя из схемы данного типа атаки, сразу выделяются несколько теоретических и практических ограничений применимости атаки:

1. Сильная зависимость атаки от применяемого статистического теста и его параметров. Используемый статистический тест является параметром метода. Причём в широком смысле: на разных раундах шифра можно использовать разные статистические тесты с целью уменьшения общей трудоёмкости атаки. Подходит любой тест, применимый для проверки гипотезы H_0 о том, что двоичная последовательность порождается бернуллиевским источником с равными вероятностями для нуля и единицы против альтернативной гипотезы H_1, являющейся отрицанием H_0. Использование того или иного статистического теста сильно влияет на количество раундов шифра, выход которых заданный тест отличает от абсолютно случайной последовательности.

2. Требование стационарности операции шифрования E: зависимость только от K_i и x_i. Существуют шифры, в которых функция шифрования функционально полно зависит от исходного ключа K, что делает градиентную атаку абсолютно неприменимой к таким шифрам.

3. Ограничение на размер раундовых ключей относительно исходного секретного ключа:

$$r \cdot m_{\max} \cdot 2^{|K_0|-|K|} < 1.$$

В некоторых шифрах размер раундового ключа либо вообще не отличается от размера исходного, либо отличается на слишком малую величину, что делает данный тип атаки бесполезным либо не эффективным, применительно шифрам данного типа.

Для проведения практических атак в качестве статистического теста был взят тест «стопка книг», разработанный Б.Я. Рябко, и описанный в [4], поскольку этот тест зарекомендовал себя как один из самых эффективных тестов для проверки криптографических генераторов (см. [4]).

3.3. Гибридная градиентная статистическая атака и её эффективность

У многих современных блоковых шифров при большом числе раундов даже «очень неслучайная» последовательность после шифрования статистически неотличима от случайной при любой длине и используемом тесте. Гибридная атака является некоторым расширением стандартного варианта градиентной статистической атаки с целью повысить число эффективно взламываемых раундов (т.е. взламываемым с трудоёмкостью меньшей, чем у полного перебора) у класса шифров с хорошими статистическими свойствами.

Пусть, например, шифр использует r раундов. Число d равно номеру максимального раунда шифра, на котором используемый в атаке статистический тест выявляет отклонения от случайности, причём $d < r$. Тогда градиентная атака может быть модифицирована следующим образом: для каждого набора ключей K_{d+1},..., K_r раундов $d+1$, ..., r повторяем описанную выше процедуру нахождения неизвестных ключей K_1,...,K_d раундов 1,...,d. Другими словами, ключи K_{d+1}, ..., K_r находим полным перебором, а K_1, ..., K_d – по описанному выше

методу, что при некоторых соотношениях параметров может быть меньше, чем количество операций, необходимое при прямом переборе всех ключей.

Для проведения такой комбинированной атаки потребуется количество операций, равное $m \cdot 2^{\sum_{j=d+1}^{r}|K_i|} + \sum_{i=1}^{d} m_i \cdot 2^{|K_i|}$. Как нетрудно понять исходя из данного соотношения, в случае когда $r = d + 1$, трудоёмкость гибридного варианта атаки совпадает с трудоёмкостью стандартного варианта.

Таким образом, применимость гибридного варианта градиентной статистической атаки определяется следующим условием:

$$2^{(\sum_{i=d+1}^{r} m_i)-|K|} + d \cdot 2^{|K_0|-|K|} < \frac{1}{m}. \quad (4)$$

Понятно, что гибридные атаки имеют смысл, когда размер раундового ключа $|K_0|$ много меньше размера секретного ключа $|K|$. Исходя из формулы (4) можно заключить, что гибридный вариант атаки позволяет расширить поддающиеся стандартной градиентной атаке $(d + 1)$ раундов на $\left(\left\lfloor\frac{|K|}{|K_0|}\right\rfloor - 2\right)$ раундов и, таким образом, общее количество эффективно взламываемых раундов становится равно $d + \left\lfloor\frac{|K|}{|K_0|}\right\rfloor - 1$, где d – максимальный номер раунда, на котором статистический тест выявляет отклонения от случайности.

3.4. Программная реализация атаки

Градиентная статистическая атака была реализована программно на языке C++ со следующими особенностями:

1. Возможность распараллелить подбор раундовых ключей шифра на произвольное количество потоков. Тестирование атаки и получение экспериментальных данных производилось на суперкомпьютерном кластере Новосибирского государственного университета.

2. В качестве статистического теста был использован тест «стопка книг» с одной степенью свободы, что позволяет осуществлять статистический анализ последовательностей с линейной трудоёмкостью и малыми затратами аппаратных ресурсов. Подробнее о реализации статистического теста – в разделе 4.3.

3. Создана библиотека для измерения меры случайности произвольной битовой последовательности и проведения атаки на заданный шифр. На вход требуется подавать функцию установки ключа/формирования раундовых ключей, функцию блокового шифрования и параметры шифра (размер блока, количество раундов). В итоге, данная библиотека представляет собой относительно универсальный API-интерфейс, применимый для потенциально любых блоковых шифров, удовлетворяющих ограничениям, описанным в разделе 3.2.

4. Статистический тест «стопка книг»

4.1. Эффективность теста

В [8] предложен новый статистический тест «стопка книг», эффективность которого исследовалась экспериментально. В частности, с его помощью проверялись датчики случайных чисел, приведённые в [9]. Они были исследованы автором с помощью мощного спектрального теста [10] и прошли его успешно, но тест «стопка книг» выявил отклонения от случайности в последовательностях чисел, порождённых этими датчиками. Кроме того, с помощью этого теста впервые удалось найти существенные статистические недостатки у блокового шифра MARS с уменьшенным числом раундов [11], а также генераторов псевдослучайных чисел RANDU [12] и RC4 [13]. В [7] и [14] показано, что для достаточно широкого класса альтернативных гипотез тест «стопка книг» позволяет проверять выборки уже при длине порядка $\sqrt{S}$, где S — количество слов в алфавите, из которого сделана выборка. Размер выборки становится решающим фактором, когда S очень велико, например, 2^{32} или 2^{64}. Именно с такими алфавитами порой приходится сталкиваться в современной криптографии и защите информации. Многие тесты, в частности критерий χ^2, просто неприменимы по сложности

в подобных случаях, так как размер тестируемой с их помощью выборки должен быть порядка *S*. Тест «стопка книг» является лучшим статистическим тестом среди всех тестов, рекомендованных NIST [4, 8].

4.2. Описание структуры теста

Тест «стопка книг» является критерием согласия, поэтому вначале приведём описание такого типа критериев. Пусть имеется выборка$X = (x_1, x_2, \dots, x_N)$ из алфавита $A = \{a_1, a_2, \dots, a_s\}$. Рассмотрим гипотезу H_0, которая заключается в том, что элементы выборки независимы, что означает истинность следующего суждения:

$$P(x_n = a_i) = p^0 = {}^1\!/_S, n = 1, \dots, N; i = 1, \dots, s.$$

Другими словами, элементы выборки имеют равномерное распределение, т. е. все буквы алфавита порождаются независимо и с равными вероятностями. Под критерием согласия с гипотезой H_0 понимается некоторая функция от выборки $\pi(X)$, такая что

$$\pi(X) = \begin{cases} H_0, \text{т. е. принимаем } H_0 \\ \overline{H_0}, \text{т. е. отвергаем } H_0 \end{cases}.$$

Критерии характеризуются прежде всего ошибками первого и второго рода. Ошибка первого рода – это вероятность отвергнуть гипотезу, если она верна. Ошибка второго рода – это вероятность принять гипотезу H_0, если она неверна. Величина $(1 - \alpha)$ называется уровнем значимости критерия.

Теперь опишем тест «стопка книг» с теми параметрами, которые будут использоваться в доказательстве. Перед тестированием выборки в алфавите A фиксируется произвольный порядок, который меняется после анализа каждого выборочного элементаx_n следующим образом: буква x_n получает номер 1; номера тех букв, которые были меньше номера этой буквы, увеличиваются на 1; у остальных букв номера не меняются. Формально эту процедуру можно описать так: пусть $\omega^n(a)$ – это номер буквы $a \in A$ после анализа элементов $x_1, x_2, \dots, x_{N-1}$, тогда номер буквы в алфавите определяется следующим образом:

$$\omega^{n+1}(a) = \begin{cases} 1, \text{если } x_n = a, \\ \omega^n(a) + 1, \text{если } \omega^n(a) < \omega^n(x_n), \\ \omega^n(a), \text{если } \omega^n(a) > \omega^n(x_n). \end{cases} \quad (5)$$

Такая конструкция похожа на стопку книг, если считать, что номер книги совпадает с её положением в стопке. Книга извлекается из стопки, после чтения кладётся наверх, и её номер становится первым. Книги, которые первоначально были над ней, двигаются вниз, а остальные остаются на месте.

В отличие от многих других тестов, например критерия χ^2, в «стопке книг» подсчитывается не частота встречаемости букв в выборке, а частота встречаемости номеров букв при описанном упорядочивании. Перед тестированием множество всех номеров {*1, ..., S*} разбивается на две непересекающиеся части: $A_1 = \{1, 2, \dots, M\}$ и $A_2 = \{M + 1, M + 2, \dots, S\}$. Затем по выборке $(x_1, x_2, \dots, x_N)$ подсчитывается V_0 – количество номеров $\omega^n(x_n)$, принадлежащих подмножеству A_1, т. е. количество попаданий букв в «верхнюю часть» «стопки книг». Число $(N - V_0)$, очевидно, равно количеству попаданий в «нижнюю часть» стопки. Далее вычисляется статистика по следующей формуле:

$$x^2 = \frac{(V_0 - NP_1)^2}{NP_1} + \frac{((N - V_0) - N(1 - P_1))^2}{N(1 - P_{1)}},$$

где $P_1 = |A_1|/S$. Если x^2 меньше критического уровня $\chi^2_{1,1-\alpha}$, то гипотеза H_0 принимается, иначе – отвергается. Величина $\chi^2_{1,1-\alpha}$ – квантиль распределения χ^2 уровня значимости $(1 - \alpha)$ с одной степенью свободы. Таким образом, критерий теста «стопка книг» будет выглядеть так:

$$\pi_{\text{bs}}(X) = \begin{cases} H_0, \text{если } x^2 < \chi^2_{1,1-\alpha} \\ \overline{H_0}, \text{иначе.} \end{cases}.$$

Сама величина x^2 является обратной к той мере Г, о которой говорилось в части 3. Если обобщить данный тест на случай с произвольными степенями свободы (т.е.будем делить алфавит не на 2, а на произвольное количество частей), получим меру Г, равную обратной к следующей величине x^2:

$$x^2 = \sum_{i=1}^{r} \frac{(V_i - NP_i)^2}{NP_i},$$

где r – количество непересекающихся частей A_i, на которые разбивается исходное множество номеров $\{1, 2, \dots, S\}$; $P_i = |A_i|/S$. Известно, что распределение случайной величины x^2 асимптотически приближается к распределению χ^2 с $(r-1)$ степенью свободы при выполнении H_0 (в данном случае H_0 примет более общий вид: будет включать в себя проверку количества попаданий в каждую из частей стопки). Это позволяет использовать для проверки гипотезы на случайность с произвольным количеством степеней свободы квантили заданного уровня доверия: $\chi^2_{r,1-\alpha}$ и ввести формально понятие «меры случайности» как обратной к значению статистики x^2 с нижним пределом практической применимости в атаке при области значений больших, чем $\chi^2_{r,1-\alpha}$.

4.3. Программная реализация теста и его параметризация

Применяемая в настоящей работе программная реализация теста «стопка книг» основывается на случае с одной степенью свободы. Данный вариант позволяет снизить трудоёмкость анализа и требования к расходуемым системным ресурсам. Это следует из того, что для определения части стопки, в которой находится слово, достаточно понять, находится ли оно в верхней части: если ответ «да», значит, увеличиваем счётчик верхней части; если ответ «нет», то слово находится в нижней части стопки, и счётчик остаётся прежним. Тест имеет два параметра, которые будем обозначать в виде пары (w, u), где w – это длина слова алфавита (в битах), а u – параметр размера верхней части, определяемый следующим образом: $u = \log_2 A_1$, где A_1 – размер верхней части (количество слов) стопки. Соответственно, размер всего алфавита равен 2^w, а размер верхней части равен 2^u.

Для случая с одной степенью свободы использовался следующий алгоритм:

1. На i-ом шаге имеем слово A_i фиксированной длины w.
2. Ищем его в верхней части стопки при помощи высокоскоростного алгоритма. Если результат поиска положителен, увеличиваем счётчик верхней части и ставим найденный элемент в голову списка, сдвигая оставшиеся на 1 вниз. Если результаты отрицательный, ставим текущий элемент на первое место и сдвигаем все элементы верхней части стопки вниз на 1, удаляя последний.
3. Вычисляем численную статистику x^2 и проверяем гипотезу в соответствии с квантилем заданного уровня доверия.

Данная схема соответствует равенству (5).

Особенности программной реализации данного алгоритма состоят в следующем:

1. Проверка принадлежности заданного элемента верхней части осуществляется за время, равное $O(1)$, т.е. за константное, не зависимое ни от каких параметров теста и потенциально – от параметров всей атаки, что является большим плюсом.
   Такое время удалось достичь за счёт использования алгоритма поиска с хэш-индексацией, который описан ниже.
2. Распараллелить данный алгоритм в силу конструктивных особенностей самого теста «стопка книг» невозможно (в отличие, к примеру, от частотного теста χ^2), но можно распараллелить саму атаку (подбирать параллельно несколько ключей), что и было сделано при реализации атаки на суперкомпьютере.

3. В памяти хранится только верхняя часть стопки, оптимальный размер которой в соответствии с исследованиями, приведёнными в [14], обычно равен $\sqrt{S}$, т.е. $u = w/2$, что понижает хранимый объём данных с 2^w до $2^{w/2}$ и к примеру, при размере слова 32 бита уменьшает объём хранимой информации в 2^{16} раз, что приводит к существенной – по сравнению с другими тестами – экономии аппаратных ресурсов.

Непосредственно верхняя часть стопки хранится в памяти в виде двусвязного динамического списка, благодаря чему операции перестановки текущего элемента на первое место со сдвигом всех последующих осуществляется всего за 2 операции.

Алгоритм поиска элемента в стопке заключается в следующем. Как было сказано выше, верхняя часть стопки хранится в виде двусвязного списка. Требуется проверить наличие искомого элемента в заданном списке. К списку добавлен механизм хэш-таблиц, заключающийся в следующем. Есть ограниченное множество допустимых значений хэш-функции H, принимающей на вход слово из алфавита теста и состоящей из упорядоченного ряда натуральных чисел фиксированной мощности, равной $|H|$ (параметр хэш-таблицы). Число $2^u/|H|$ называется коэффициентом заполнения хэш-таблицы. Также есть массив указателей, хранящий адреса разных элементов списка. При этом элементы с одинаковым хэш-номером объединяются в один хэш-класс. Помимо стандартных связей, отображающих положение всех элементов в стопке, элементы хэш-класса имеют связи между собой: хэш-массив содержит адрес первого элемента хэш-класса (он может быть любым), а тот указывает на следующий элемент того же класса и так далее. В результате этого все элементы хэш-класса объединяются в ещё один – второстепенный – список. Встречая элемент a, мы определяем его хэш-номер, по нему – через хэш-таблицу – адрес первого элемента его хэш-класса, далее идём по списку хэш-класса и смотрим наличие заданного элемента в списке. Таким образом, операционная сложность поиска сводится к количеству операций, равному коэффициенту заполнения хэш-таблицы. Понятно, что чем выше коэффициент заполнения, тем меньше пространственная трудоёмкость, но тем больше временна́я. Тем не менее, временна́я трудоёмкость операций поиска, добавления, перемещения элементов в любом случае остаётся равной $O(1)$.

4.4. Анализ временно́й сложности теста

Проведём практические оценки времени анализа последовательности статистическим тестом «стопка книг». В табл. 1 приведены результаты анализа выборки длиной 2^{32} бит (512 Мб) в зависимости от параметров теста «стопка книг». При этом используются следующие обозначения: w – размер слова в алфавите теста; u – параметр, выражающий размер верхней части стопки, как 2^u.

Таблица 1. Временной анализ теста «стопка книг»

№ теста	Размер файла выборки, Мб	Параметры теста (w, u)	Время анализа, сек
1	512	(32, 24)	22.8
2	512	(32, 20)	19.2
3	512	(32, 16)	6.6
4	512	(32, 12)	5.8
5	512	(32, 8)	5.1
6	512	(24, 20)	31.3
7	512	(24, 16)	8.2
8	512	(24, 12)	6.9
9	512	(24, 8)	5.9
10	512	(16, 15)	9.2
11	512	(16, 12)	8.9
12	512	(16, 8)	7.5
13	512	(8, 7)	12.5
14	512	(8, 4)	12.3

Исходя из выше приведённых данных, проведём эмпирическуюаппроксимациию зависимости времени анализа последовательности от параметров теста:

$$T(V, w, u) = C \cdot V \sqrt{\frac{u}{w}},$$

где C – константа, зависящая от аппаратных параметров тестируемой системы: частота процессора, время чтения информации с жёсткого диска, используемая операционная система.

5. Анализ градиентной статистической атаки на шифр RC6

5.1. Описание структуры шифра

RC6 – симметричный блочный криптографический алгоритм, производный от алгоритма RC5. Вариант шифра RC6, заявленный на конкурс AES, поддерживает блоки длиной 128 бит и ключи длиной 128, 192 и 256 бит, но сам алгоритм, как и RC5, может быть сконфигурирован для поддержки более широкого диапазона длин как блоков, так и ключей (от 0 до 2040 бит). Шифр RC6 является финалистом AES и используется во многих современных пакетных криптосистемах (например, PGP и GPG), поэтому его криптоанализ представляет практический интерес.

Для спецификации алгоритма с конкретными параметрами принято обозначение RC6-w/r/b, где

- w – длина блока в битах;
- r – число раундов;
- b – длина ключа в битах. Возможные значения: 0...255 байт.

В данной работе рассматривается самый распространённый вариант шифра RC6-128/r/128-256, когда на вход подаются 4 блока по 32 бита, а ключ может принимать 3 длины: 128, 192 и 256 бит.

На начальном этапе секретный 128 (192, 256)-битный секретный ключ преобразуется в массив размера $(2r + 4)$ 32-битных раундовых ключа $S[0. \ldots, 2r + 3]$. Затем, ко второму и четвёртому блоку прибавляются раундовый ключи S_0 и S_1. Тоже самое происходит и в самом конце шифрования. Данные операции называются операциями «забеливания». Далее каждый блок делится на четыре 32-битных блока(A_i, B_i, C_i, D_i), которые потом шифруются на каждом раунде по схеме сети Фейстеля с использованием функции раундового шифрования F. Функция F применяется к двум 32-битным блокам из четырёх и имеет следующий вид: $F(A, S) = f(A) + S$, где S – раундовый ключ.

Подробное описание шифра RC6 можно найти в [6] и [19].

5.2. Атака на RC6

В качестве статистического теста для исследования всех раундов шифра RC6 был взят тест «стопка книг» с одной степенью свободы. Параметры теста будем обозначать в соответствии с приведёнными ранее обозначениями. Все практические результаты будем проводить для квантилей двух уровней доверия: 0.95 и 0.99.

В ходе работы была проверена устойчивость шифра RC6 по отношению к градиентной атаке. Были рассмотрены 128-битный и 64-битный режимы RC6 со 128-битным ключом (в разделах 5.2, 5.3 – без операций «забеливания», в разделе 5.4 – модификация для RC6 с операциями «забеливания»). Как видно из описания алгоритма в 5.1, шифр RC6 устроен таким образом, что он зашифровывает 128-битное число в два этапа. Сначала первые 32 бита шифруются с помощью первого ключа, так называемого полураунда, потом следующие 32 бита, используя другой ключ полураунда, оставшиеся 64 бита на данном раунде не меняются (они шифруются на следующем раунде). Исходя из этого, можно проводить атаку не на весь раунд, а на каждый полураунд в отдельности, что значительно сокращает трудоёмкость атаки.

В табл. 2 приведены результаты тестирования шифра с помощью «стопки книг». Было проведено 100 тестов со 100 различными случайно подобранными ключами.

Таблица 2. Предварительный статистический анализ RC6

Раунд	Размер выборки	Параметры теста	$> Q_{0.95}$	$< Q_{0.05}$	$E(x^2)$
3	2^{18}	(24, 10)	100	0	$6.05 \cdot 10^7$
4	2^{18}	(24, 10)	100	0	14775.6
5	2^{18}	(32, 22)	16	14	1.72
5	2^{29}	(32, 22)	87	81	157.14

Показано, что отклонения от случайности в 16% случаев могут быть зафиксированы на 5-ом раунде. Это говорит о том, что градиентная атака может эффективно применяться на 6-ой раунд шифра при размере выборки 2^{18}. Но об эффективности применения атаки на все 100% можно говорить лишь в случае 5 раундов, т.к. на 5-ом раунде фиксация отклонений от случайности происходит лишь в 16% случаев, а средняя величина x^2 ощутимо меньше квантиля уровня доверия 0.95. Так же по тестам на раунде 5 хорошо видно, что 14% ключей дают значения x^2 меньше квантиля уровня доверия 0.05, что говорит о высоком уровне случайности выходной последовательности. Фактически можно предположить, что существует определённый класс «слабых» ключей, для которых данная атака может эффективно применяться на 6 раундов, а есть «сильные» ключи, для которых атака эффективна лишь на 5 раундов.

Было проведено исследование зависимости эффективности рассматриваемой атаки от размера выборки и от параметров теста. Результаты исследования представлены в нижеследующей таблице.

Таблица 3. Оптимизация параметров атаки

№ теста	Раунд	Размер выборки	Параметры теста	$> Q_{0.95}$	$E(x^2)$
1	4	2^{14}	(8, 7)	50	133.4
2	4	2^{14}	(8, 6)	50	210.9
3	4	2^{14}	(16, 15)	100	1265.0
4	**4**	$\mathbf{2^{14}}$	**(16, 10)**	**100**	**18484.9**
5	4	2^{18}	(8, 7)	100	1653.5
6	4	2^{18}	(8, 6)	78	1204.7
7	4	2^{18}	(16, 15)	100	21142.5
8	**4**	$\mathbf{2^{18}}$	**(16, 8)**	**100**	**118515.0**
9	4	2^{18}	(24, 20)	65	170.4
10	4	2^{18}	(24, 10)	100	14775.7
11	4	2^{18}	(32, 24)	70	36.3
12	4	2^{18}	(32, 16)	11	1.02
13	**4**	$\mathbf{2^{22}}$	**(16, 8)**	**100**	**123552**
14	4	2^{22}	(24, 10)	100	83179.5
15	4	2^{22}	(32, 18)	11	1.68
16	4	2^{22}	(32, 24)	68	51.2

Из данной таблицы видно, что на каждом отдельном файле заданного размера есть свой оптимальный набор параметров. Этот набор существенно зависит от размера файла. Чем больше размер, тем больше должна быть длина слова в тесте. Сверху длина слова ограничена условием применимости критерия χ^2: $S \cdot 2^{w-u} \geq 5$ (*S*– размер выборки в блоках), но даже при выполнении этого условия, но при слишком большом значении *w* тест даёт плохие результаты. Точно так же и при слишком маленьком значении *w* или слишком маленькой

длине слова u, тест плохо распознаёт случайность последовательности. Исходя из представленных данных, оптимальная длина слова w равна в среднем:

$$w_{\text{opt}} = 8 \cdot \left\lfloor \frac{\log_2(L)}{8} \right\rfloor, \tag{6}$$

где L – длина входной последовательности в битах ($L = S \cdot 2^b$). В таблице (3) жирным шрифтом выделены оптимальные параметры для каждого из размеров последовательности. Оптимальный размер верхней части варьируется и точно его определить нельзя, но в среднем оптимум u находится следующим образом:

$$u_{\text{opt}} = \frac{w}{2} \pm 10\%. \tag{7}$$

Кроме того, задача подбора пары оптимальных параметров для теста является очень существенной, что ярко видно на примере тестов 14 и 15, а также тестов 8 и 12. При оптимально и правильно подобранных под конкретный размер выборки (не превышающей 2^{22}) параметрах теста шифр RC6 может быть взломан до 5–6 раундов, а при неправильно подобранных параметрах это число снижается до 3–4 раундов, а в некоторых случаях даже ниже.

Рассмотрим зависимость времени анализа выборки от её размера и параметров теста. В нижеследующей таблице приведены данные для 4-го раунда теста и 100 случайно выбранных ключей.

Таблица 4. Временно́й анализ при различных параметрах атаки на 4 раунд RC6

№ теста	Размер выборки	Параметры теста	$E(x^2)$	Время,сек
1	2^{14}	(16, 8)	11752.6	0.14
2	2^{18}	(16, 8)	191344.2	1.6
3	2^{22}	(16, 8)	1364622.9	24.5
4	2^{20}	(24, 23)	693.4	7.2
5	2^{20}	(24, 16)	6008.0	5.7
6	2^{20}	(24, 12)	24056.4	5.1

В табл. 6 хорошо видна строго линейная зависимость размера выборки от времени анализа файла тестом. Кроме того, данные результаты говорят о том, что с ростом размера выборки растёт величина $E(x^2)$, причём растёт она, как и время, линейно от размера файла, что существенно. Таким образом, можно сделать предположение о возможности повышения числа взламываемых данным статистическим тестом раундов (число раундов, в которых тест распознаёт отклонения от случайности) с ростом размера файлов. Вполне понятно, что с ростом размера файлов будет расти и оптимальный размер слова, что требует дополнительного и существенного расхода памяти. Тесты 4–6 в табл. 4 подтверждают идею оптимального объёма верхней части, как корень квадратный из объёма всего алфавита теста: в случае верхней части, равной по объёму корню квадратному из u, значение $E(x^2)$ в 34.7 раза больше. Кроме того, чем меньше верхняя часть, тем быстрее происходит анализ выборки. Соответственно, кроме правильного подбора параметров теста, для взлома шифра требуется определить и оптимальный размер выборки, которая нужна для взлома заданного количества раундов, так как при слишком большой выборке время анализа выборки, а соответственно, и каждого подбираемого ключа, становится слишком большим.

В работе [2] было проведено исследование данной атаки в 64-битном режиме шифра RC6. Полученные результаты представлены в табл. 5.

Таблица 5. Атака на 64-битный вариант RC6

Раунд	Число тестов	$> Q_{0.99}$	Объём выборки
3	100	100	2^9
4	100	100	2^9
5	100	100	2^{18}
6	20	16	2^{29}

Из приведённых данных хорошо видно, что в этом режиме количество раундов, в которых статистический тест «стопка книг» выявляет отклонения от случайности, в среднем выше на 1, т.е. в 64-битном варианте RC6 число взламываемых раундов равно 6 – 7. Таким образом, становится ясно, что эффективность атаки существенно зависит от размера блока в шифре: чем он больше, тем менее эффективна атака.

5.3. Программная реализация и оценка эффективности атаки

До сих пор в рассмотрение попадала работа шифра RC6 на выборках длиной до 2^{22}. Из табл. 3 явно видно, что эффективность работы теста, а соответственно, и всей атаки существенно зависит от объёма выборки: при объёме выборки 2^{10} число взламываемых раундов равно уже 4 (для 4-го раунда $Q_{0.95}$ = 25). Исходя из этого можно сделать следующее предположение.

Утверждение 2. *Для повышения числа взламываемых раундов на 1, надо увеличивать выборку в 2^8 раз.*

Тем не менее, возникает законный вопрос: можно ли взламывать градиентной статистической атакой 128-битный шифр RC6 на большем, чем 5 – 6, числе раундов? Ответ на данный вопрос содержится в исследованиях Американского института стандартов NIST. С применением менее эффективных, чем «стопка книг», статистических тестов, они получили следующие данные касательно взлома RC6 при помощи градиентной статистической атаки.

Таблица 6. Аппроксимация трудоёмкости атаки для старших раундов RC6

Раунд	Размер ключа	Объём выборки	Расход памяти
12	128	2^{94}	2^{42}
14	128	2^{118}	2^{112}
14	192	2^{110}	2^{42}
14	256	2^{108}	2^{74}
15	256	2^{119}	2^{138}

Данные табл. 6 можно интерпретировать следующим образом. Число взламываемых раундов 128-битного шифра RC6 на домашнем компьютере составляет 5–6. В общем случае, можно взламывать потенциально весь шифр, т.е. все 20 раундов, но для этого потребуется на данный момент технически недостижимый объём памяти. Несмотря на то, что в исследовании, результаты которого представлены в табл. 6, принимал участие статистический тест, по своим показателям уступающим «стопке книг», тем не менее исходя из данных, полученных на «стопке книг», можно сделать вывод о том, что на высоких раундах разница со стандартными статистическими тестами не очень существенная. Тем не менее, для точного ответа на данный вопрос требуется провести исследование рассматриваемой атаки на суперкомпьютере, после чего сравнить результаты с другими тестами.

Ответим теперь на вопросы:

– какое количество раундов можно взломать на современном этапе развития вычислительной техники?

– какова интерполяционная зависимость числа раундов от потребностей в памяти, если учесть, что данные табл. 6 соответствуют тесту «стопка книг»?

Будем называть шифр с n раундами эффективно взламываемым градиентной статистической атакой, если с применением этой атаки количество операций по взлому шифра будет существенно меньше полного перебора, т.е. если для неё выполнено условие (2) или (3).

Так как раунды шифра RC6 можно разбить на полураунды, то реализуя на нём рассматриваемую атаку, можно подбирать ключи для каждого из полураундов в отдельности. В 128-битном варианте шифра RC6 каждый ключ полураунда имеет длину 32 бита, т.е. для подбора ключа к раунду r требуется в худшем случае $m_r \cdot 2^{32}$ операций, где m_r – размер выборки, подаваемой на вход дешифратора r-го раунда. При этом стоит отметить, что подбирая ключи к полураундам, следует анализировать не всю выходную последовательность, а только чётные/нечётные блоки, так как за операцию одного полураунда меняется только половина последовательности (чётные/нечётные блоки) и анализировать остальные смысла нет.

Исходя из утверждения 2 и данных табл. 6, можно сделать вывод о том, что объём выборки, при которой статистический тест «стопка книг» может успешно применяться при реализации градиентной атаки, действительно равен

$$L(R) \approx 2^{8\cdot\left\lceil\frac{9\cdot R-30}{7}\right\rceil}. \tag{8}$$

Число R при этом может быть потенциально любым. Отсюда легко сделать оценку для числа операций:

$$T(R) = 2^{8\cdot\left\lceil\frac{9\cdot R-30}{7}\right\rceil}\cdot 2^{32} = 2^{8\cdot\left\lceil\frac{9\cdot R-2}{7}\right\rceil}.$$

Следует заметить, что вышеприведённая оценка справедлива лишь для взлома самого старшего раунда шифра, тогда как атака должна применяться на каждом раунде, начиная со старшего к младшему. Тем не менее, эту оценку можно применять и для всей атаки, а не только для старшего раунда. Поясним почему. Из оценки трудоёмкости в разделе 3.2 видно, что число операций линейно зависит от размера выборки m_i. Одновременно из формулы (5) следует, что на каждом следующем – более младшем – раунде объём выборки для взлома уменьшится в $\simeq 2^{10}$ раз. Таким образом, уже второй – следующий – член суммы ряда формулы сложности атаки в разделе 3.2 на ~3 порядка меньше, т.е. его вклад в сумму слишком мал. Расход памяти в байтах будет равен $2^{u+\log_2(w)} = w\cdot 2^u$, где$(u, w)$ – параметры теста. Такая оценка памяти подтвердилась экспериментально и является точной. Из (6), (7) и (8) следует, что $w_{\text{opt}}(R) = 8\cdot\left\lceil\frac{9\cdot R-30}{7}\right\rceil$ и из (5) получаем трудоёмкость по памяти:

$$\text{П}(R) = 8\cdot\left\lceil\frac{9\cdot R-30}{7}\right\rceil\cdot 2^{4\cdot\left\lceil\frac{9\cdot R-30}{7}\right\rceil}.$$

По приведённым формулам и по данным таблиц 3 и 6 составим табл. 7.

Таблица 7. Аппроксимация трудоёмкости атаки для старших раундов RC6

Раунды	Количество операций	Расход памяти (ОЗУ)	Расход памяти (ПЗУ)
5	2^{50}	2^{19}	2^{24}
7	2^{68}	2^{25}	2^{28}
9	2^{87}	2^{35}	2^{44}
12	2^{126}	2^{52}	2^{98}
14	2^{142}	2^{66}	2^{114}
15	2^{151}	2^{70}	2^{123}
20	2^{204}	2^{70}	2^{176}

5.4. Модификация атаки для варианта RC6 с операциями «забеливания»

Методы градиентной атаки, предложенные выше, рассматривались в применении к шифру RC6 без операций пред- и постзабеливания, то есть фактически – в отношении неполного варианта шифра. Рассмотрим модифицированную версию атаки, которую можно применять к полному варианту шифра RC6 [3].

Процесс шифрования блока (A_i, B_i, C_i, D_i), описанного части 6.1, можно представить следующим образом (приведём пример для нечётного случая):

$$B_{i+1} = G(C_i, B_i) + \mathrm{S}_{2i+1}; D_{i+1} = G(A_i, D_i) + S_{2i}, A_{i+1} = C_i; C_{i+1} = D_i. \quad (9)$$

Отсюда видно, что раундовые ключи в функции шифрования G не участвуют, а только прибавляются в виде обыкновенных слагаемых к конечному результату. При этом преобразование G является обратимым и не требует для обращения никаких ключей. Таким образом, если подать на вход i-го полураунда последовательность, к которой применено обратное преобразование, т.е. преобразование вида: $B_i^{'} = G^{-1}(C_i, B_i); D_i^{'} = G^{-1}(A_i, D_i)$, то на выходе $(i+1)$-го полураунда получим последовательность стандартного входа в i-ый раунд плюс подключи $(i+1)$-го полураунда, которые сами по себе оказывают слабое влияние на статистические свойства последовательности. Применим данную схему для первого полураунда шифра RC6, а на вход будем подавать последовательность вида:

$$\begin{gathered} B_0 = a_n - S_0; D_0 = c_n - S_1; \\ A_0 = \big(d_n \ggg (F(c_n) \bmod 32)\big) \oplus F(a_n);\ C_0 = \big(b_n \lll (F(a_n) \bmod 32)\big) \oplus F(c_n), \end{gathered} \quad (10)$$

где (a_n, b_n, c_n, d_n)– последовательность вида (1), а S_0, S_1 – ключи пред-забеливания. Тогда на выходе 1-го раунда получим последовательность $(a_n, b_n + S_3, c_n, d_n + S_2)$, что ведёт к тому, что на выходе r-го раунда мы получим последовательность, статистически равную последовательности на выходе $(r-1)$-го раунда.

Модифицированная версия атаки сводится к тому, что на вход мы подаём последовательность вида не (1), а вида (10), что ведёт к фактическому поступлению на вход второго раунда последовательности вида (1) и сокращению раундов шифрования на единицу. При этом, если в последовательности (10) правильно подобраны подключи (S_0, S_1), то результат должен давать статистику по $(r-1)$ раунду; если же они подобраны неверно, то результат должен давать статистику (значение x^2), равную r раундам шифра. После подбора подключей (S_0, S_1) подбираем по тому же принципу подключи (S_2, S_3), и так далее. Таким образом, мы можем подбирать подключи, начиная с начала, а не с конца, как это было в традиционном варианте атаки.

Понятно, что если подбирать ключи сразу парой, то сложность перебора составит $CM_r \cdot 2^{64}$, т.к. все подключи имеют длину в 32 бита, поэтому можно модифицировать текущий вариант так, чтобы подключи подбирались раздельно. Для этого просто будем подбирать вначале первый подключ: смотреть, на каком из них значение x^2 максимально, затем – второй по тому же принципу. Как показывают полученные результаты (табл.8 и 9), данный подход имеет практическую эффективность.

Итак, модифицированная атака на RC6 со сложностью перебора $CM_r \cdot 2^{32}$:

1. Для каждого S_i вычисляем последовательность вида (10) для i-го раунда при константном S_{i+1} и прибавляем к ней последовательность, полученную после шифрования на i раундов, подключи которых уже найдены.
2. Подаём полученную последовательность на вход шифру и вычисляем значение статистики x^2.
3. Выбираем S_i, для которого значение x^2 максимально (является существенно бóльшим, чем для класса неправильных подключей).
4. Фиксируем правильный S_i и возвращаемся на шаг 1 для подбора S_{i+1}.
5. После подбора обоих подключей переходим на раунд $(i+1)$.

Приведём результаты экспериментов. В табл. 8 приведены статистические результаты модифицированной версии атаки с использованием правильных и неправильных подключей последнего раунда.

При этом брались пары правильных подключей (S_0, S_1) и для каждой пары проверялись 100 неправильных пар подключей. В первой колонке представлены номера раундов.

Таблица 8. Результаты модифицированной версии атаки на первые 5 раундов

№ теста	Размер выборки	Минимальное x^2 (прав. ключи)	Максимальное x^2 (неправ. ключи)	$E(x^2)$ (неправ. ключи)	$> Q_{0.99}$ (неправ. ключи)
1	2^{14}	49.64	12.76	2.63	9
2	2^{16}	218.07	12.39	2.89	9
3	2^{18}	940.99	18.20	3.15	28
4	2^{20}	3621.84	20.69	3.06	40
5	2^{22}	15360.21	25.42	3.37	68

Основные выводы:

- Разница между случайностью при правильно подобранных и неправильно подобранных ключах раунда растёт линейно от размера выборки.
- При достаточном размере выборки значимость уровня доверия квантиля отпадает (ищем максимальное x^2).

Далее – в табл. 9 – приведены результаты модифицированной атаки на 5-ый раунд шифра. При этом было взято 5 подключей (S_0, S_1), в которых S_0 был правильным (S_1 – произвольный), и для каждой пары проверено 100 неправильных пар подключей.

Таблица 9. Результаты модифицированной версии атаки на 5 раунд

№ теста	Размер выборки	Минимальное x^2 (прав. ключи)	Максимальное x^2 (неправ. ключи)	$E(x^2)$ (неправ. ключи)
1	2^{20}	27930	3.51	1.51
2	2^{20}	2127	80.93	3.38
3	2^{20}	1891	17.06	2.31
4	2^{20}	2776	16.87	1.64
5	2^{20}	1821	6.72	2.36

Ключи пост-забеливания можно найти после подбора всех раундовых ключей и ключей пред-забеливания простым раздельным перебором до получения исходной последовательности вида (1) на минимальном количестве блоков шифра (например, на 4 блоках).

5.5. Модификация атаки с уменьшением длины ключа раунда

Из вида функции шифрования RC6 (9) видно, что подключи S_i и S_{i+1} участвуют только в явном виде, как слагаемые, изменяющие часть всего блока. Соответственно, если бóльшая часть ключа полураунда (а не весь ключ целиком) подобрана правильно, итоговая последовательность должна стать менее случайной, но её мера случайности должна быть больше, чем при полностью правильно подобранном ключе полураунда. Этот факт позволяет разработать следующую модификацию атаки: вначале подбираем ключ полураунда, перебирая только первые $(32 - p)$ бит, где p – количество урезанных бит, начиная с конца, – параметр метода, находим часть ключа с максимальным значением x^2, далее подбираем оставшиеся p бит, максимизирующие x^2. Если же все промежуточные значения x^2 оказываются меньше исходного (или не превышают квантиль заданной значимости), то подбираем подключ по традиционному алгоритму. Сложность метода тогда будем считать, как матожидание от традиционной трудоёмкости:

$$T'(T_0) = p \cdot \frac{T}{2^p} + (1 - p) \cdot (T + \frac{T}{2^p}), \tag{11}$$

где p – вероятность успеха модифицированного варианта атаки.

В табл. 10 представлены результаты атаки заданным методом. Было произведено шифрование последовательности размера 2^{28} элементов (т.е. 2^{32} бит для длины слова = 16 бит) на 5 раундов и далее, для 100 различных ключей было произведено 100 попыток найти первый ключ полураунда. В таблице представлен результат до дешифрования ($E(x^2)$ 1), промежуточный (после подбора части подключа: $E(x^2)$ 2) и конечный: после подбора всего подключа. Также приведено количество успешных попыток подбора подключа таким способом. Использованные параметры статистического теста: (24, 16).

Таблица 10. Атака с урезанными битами подключа

Количество урезанных бит	$E(x^2)$ 1	$E(x^2)$ 2	$E(x^2)$ 3	Монотонных последовательностей
2 из 32	10.35	283339	479478	88
4 из 32	10.35	172447	479478	79

Исходя из (11) получаем трудоёмкости модифицированного варианта атаки:

$$T_1 = 0.1 \cdot \left(T + \frac{T}{4}\right) + 0.9 \cdot \frac{T}{4} = 0.35 \cdot T;$$

$$T_2 = 0.2 \cdot \left(T + \frac{T}{16}\right) + 0.8 \cdot \frac{T}{16} = 0.26 \cdot T.$$

В общем итоге, видно, что вариант с 4 урезанными битами – лучший и снижает сложность атаки на любой раунд шифра в 4 раза. В частности, сложность атаки на 5 раунд шифра может быть уменьшена с 2^{50} до 2^{48}.

5.6. Итоги

Подведём основные итоги анализа эффективности градиентной статистической атаки на шифр RC6:

- Найти ключи пред- и постзабеливания можно с трудоёмкостью = T(последнего раунда) в терминах O-символики, т.е. с трудоёмкостью традиционной градиентной атаки: $O(M_r \cdot 2^{32})$, хотя точная трудоёмкость атаки возрастает из-за лишних операций обращения последовательности.
- Наличие операций пред- и постзабеливания не усложняет атаку по сравнению с традиционным вариантом, если использовать модификацию, описанную в части 6.4.
- Модификации атаки, описанные в частях 6.4 и 6.5, можно совместить, что снизит трудоёмкость традиционного варианта атаки в 4 раза, позволив при том находить ключи забеливания.
- Полученные результаты можно интерпретировать следующим образом. При доступных на текущий момент вычислительных мощностях градиентная статистическая атака с применением теста «стопка книг» может быть применена к 9-раундовому шифру RC6, для этого потребуется: 32 Гб ОЗУ, 32 Тб ПЗУ и 2^{87} операций перебора. Тогда как в случае атаки полным перебором понадобится 2^{128} операций. Начиная с 13 раунда, применять градиентную статистическую атаку имеет смысл лишь в случае использования 192-/256-битных ключей – это следует из (3). Для 19–20 раундов – лишь в случае 256-битных ключей. Кроме того, можно интерпретировать эти результаты по-другому: при 128-битном ключе и доступных ресурсах 32 Гб ОЗУ, 32 Тб ПЗУ, количество эффективно взламываемых градиентной атакой раундов равно 10 (87+32 = 119 < 128). При 192-битном ключе число эффективно взламываемых раундов увеличивается до 12 (87 + 32·3 = 183 < 192). А в случае 256-битного ключа это число равно 14 (87 + 5·32 = 247 < 256).

6. Анализ градиентной статистической атаки на шифр MARS

6.1. Описание структуры шифра

Шифр MARS является 128-битным блоковым шифром с длиной исходного ключа от 128 до 448 бит. Он использует 32 раунда шифрования, из них раундовые ключи участвуют в 16. Размер двухсоставного раундового ключа составляет 64 бита (два подключа по 32 бита).

Структуру алгоритма шифрования MARS можно описать следующим образом:

1. Происходит предварительное наложение ключа: на 32-битные подблоки A, B, C, D накладываются 4 фрагмента расширенного ключа $k_0, \dots, k_3$ операцией сложения по модулю 2^{32};
2. Выполняются 8 раундов прямого перемешивания (без участия ключа шифрования);
3. Выполняются 8 раундовпрямогокриптопреобразования;
4. Выполняются 8 раундовобратногокриптопреобразования;
5. Выполняются 8 раундов обратного перемешивания без участия ключа шифрования;
6. Финальное наложение фрагментов расширенного ключа $k_{36}, \dots, k_{39}$ операцией вычитания по модулю 2^{32}.

Процесс дешифрования аналогичен шифрованию в обратном порядке.

Важно отметить, что раунды прямого и обратного перемешивания не используют какой-либо секретной (ключевой) информации: они выполняют только роль статического перемешивания последовательности. Подробное описание шифра MARS и его раундов можно найти в [6] и [20].

6.2. Атака на MARS

В качестве статистического теста для реализации атаки так же, как и в случае RC6, был взят тест «стопка книг» с одной степенью свободы. Параметры реализации и обозначения атаки аналогичны приведённым для случая атаки на RC6.

В ходе работы была проверена устойчивость шифра MARS по отношению к градиентной атаке. Был рассмотрен 128-битный режим MARSco 128-битным ключом без операций пред-/постзабеливания.

Параметры статистического теста определялись экспериментальным путём для каждого типа раундов и размера выборки в отдельности. Результаты данных экспериментов приведены в табл. 11, где жирным шрифтом выделены оптимальные параметры.

Таблица 11. Результаты атаки на MARS

№ теста	Раунд	Размер выборки	Тип раунда	Параметры теста (w, u)	$E(x^2)$
1	5	2^{26}	FC	(24, 20)	1.07
2	**5**	$\mathbf{2^{26}}$	**FC**	**(24, 16)**	**1.11**
3	5	2^{26}	FC	(24, 12)	0.65
4	5	2^{26}	FC	(32, 24)	0.21
5	5	2^{26}	FC	(32, 20)	0.66
6	5	2^{26}	FC	(32, 16)	0.82
7	5	2^{26}	FC	(16, 12)	0.42
8	5	2^{26}	FC	(16, 10)	0.12
9	5	2^{24}	BC	(16, 8)	0.44
10	5	2^{24}	BC	(24, 20)	7.12
11	5	2^{24}	BC	(24, 16)	0.098
12	5	2^{24}	BC	(24, 12)	0.53
13	**5**	$\mathbf{2^{24}}$	**BC**	**(32, 24)**	**13.7**
14	5	2^{24}	BC	(32, 20)	0.18

15	5	2^{24}	BC	(32, 16)	0.97
16	5	2^{24}	BC	(16, 12)	4.87
17	5	2^{24}	BC	(16, 10)	1.53
18	5	2^{24}	BC	(16, 8)	0.44
19	6	2^{26}	FM	(24, 20)	3.63
20	6	2^{26}	FM	(24, 16)	5.43
21	6	2^{26}	FM	(24, 12)	4.67
22	**6**	$\mathbf{2^{26}}$	**FM**	**(32, 24)**	**836.24**
23	6	2^{26}	FM	(32, 20)	58.64
24	6	2^{26}	FM	(32, 16)	1.40
25	6	2^{26}	FM	(16, 12)	0.47
26	6	2^{26}	FM	(16, 10)	3.02
27	6	2^{26}	FM	(16, 8)	2.08
28	7	2^{26}	BM	(24, 20)	1.05
29	7	2^{26}	BM	(24, 16)	0.73
30	7	2^{26}	BM	(24, 12)	0.53
31	**7**	$\mathbf{2^{26}}$	**BM**	**(32, 24)**	**3810.06**
32	7	2^{26}	BM	(32, 20)	416.26
33	7	2^{26}	BM	(32, 16)	3.80
34	7	2^{26}	BM	(16, 12)	1.56
35	7	2^{26}	BM	(16, 10)	0.73
36	7	2^{26}	BM	(16, 8)	1.34

FM/BM – прямое/обратное перемешивание; FC/BC – прямой/обратный режим ядра.

Далее была проведена оптимизация по объёму выборки, результаты которой представлены в табл. 12.

Таблица 12. Оптимизация атаки на MARS по объёму выборки

№ теста	Раунд	Размер выборки	Тип раунда	Параметры теста (w, u)	$E(x^2)$
1	5	2^{16}	FM	(16, 8)	0.56
2	5	2^{20}	FM	(24, 16)	0.88
3	5	2^{23}	FM	(32, 24)	$5.11 \cdot 10^3$
4	5	2^{26}	FM	(32, 24)	$3.94 \cdot 10^7$
5	4	2^{16}	FC	(16, 8)	0.11
6	4	2^{20}	FC	(24, 20)	0.20
7	4	2^{23}	FC	(24, 20)	$3.49 \cdot 10^3$
8	4	2^{26}	FC	(24, 20)	$1.11 \cdot 10^5$
9	5	2^{16}	BC	(16, 8)	0.38
10	5	2^{20}	BC	(24, 16)	2.50
11	5	2^{23}	BC	(32, 24)	122.69
12	5	2^{26}	BC	(32, 24)	14.05
13	7	2^{16}	BM	(16, 8)	0.07
14	7	2^{20}	BM	(24, 16)	0.99
15	7	2^{23}	BM	(32, 24)	$6.21 \cdot 10^4$
16	7	2^{26}	BM	(32, 24)	$2.56 \cdot 10^7$

Затем для каждого случая было взято 100 случайных ключей и проверены статистические отклонения. Кроме того, реализована непосредственно сама атака на ядро шифра. Итоговые результаты по проведённым атакам представлены в табл. 13.

Таблица 13. Итоговые результаты атаки на MARS

Раунд	$> Q_{0.95}$	$> Q_{0.99}$	$E(x^2)$	Объём выборки
Прямое перемешивание				
1	100	100	$1.44 \cdot 10^8$	2^{23}
2	100	100	$1.22 \cdot 10^8$	2^{23}
3	100	100	$6.63 \cdot 10^7$	2^{23}
4	100	100	$1.27 \cdot 10^7$	2^{23}
5	94	83	122.03	2^{23}
6	3	0	1.34	2^{23}
Ядро (прямой ход)				
1	100	100	$3.50 \cdot 10^7$	2^{21}
2	100	100	$3.48 \cdot 10^7$	2^{21}
3	100	100	$7.09 \cdot 10^6$	2^{21}
4	100	100	1872.36	2^{21}
5	0	0	0.35	2^{23}
Ядро (обратный ход)				
1	100	100	$1.65 \cdot 10^7$	2^{20}
2	100	100	$1.64 \cdot 10^7$	2^{20}
3	100	100	$3.03 \cdot 10^6$	2^{20}
4	100	100	$1.38 \cdot 10^6$	2^{22}
5	68	40	2.90	2^{24}
Обратное перемешивание				
1	100	100	$1.64 \cdot 10^7$	2^{20}
2	100	100	$1.64 \cdot 10^7$	2^{20}
3	100	100	$2.07 \cdot 10^7$	2^{20}
4	100	100	$7.95 \cdot 10^6$	2^{20}
5	100	100	$7.48 \cdot 10^6$	2^{20}
6	100	100	$6.60 \cdot 10^4$	2^{20}
7	100	100	140.42	2^{24}
8	5	1	0.61	2^{24}

Из табл. 13 видно, что градиентная статистическая атака успешно реализуется на 5 раундах преобразований ядра. На этапах перемешивания отклонения от случайности устойчиво обнаруживаются на 5 раунде на этапе прямого перемешивания и на 7 раунде на этапе обратного перемешивания. Учитывая, что все преобразования перемешивания обратимы и не требуют никакой ключевой информации, можно подавать на вход такие данные, что на этап криптографического ядра будет поступать заранее определённая последовательность вида (1), что позволяет увеличить количество раундов, поддающихся атаке, сразу на 8. Одновременно, перед анализом выходной последовательности, применим к ней преобразование, обратное к обратному перемешиванию, что также увеличит число атакуемых раундов на 8. В итоге, при реализации атаки нам необходимо атаковать только раунды криптографического ядра. В общем итоге, число успешно атакуемых раундов в варианте шифра без операций забеливания равно 21 при длине входной последовательности, равной 2^{23} бит. В общем слу-

чае, при реализации атаки на полный раунд, требуется использование раундового ключа длиной в 64 бита. Трудоёмкость в таком случае будет равной 2^{87} бит без возможности расширения методом применения гибридного варианта.

Рассмотрим модификацию атаки, применимой к полураундам. Функция шифрования ядра (*E*-функция) использует подключи раунда key_1 и key_2 в формировании блоков *M, L, R*. При этом в формировании блока *R* используется только подключ key_2. Одновременно в криптографическом ядре блок делится на 4 части, циклически сдвигаемых на входе в преобразование, на каждом раунде. По номеру раунда можно определить циклический сдвиг. Кроме того, на каждом раунде шифруется 3 блока из 4 с использованием, соответственно, 3 блоков *L, M* и *R*, сгенерированных *E*-функцией. Это означает, что в формировании 2 блоков из 4 на каждом раунде участвует только подключkey_2. Будем проводить атаку, беря в анализ на каждом новом раунде только те 2 блока, в формировнии которых участвует только один подключ. Подбираем его стандартным путём. Далее, зная один подключ, реализуем перебор по второму подключу и таким образом реализуем атаку на весь раунд.

Указанная модификация атаки была реализована и проведена с длиной входной последовательности, превышающей требуемую для атаки в варианте полных раундов в два раза. Результаты получились в точности теми же, что и в случае атаки на полные раунды: 5 раундов ядра (21 раундов всего). При этом трудоёмкость уменьшилась с 2^{87} до 2^{56}, а длина входа стала равной 2^{24}. Учитывая 128-битную длину ключа, можем применить гибридный вариант атаки, увеличив количество атакуемых раундов на 1. В итоге, получим 21 раунд при $T = 2^{56}$ и 22 раунда при$T = 2^{120}$. Из ранее известных результатов лучшим является дифференциальная атака на шифр MARS, описания и результаты которой приведены в [15] и [18]. Количество поддающихся атаке раундов равно 16. Это позволяет говорить о том, что без учёта операций забеливания, разработанная модификация градиентной статистической атаки является лучшей на текущий момент атакой на шифр MARS.

7. Анализ градиентной статистической атаки на шифр CAST-128

7.1. Описание структуры шифра

CAST-128 – блочный алгоритм симметричного шифрования на основе сети Фейстеля. Размер секретного ключа составляет 128 бит. CAST-128 состоит из 16 раундов сети Фейстеля с размером блока 64 бита и длиной ключа от 40 до 128 бит (но только с инкрементацией по 8 бит). 16 раундов используются, когда размеры ключа превышают 80 бит. В алгоритме используются S-блоки, основанные на бент-функции, операции XOR и модулярной арифметике (модулярное сложение и вычитание). При этом S-блоки в шифре CAST-128 статические, т.е. не зависят от исходного ключа. Есть три различных типа функций раундов, но они похожи по структуре и различаются только в выборе выполняемой операции (сложение, вычитание или XOR) в различных местах. Подробное описание алгоритма CAST-128 можно найти в [6] и [21].

7.2. Атака на CAST-128

Был рассмотрен вариант шифра CAST-128 со 128-битным ключом и, соответственно, 16 раундами. В данном шифре операции забеливания не используется, что делает возможным применение стандартной версии градиентной атаки. Раунды шифра и подключи неделимы, а размер подключа только в 2 раза меньше размера исходного ключа, что делает применимым только стандартный вариант атаки без возможности расширения, как это было в случае реализации атаки на шифры RC6 и MARS.

Результаты реализации градиентной статистической атаки на шифр приведены в табл. 14. Результаты приведены сразу для наиболее оптимальных параметров статистического теста для двух последних поддающихся атаке раундов (на которых видны статистические различия).

Таблица 14. Атака на шифр CAST-128

Раунд	Размер выборки	Параметры теста (w, u)	$> Q_{0.95}$	$E(x^2)$
2	2^{25}	(24, 12)	93	268.47
2	2^{25}	(16, 15)	98	1518.9
2	2^{25}	(16, 10)	100	1089.4
2	2^{25}	(16, 8)	95	982.27
3	2^{31}	(24, 12)	52	3.14
3	2^{31}	(24, 16)	61	3.91
3	2^{31}	(16, 10)	84	8.87
3	2^{31}	(16, 8)	71	8.40

Исходя из табл. 14 можно сделать вывод о возможности реализации атаки на 3–4 раунда шифра CAST-128 при трудоёмкости $T = 2^{89}$ для 3 раундов и $\sim 2^{97}$ для 4 раундов. Опубликованных атак на полную версию шифра нет, что позволяет считать полученные результаты лучшими на текущий момент.

8. Анализ градиентной статистической атаки на шифр IDEA

8.1. Описание структуры шифра

Рассмотрим блоковый шифр IDEA. Он использует 8.5 раундов со 128-битным исходным ключом. Размер блока равен 64 битам, а размер раундового ключа составляет 96 бит, что сразу исключает применимость гибридной градиентной атаки.

Процесс шифрования в IDEA основан на модифицированной сети Фейстеля. Каждый исходный 64-битный блок делится на четыре подблока по 16 бит каждый: все алгебраические операции, использующиеся в процессе шифрования, совершаются над 16-битными числами. Для шифрования и дешифрования IDEA использует один и тот же алгоритм. Подробное описание алгоритма IDEA можно найти в [22].

8.2 Атака на шифр IDEA

Результаты атаки на полную версию шифра приведены в табл. 15. Результаты приведены сразу для некоторого множества оптимальных параметров статистического теста.

Таблица 15. Атака на шифр IDEA

Раунд	Размер выборки	Параметры теста (w, u)	$> Q_{0.95}$	$E(x^2)$
1	2^{16}	(16, 8)	100	18.42
1	2^{14}	(8, 4)	76	9.84
2	2^{25}	(16, 10)	0	1.47
2	2^{31}	(24, 12)	7	3.14

Исходя из данных табл. 15 можем говорить об успешной реализации градиентной атаки лишь на 2-ой раунд шифра IDEA при выборке размером 2^{16} блоков, т.е. при общей трудоёмкости $T = 2^{112}$ операций. Как видно, в случае данного шифра градиентная атака показывает малую эффективность. Лучшая атака на шифр MARS, описанная в [16], взламывает 6 раундов шифра.

9. Анализ статистических свойств шифра Blowfish

Алгоритм блокового шифрования Blowfish основан на сети Фейстеля, состоящей из 16 раундов, и использует 64-битные блоки с размером исходного секретного ключа от 32 до 448 бит. Размер раундового ключа составляет 32 бита. В раундах шифрования Blowfish исполь-

зуются S-блоки, которые не являются статическими: на стадии генерации подключей генерируются и S-блоки, зависящие функционально полно от исходного секретного ключа. Это делает невозможным применение градиентной атаки как таковой в силу фактической зависимости функции раундового шифрования от всего исходного секретного ключа. Тем не менее, можно провести анализ статистических свойств шифра. Подробное описание алгоритма Blowfish можно найти в [6].

Основные результаты, полученные после анализа выходной последовательности шифра при стандартном входе (1), представлены в табл. 16.

Таблица 16. Статистический анализ шифра Blowfish

Раунд	Размер выборки	Параметры теста (w, u)	$> Q_{0.95}$	$E(x^2)$
0	2^{19}	(16, 10)	100	10^8
0	2^{25}	(16, 10)	100	10^9
1	2^{19}	(16, 10)	100	10^7
1	2^{19}	(8, 4)	100	10^7
1	2^{25}	(16, 10)	100	10^8
2	2^{19}	(16, 10)	64	15.44
2	2^{19}	(8, 4)	74	32.4
2	2^{25}	(16, 10)	95	1010
3	2^{19}	(16, 10)	3	0.12
3	2^{19}	(8, 4)	5	1.15
3	2^{25}	(16, 10)	5	0.75
3	2^{30}	(32, 24)	16	1.51
3	2^{31}	(24, 12)	17	1.35
3	2^{31}	(24, 16)	6	0.77
3	2^{31}	(32, 24)	5	1.03

Как видно из вышеприведённых результатов, устойчивые статистические отклонения от случайности удаётся обнаружить на 2-ом раунде. При выборке размером в 2^{31} блоков (что равно входной последовательности длиной в 4 Гб) отклонения обнаруживаются максимум лишь в 17% случаев.

Лучшая атака на данный шифр, разработанная Джоном Кесли, позволяет вскрыть 3 раунда из 16 и описана в [17].

10. Заключение

В результате проделанной работы были получены множественные результаты по атакам и анализу криптографических свойств пяти различных блоковых шифров: RC6, MARS, IDEA, CAST-128, Blowfish; исследованы особенности применения статистического теста «стопка книг». Были разработаны новые модификации градиентной статистической атаки и новые методы её применения в случае конкретных видов шифров. Также были выявлены и практически обоснованы теоретические зависимости между такими величинами, как эффективность атаки на примере конкретных шифров, временна́я и аппаратная сложность, параметры применения атаки.

Итоговые результаты по атакам на шифры приведены в табл. 17.

Таблица 17. Итоговые результаты атак и статанализа рассмотренных шифров

Название шифра	Максимальный раунд	Лучшие опубликованные результаты	Трудоёмкость
RC6	5/6	5	$2^{50}/2^{57}$
MARS	21/22	16	$2^{56}/2^{120}$
CAST-128	3–4	–	2^{63}
IDEA	2	6	2^{108}
Blowfish (стат. отклон.)	2	3	2^{25}

Как видно из данных табл. 17, по шифрам RC6, MARS и CAST-128 были получены результаты, превосходящие все ранее известные.

Литература

1. *Knudsen L., Meier W.* Correlations in RC6 with a reduced number of rounds // FSE 2000. LNCS 1978(2000). Springer–Verlag. P. 94–108.
2. *Меняшев А.А., Монарев В.А., Рябко Б.Я., Фионов А.Н.* Применение градиентной статистической атаки к блоковым шифрам RC5, RC6 И SAFER // Научно-практический журнал «Информационное противодействие терроризму», 2006. №7. С. 189-193.
3. *Монарёв В.А., Лубкин А.М.* Эффективная атака на блоковый шифр RC6 // Вестник Сиб-ГУТИ, 2010. №4. С. 55-60.
4. *Рябко Б.Я., Фионов А.Н.* Криптографические методы защиты информации, 2006, Москва. С. 181-202.
5. *Рябко Б.Я., Монарев В.А., Шокин Ю.И.* Новый тип атак на блоковые шифры // Проблемы передачи информации, 2005. т. 41. н.4. С. 181– 128.
6. *Шнайер Б.* Прикладная криптография// NewYork:Wiley, 1996. P. 124-147.
7. *Ryabko B.Ya., Monarev V.A.* Using information theory approach to randomness testing // Journal of Statistical Planning and Inference, 2005. Vol. 133. № 1. PP. 95-110.
8. *Рябко Б.Я., Пестунов А.И.* «Стопка книг» как новый статистический тест для случайных чисел // Пробл. передачи информации, 2004. Т. 40. вып. 1. С. 73-78.
9. *L'Ecuyer P.* Tables of linear congruential generators of different sizes and good lattice structure // Math. of Comp, 1999. Vol. 68. P. 249-260.
10. *Кнут Д.Э.* Искусство программирования. Т. 2: Получисленные алгоритмы. М.: Изд. дом «Вильямс», 2000.С. 60-148.
11. *Pestunov A.* Statistical Analysis of the MARS Block Cipher // Cryptology ePrint Archive. Report 2006/217 [Электронный ресурс]. URL:`http://eprint.iacr.org/2006/217` (дата обращения: 18.06.2011).
12. *Ryabko B., Monarev V.* Using information theory approach for randomness testing // J. of Statistical Planning and Reference. 2005. Vol. 133, N 1. P. 95-110.
13. *Doroshenko S., Ryabko B.* The experimental distinguishing attack on RC4 // Cryptology ePrint Archive. Report 2006/070. 2006. [Электронный ресурс]. URL:`http://eprint.iacr.org/2006/070` (дата обращения: 21.10.2011).
14. *Пестунов А.И.* Теоретическое исследование свойств статистического теста «стопка книг». Вычислительные технологии 2006, Т.11, №6. С. 96-102.
15. *Пестунов А.И.* Дифференциальный криптоанализ блочного шифра MARS // Прикладная дискретная математика 2009. №6. С. 56-63.
16. *Biham E., Dunkelman O., Keller N.* A New Attack on 6-Round IDEA // Lecture Notes in Computer Science. Berlin, Heidelberg: Springer-Verlag, 2007. V. 4593. P. 211-224.
17. *Schneier Bruce.* The Blowfish Encryption Algorithm -- One Year Later // *Dr. Dobb's Journal*, 1995. P. 101-137.

18. *Meier W., Brugg-Windisch HTL.*, Switzerland. Workshop on the theory and application of cryptographic techniques on Advances in Cryptology EUROCRYPT '93 Proceedings. Secaucus, NJ, USA: Springer-Verlag New York, Inc, 1994. P. 371-385.
19. *Rivest R.L., Robshaw M.J.B., Sidney R., and Yin Y.L.* The RC6 BlockCipher, 1998. P. 42-57.
20. *Carolynn Burwick Don Coppersmith*. The MARS Encryption Algorithm. [Электронный ресурс]. URL: http://csis.bits-pilani.ac.in (дата обращения: 24.05.2011).
21. *Schneier Bruce*. Applied Cryptography, (2nd edition). John Wiley & Sons. 1996. PP. 334–335.
22. *Demirci Hüseyin, Türe Erkan, Selçuk Ali Aydin*. A New Meet in the Middle Attack on The IDEA Block Cipher, 10th Annual Workshop on Selected Areas in Cryptography, 2004.

Статья поступила в редакцию 26.12.2012;
переработанный вариант — 28.02.2013

Лысяк Александр Сергеевич
Инженер лаборатории информационных технологий в вычислительных системах СибГУТИ, ассистент кафедры систем информатики НГУ (630090, Новосибирск, ул. Пирогова, 2), тел. +79130110883; e-mail: accemt@gmail.com.

Рябко Борис Яковлевич
д.т.н., ректор СибГУТИ, профессор кафедры прикладной математики и кибернетики СибГУТИ (630102, Новосибирск, ул. Кирова, 86) тел. (383) 2-698-272, e-mail: boris@ryabko.net.

Фионов Андрей Николаевич
д.т.н., проректор по научной работе СибГУТИ, профессор кафедры прикладной математики и кибернетики СибГУТИ (630102, Новосибирск, ул. Кирова, 86) тел. (383) 2-698-272, e-mail: fionov@sibsutis.ru.

Efficiency analysis of gradient statistical attack on block ciphers RC6, MARS, CAST-128, IDEA, Blowfish

A.S. Lysyak, B.Ya. Ryabko, A.N.Fionov

This paper is devoted to the pilot study of statistical techniques in cryptanalysis on the basis of block ciphers RC6, MARS, CAST-128, IDEA and Blowfish. The suggested gradient attack is based on statistical test attack called the «book stack». The described scheme and modifed attacks can significantly reduce the complexity of finding the secret key in comparison with previously known types of attacks. This paper studies the effectiveness of gradient statistical attack demonstrating its modern practical and theoretical application (9 rounds of RC6, 21 rounds of MARS and 4 rounds of CAST-128). Analytical dependence between effectively attacked rounds and time/hardware complexity is derived. In addition, in the case of RC6 and MARS, modifications significantly reducing space and operational complexity were developed. Also, this paper shows a selection method of optimum parameters of attack; operational and space complexity were investigated and their dependence on parameters and ciphertext size were shown. Theoretical demand for computing power needed to carry out the attack is described.

Keywords: gradient attack, statistical attack, cryptoanalysis, block ciphers, RC6, MARS, CAST-128, IDEA, Blowfish.