Шарифуллин Сергей Равильевич – e-mail: SharifullinSR@mail.ru; тел.: 89054726222; кафедра криптографических средств защиты информации и математических основ криптологии; начальник; к.т.н.; доцент.

Хисамов Франгиз Гильфанетдинович – e-mail: kiiz@rambler.ru; тел.: 88612523031; кафедра специальной связи; д.т.н.; профессор.

Culish Olga Aleksandrovna – Branch of the Military Academy of Communications, Krasnodar; e-mail: culish_olga@mail.ru; 4, Krasin street, Krasnodar, 350035, Russia; phone: +79615213577; the department of cryptographic systems for information protection and mathematical foundations of cryptology; senior lecturer; cand. of phis.-math. sc; associate professor.

Sharifullin Sergey Raviljevich – e-mail: SharifullinSR@mail.ru; phone: +79054726222; the department of cryptographic systems for information protection and mathematical foundations of cryptology; cand. of eng. sc; associate professor.

Khisamov Frangiz Gilfanetdinovich – e-mail: kiiz@rambler.ru; phone: +78612523031; the department of special communications; dr. of eng. sc.; professor.

УДК 681.03.245

Л.К. Бабенко, Е.А. Ищукова

АНАЛИЗ СИММЕТРИЧНЫХ КРИПТОСИСТЕМ*

Рассматриваются основные аспекты применения современных методов криптоанализа для оценки стойкости симметричных блочных алгоритмов шифрования. В том числе, рассмотрены такие методы анализа как линейный, дифференциальный, алгебраический анализы, слайдовая атака. Также рассмотрены различные подходы к анализу стандарта AES. Дано определение Парадоксу дней рождений и объяснена его роль в решении задач защиты информации. При рассмотрении подходов к анализу современных симметричных криптосистем отдельное внимание уделено вопросам возможности применения распределенных многопроцессорных вычислений с целью сокращения времени анализа.

Криптография; крипроанализ; симметричное шифрование; секретный ключ; блочный шифр; стойкость; распределенные многопроцессорные вычисления.

L.K. Babenko, E.A. Ischukova

ANALYSIS OF SIMMETRIC CRYPTOSYSTEMS

Article considers the main aspects of application of modern methods of cryptanalysis for an assessment of strength of symmetric block algorithms of enciphering. Considering include such methods of the analysis as linear, differential, algebraic analyses and slide attack. The different variants of analysis of AES are considered. It is described what the Birthday Paradox is and what is its role in the information security. The special attention is given to questions of possibility of using of the distributed multiprocessing calculations for reduction of time of the analysis.

Cryptography; cryptanalysis; symmetric cipher; secret key; block cipher; strength; multiprocessing calculations.

Современные алгоритмы шифрования разрабатываются таким образом, чтобы аналитик имел как можно меньше шансов отыскать секретный ключ, с помощью которого были зашифрованы данные, даже если ему известен сам алгоритм шифрования и есть в наличие несколько текстов и соответствующих им шифртекстов. Приступая к задаче анализа, первым делом аналитик определяет тот набор данных, который ему изначально известен для анализа. От этого зависит тот тип

* Работа поддержана грантами РФФИ: № 12-07-00037-а, №12-07-31120-мол_а.

криптоанализа, который аналитик сможет использовать. Рассмотрим основные виды криптоанализа современных симметричных криптосистем.

Метод полного перебора. Если известен алгоритм шифрования и есть хотя бы одна пара открытый – шифрованный текст, то самым естественным способом анализа, который сразу приходит в голову, является последовательное опробование всех возможных вариантов ключа, которые могли быть использованы. Опробование производят до тех пор, пока зашифрование открытого текста на очередном ключе не приведет к получению имеющегося шифрованного сообщения. Такой способ анализа в разных источниках литературы имеет разные названия, например «Метод полного перебора» [1] или «Метод грубой силы» [2] или «Метод атаки в лоб» [3] или «Brut-force атака» [2]. У этого метода есть одно неоспоримое преимущество: рано или поздно искомый ключ будет найден и для этого будет необходим минимальный набор данных. Быстрота нахождения ключа будет зависеть от длины используемого секретного ключа и от вычислительной мощи, которая есть в наличии у аналитика. А также от доли везения. Ведь может случиться так, что искомый ключ встретится одним из первых.

Вместе с тем нам известно, что одним из важных свойств информации является ее своевременность. Поэтому применение метода полного перебора на практике легко реализуется, но как правило не используется. Так, например, когда разрабатывался алгоритм шифрования DES, длина его фактического секретного ключа была определена в 56 бит. То есть для того, чтобы перебрать все возможные варианты секретных ключей, необходимо было сделать 2^{56} опробований. С помощью имевшихся в то время вычислительных средств это можно было бы сделать за несколько десятков лет! Конечно, с той поры как был разработан алгоритм шифрования DES, в развитии вычислительной технике произошел огромный скачок и вычислительные мощности возросли в тысячи раз. Сегодня с использованием мощных вычислительных кластеров задача по поиску секретного ключа для алгоритма DES может быть решена за несколько минут. В связи с тем, что вычислительная мощность с каждым днем неумолимо растет, стандарт DES был заменен на новый стандарт AES (Advanced Encryption Standard), где длина секретного ключа возросла до 128 бит. Так или иначе, в криптографии принято время анализа с помощью метода полного перебора считать эталонным. Что это означает? Это значит, что если аналитику удастся провести анализ алгоритма шифрования быстрее, чем это можно сделать с помощью полного перебора, то данный алгоритм шифрования будет считаться уязвимым, в связи с чем его использовать для шифрования данных будет нецелесообразно.

Задача поиска секретного ключа шифрования методом полного перебора хорошо распараллеливается и легко может быть реализована для многопроцессорных вычислительных систем.

Метод встречи посередине. Метод встреча посередине применим к алгоритмам шифрования, в которых используется два различных ключа К. Это может быть достигнуто в том случае, если секретные подключи появляются с какой-то периодичностью, или, например, если было произведено двойное зашифрование данных, то есть сначала данные зашифровали на одном ключе $К_1$, а затем полученный результат шифрования еще раз зашифровали на другом секретном ключе $К_2$.

Пусть нам известна пара открытый – закрытый текст, зашифрованная подобным образом. В этом случае, необходимо произвести зашифрование открытого текста на всех возможных значениях ключа $К_1$. Параллельно с этим необходимо произвести дешифрование закрытого текста на всех возможных значениях ключа $К_2$. Та пара ключей ($К_1$, $К_2$), для которой результат шифрования открытого текста и результат дешифрования закрытого текста совпадут, и будет являться искомой.

Как видно из объяснений, анализ на основе метода «встреча посередине» может быть распараллелен и реализован с использованием распределенных многопроцессорных вычислений. В качестве примера работы метода можно рассмотреть варианты анализа двойного алгоритма DES или, например, анализа алгоритма ГОСТ 28147-89, в котором один и тот же ключ фактически используется четыре раза.

Линейный криптоанализ. Метод линейного криптоанализа впервые был предложен в начале 90-х гг. XX в. японским ученым М. Матсуи (Matsui). В своей работе [4] М. Матсуи показал, как можно осуществить атаку на алгоритм шифрования DES, сократив сложность анализа до 2^{47}. Существенным недостатком метода стала необходимость иметь в наличии большой объем данных, зашифрованных на одном и том же секретном ключе, что делало метод малопригодным для практического применения к вскрытию шифра. Однако, если предположить, что к аналитику в руки попал шифрованный текст, содержащий важные сведения, а также некий черный ящик (устройство или программа), который позволяет выполнить любое число текстов, зашифрованных с помощью известного алгоритма шифрования на секретном ключе, не раскрывая при этом самого ключа, то применение метода линейного криптоанализа становится вполне реальным. Многие алгоритмы шифрования, известные на момент опубликования работы Матсуи [4], в последствии были проверены на устойчивость к этому методу и не все из них оказались достаточно стойкими и, как следствие, потребовали доработки.

Знание механизмов работы метода линейного криптоанализа позволяет криптографам еще на этапе проектирования крптоалгоритмов обеспечить стойкость шифров. Вот почему так важно уметь применять известные методы криптоанализа на практике.

Итак, рассмотрим основные понятия, связанные с методом линейного криптоанализа. Любой алгоритм шифрования в самом общем виде можно представить как некоторую функцию E, зависящую от входного сообщения X, секретного ключа K и возвращающую шифрованное сообщение Y:

$$Y = E(X, K). \quad (1)$$

Зная само преобразование E и входное сообщение X, нельзя однозначно сказать каким будет выходное сообщение Y. В данном случае нелинейность функции (1) зависит от внутренних механизмов преобразования E и секретного ключа K. М. Матсуи показал, что существует возможность представить функцию шифрования (1) в виде системы уравнений, которые выполняются с некоторой вероятностью **p**. При этом для успешного проведения анализа вероятность уравнений **p** должна быть как можно дальше удалена от значения 0,5 (то есть приближаться либо к 0 либо к единице). Так как уравнения, получаемые в ходе анализа криптоалгоритма, являются вероятностными, то их стали называть линейными статистическими аналогами.

Линейным статистическим аналогом нелинейной функции шифрования (1) называется величина Q, равная сумме по модулю два скалярных произведений входного вектора X, выходного вектора Y и вектора секретного ключа K соответственно с двоичными векторами α, β и γ, имеющими хотя бы одну координату равную единице:

$$Q = (X, \alpha) \oplus (Y, \beta) \oplus (K, \gamma)$$

в том случае, если вероятность того, что Q=0 отлична от 0,5 ($P(Q=0) \neq 0,5$).

В отличие от дифференциального криптоанализа, в котором большое значение вероятности гарантирует успех атаки, в линейном криптоанализе успех анализа может быть обеспечен как уравнениями с очень большой вероятностью, так и уравнениями с очень маленькой вероятностью. Для того, чтобы понять, какое из возможных уравнений лучше всего использовать для анализа используют понятие отклонения.

Отклонением линейного статистического аналога называют величину $\eta = |1–2p|$, где p – вероятность, с которой выполняется линейный аналог.

Отклонение определяет эффективность линейного статистического аналога. Чем отклонение больше, тем выше вероятность успешного проведения анализа. Фактически отклонение показывает насколько вероятность статистического аналога отдалена от значения $p = 0,5$.

Для успешного применения метода линейного криптоанализа необходимо решить следующие задачи:

1. Найти максимально эффективные (или близкие к ним) статистические линейные аналоги. При нахождении аналогов обратить внимание на то, что в них должно быть задействовано как можно больше битов искомого секретного ключа К.
2. Получить статистические данные: необходимый объем пар текстов (открытый – закрытый текст), зашифрованных с помощью анализируемого алгоритма на одном и том же секретном ключе.
3. Определить ключ (или некоторые биты ключа) путем анализа статистических данных с помощью линейных аналогов.

Первый шаг анализа заключается в нахождении эффективных статистических аналогов. Для алгоритмов шифрования, в которых все блоки заранее известны, этот шаг можно выполнить единожды, основываясь на анализе линейных свойств всех криптографических элементов шифра. В результате анализа должна быть получена система уравнений, выполняющихся с некоторыми вероятностями. Левая часть уравнений должна содержать в себе сумму битов входного и выходного сообщения, правая часть уравнения – биты секретного ключа. Система уравнений должна быть определенной, то есть содержать все биты исходного секретного ключа. Данный этап не является вычислительно сложным, однако требует больших знаний, логики работы и внимательности. Он может быть автоматизирован. Однако при этом необходимо помнить, что для каждого определенного алгоритма шифрования система линейных аналогов строиться всего один раз и в дальнейшем может быть использована для нахождения разных секретных ключей шифрования, которые используются для шифрования данных с помощью анализируемого шифра.

Если первый шаг анализа является чисто теоретическим и полностью зависит от структуры алгоритма, то второй шаг – является исключительно практической частью, которая заключается в анализе известных пар открытый-закрытый текст с помощью полученной ранее системы статистических аналогов. Для этого используется следующий алгоритм.

Алгоритм. Пусть N – число всех открытых текстов и T – число открытых текстов, для которых левая часть линейного статистического аналога равна 0. Рассмотрим два случая.

1. Если $T > N/2$, то в этом случае число открытых текстов, для которых левая часть аналога равна нулю, больше половины, то есть в большинстве случаев в левой части аналога появляется значение, равное нулю, то

а) если вероятность этого линейного статистического аналога $p > 1/2$, это говорит о том что в большинстве случаев правая и левая части аналога равны, а значит левая часть аналога, содержащая биты ключа, равна 0.

б) если вероятность этого линейного статистического аналога $p < 1/2$, это говорит о том что в большинстве случаев правая и левая части аналога не равны, а значит левая часть аналога, содержащая биты ключа, равна 1.

2. Если $T < N/2$, то в этом случае число открытых текстов, для которых левая часть аналога равна нулю, меньше половины, то есть в большинстве случаев в левой части аналога появляется значение, равное единице, то

а) если вероятность этого линейного статистического аналога p >1/2, это говорит о том что в большинстве случаев правая и левая части аналога равны, а значит левая часть аналога, содержащая биты ключа, равна 1.

б) если вероятность этого линейного статистического аналога p < 1/2, это говорит о том что в большинстве случаев правая и левая части аналога не равны, а значит левая часть аналога, содержащая биты ключа, равна 0.

Данный алгоритм будет иметь успех при анализе большого числа текстов N. Следовательно, второй шаг анализа является вычислительно сложным. Поэтому для ускорения времени анализа можно и нужно использовать параллельные вычисления.

В результате работы вышеприведенного алгоритма будет получена определенная (а возможно и переопределенная) система уравнений, отражающая взаимосвязь битов ключа. Третий шаг анализа заключается в решении данной системы, например, методом Гаусса, что позволит получить значения битов секретного ключа шифрования.

Более подробно о линейном криптоанализе различных блочных алгоритмов шифрования можно почитать в работе [1].

Дифференциальный криптоанализ. Метод дифференциального криптоанализа (ДК) впервые был предложен в начале 90-х годов прошлого века Э. Бихамом и А. Шамиром для анализа алгоритма шифрования DES. Хотя в книге Б. Шнайера [3] упоминается о том, что разработчики алгоритма DES знали о возможности такого анализа еще во время разработки алгоритма в 70-х годах XX века, широкая общественность узнала о дифференциальном криптоанализе именно из работ [5, 6]. Метод ДК оказался первым методом, позволяющим взломать DES при оценке сложности задач менее 2^{55}. Согласно [5], с помощью данного метода можно провести криптоанализ DES при усилиях порядка 2^{37}, но при наличии 2^{47} вариантов избранного открытого текста. Хотя 2^{47}, очевидно, значительно меньше, чем 2^{55}, необходимость при этом иметь 2^{47} вариантов избранного открытого текста превращает данный вариант схемы криптоанализа в чисто теоретическое упражнение [7]. Это связано с тем, что метод ДК был известен в момент разработки DES, но засекречен по очевидным соображениям, что подтверждается публичными заявлениями самих разработчиков [3]. В работе [6] показано, что если поменять порядок следования блоков замены в алгоритме шифрования DES или использовать другие наборы таблиц подстановок и перестановок, то алгоритм становится сразу намного слабее и может быть взломан менее чем за половину времени, требуемой для анализа алгоритма DES с помощью полного перебора. Это показывает значимость знания возможных путей анализа разрабатываемого алгоритма.

С помощью метода ДК сложность анализа сократилась до 2^{37}. Однако при этом для проведения анализа необходимо было иметь 2^{37} особым образом подобранных текстов, зашифрованных на одном и том же секретном ключе. Не смотря на накладываемые ограничения в использовании новых предложенных методов анализа – это был прорыв! Дальнейшее развитие этого метода показало возможность его применения к целому классу различных видов шифров, позволило выявить слабые места многих используемых и разрабатываемых алгоритмов шифрования. Сегодня этот метод, а также некоторые его производные, такие как метод линейно-дифференциальный, метод невозможных дифференциалов, метод бумеранга широко используются для оценки стойкости вновь создаваемых шифров. Именно поэтому специалисту по защите информации необходимо иметь представление о механизмах анализа шифров с использованием современных методов криптоанализа.

Само название дифференциальный криптоанализ происходит от английского слова difference, т.е. разность. Именно поэтому в отечественной литературе этот вид анализа еще иногда называют разностным методом. Исходя из названия, можно понять, что при рассмотрении возможности анализа некоторого блочного алгоритма шифрования ученым пришло в голову использовать не отдельные тексты, а пары текстов. Понятно, что два текста будут иметь различия в некоторых позициях. Для того, чтобы определить это различие, достаточно пару текстов сложить между собой по модулю два. Результат такого сложения даст на выходе значение 0 в тех позициях, в которых исходные тексты были равны между собой, и соответственно значение 1 в тех позициях, в которых исходные тексты отличались. Например, рассмотрим два 4-битовых сообщения: X = 0011 и X’ = 1010. В результате сложения текстов X и X’ была получена разность $\Delta X = 1001$, полученное значение ΔX принято называть дифференциалом или разностью. В дифференциальном криптоанализе значение разности (дифференциала) принято обозначать символом Δ. Разность, полученная в результате сложения текстов X и X’ показывает, что во второй и третьей позициях исходные сообщения X и X’ были равны, а в первой и четвертой отличались друг от друга.

В общем виде дифференциальный анализ блочных алгоритмов шифрования сводится к следующим основным пунктам:

1. Нахождение для алгоритма шифрования характеристик, обладающих максимальными характеристиками. Поиск характеристик ведется на основе дифференциальных свойств нелинейных криптографических примитивов, входящих в состав алгоритма шифрования.
2. Поиск правильных пар текстов с использованием найденных характеристик.
3. Анализ правильных пар текстов и накопление статистики о возможных значениях секретного ключа шифрования.

Первый пункт, заключающийся в поиске лучших характеристик для большинства алгоритмов выполняется единожды и является теоретической задачей. Значения характеристик полностью зависят от структуры алгоритма шифрования и используемых криптографических примитивов. Иначе дело обстоит лишь с теми алгоритмами, которые обладают нефиксированными элементами. К таким алгоритмам можно, например, отнести алгоритм шифрования ГОСТ 28147-89, у которого S-блоки замены могут выбираться произвольным образом. Для таких алгоритмов поиск характеристик необходимо каждый раз начинать сначала, основываясь на дифференциальных свойствах выбранных S-блоков. Для автоматизации процесса анализа можно разработать алгоритм поиска лучших характеристик, основываясь на алгоритмах поиска по дереву. Для таких алгоритмов можно использовать параллельные модели для ускорения поиска характеристик.

Второй шаг анализа является вычислительно стойкой задачей для любого алгоритма шифрования, при этом не важно, обладает он фиксированными или нефиксированными элементами. Анализ заключается в опробовании большого числа пар текстов с целью определения являются ли они правильной парой текстов, то есть той парой текстов, которую в дальнейшем можно использовать для анализа с целью поиска секретного ключа шифрования. Данный шаг может и должен быть легко представим в виде параллельных вычислений для сокращения времени анализа.

Последний шаг легко реализуем, требует гораздо меньше вычислений в сравнении со вторым шагом. Он может быть реализован как отдельно в виде последовательного алгоритма, так и быть включенным в состав параллельных алгоритмов по поиску правильных пар текстов. В последнем случае при нахождении правильной пары текстов сразу можно провести ее анализ по накоплению статистики о возможном значении секретного ключа.

Более подробно о дифференциальном криптоанализе различных блочных алгоритмов шифрования, а также о различных производных данного метода можно почитать в работах [1, 5, 6, 8].

Алгебраический анализ. Сущность алгебраических методов анализа заключается в получении уравнений, описывающих нелинейные преобразования замены S-блоков, с последующим решением найденных систем уравнений и получением ключа шифрования. Данный метод криптоанализа относится к атакам с известным открытым текстом, поэтому для успешного анализа достаточно иметь одну пару открытый текст/шифртекст. Алгебраические методы криптоанализа состоят из следующих этапов:

- составление системы уравнений, описывающей преобразования в нелинейных криптографических примитивах анализируемого шифра (чаще всего для симметричных алгоритмов шифрования такими нелинейными компонентами являются S-блоки замены);
- решение полученной системы уравнений.

Рассмотрим подробнее первый этап алгебраического криптоанализа. Для шифров, подобных Rijndael, при составлении уравнений используется таблица замены S-блоков. Ограничимся рассмотрением одночленов, состоящих из произведения двух переменных. Тогда уравнения, описывающие работу S-блоков, имеют вид [9]:

$$\sum \alpha_{ij} x_i x_j + \sum \beta_{ij} y_i y_j + \sum \gamma_{ij} x_i y_j + \sum \delta_i x_i + \sum \varepsilon_i y_i + \eta = 0,$$

где $x_i x_j$ – комбинация входных битов S-блока; $y_i y_j$ – комбинация выходных битов S-блока; $x_i y_j$ – комбинация входных и выходных битов; x_i и y_i – соответственно входные и выходные биты S-блока; η – коэффициент, принимающий значения 0 или 1.

При получении уравнений нужно рассмотреть все возможные комбинации данных одночленов. В случае, когда число бит на входе S-блоков равно s, получаем, что число одночленов, встречающихся в системе, вычисляется по формуле $t=\frac{2s}{2}+2s+1$ и включает в себя входные и выходные значения S-блока (2s), все их возможные произведения $\binom{2s}{2}$ и коэффициент η. Число всех возможных комбинаций одночленов составляет 2^t.

Для произвольного блока замены число линейно независимых уравнений $r \geq t-2^s$.

Для проверки всех полученных комбинаций на соответствие заданному S-блоку требуется составить таблицу истинности на основании замен, выполняемых в исследуемом S-блоке.

Для проверки комбинаций на соответствие таблице истинности следует осуществить построковую подстановку значений одночленов из таблицы и выполнить операцию сложения по модулю 2. Таким образом, для каждой комбинации выполняется подстановка и сложение для всех возможных входных значений S-блока (2^s раз). Результаты суммирования сравниваются с нулем. Если для всех строк таблицы истинности равенство оказывается верным, то уравнение, заданное данной комбинацией одночленов, удовлетворяет таблице замены исследуемого S-блока, и его следует отобрать для составления искомой системы. Далее необходимо провести анализ уравнений и выбрать для формирования системы линейно независимые уравнения, содержащие минимальное число нелинейных элементов.

Второй этап алгебраического криптоанализа заключается в решении системы. В криптоанализе разработаны различные подходы к решению нелинейных систем булевых уравнений. Наиболее эффективными, как показывает практика криптоанализа, являются методы, использующие линеаризацию исходной системы.

XL метод (eXtended Linearization) предложен Nicolas Courtois, Alexander Klimov, Jacques Patarin и Adi Shamir в работе [10].

Пусть имеется нелинейная система, содержащая m уравнений и 2s переменных. XL метод базируется на умножении каждого уравнения 1…m на произведения переменных степени меньшей или равной D-2. Рассмотрим вычисление параметра D алгоритма XL атаки. При умножении исходных уравнений системы на одночлены степени ≤(D-2) получаем примерно $R \approx \binom{2s}{\mathrm{D}-2} m$ новых уравнений. Общее число одночленов, встречающихся в этих уравнениях, составляет $T = \binom{2s}{D}$. Так как система будет решаться способом линеаризации, то есть путем замены всех нелинейных одночленов на новые переменные, необходимо чтобы число уравнений было больше числа одночленов $R = \binom{2s}{\mathrm{D}-2} m \geq \binom{2s}{D} = T$. Отсюда получаем, что $m \geq \binom{2s}{D} / \binom{2s}{\mathrm{D}-2} \approx (2s)^2 / D^2$. Следовательно, $D \approx \frac{2s}{\sqrt{m}}$. При этом должно выполняться условие D>2, иначе не будет получено новых уравнений, так как степень отобранных для умножения уравнений одночленов, определяемая разностью D-2, будет равна нулю.

Алгоритм XL метода состоит из двух шагов:

- Multiply: умножение каждого уравнения исходной системы на произведение переменных в степени ≤ D-2.
- Linearize: замена каждого одночлена в степени ≤ D на новую переменную и применение метода исключения Гаусса.

Сложность анализа заключается в построении системы всех возможных линейных уравнений и последующего ее решения. Для ускорения процесса анализа построение уравнений для системы можно производить параллельно. Также переопределенную систему многих уравнений целесообразно решать с использованием параллельных вычислений с последующим объединением результата.

Более подробно об алгебраическом криптоанализе различных блочных алгоритмов шифрования можно почитать в работах [9–13].

Анализ стандарта AES. Атака типа «Квадрат». С появлением нового стандарта шифрования AES, в основе которого лежит алгоритм шифрования Rijndael, стал рассматриваться новый принцип построения блочных шифров. По принципу построения и обработки данных стали выделять новую архитектуру под названием «Квадрат». С появлением новой архитектуры в алгоритмах шифрования возникла потребность в разработке новых подходов в анализе подобных криптосистем. Рассмотрим атаку типа «Квадрат» на примере упрощенного алгоритма Rijndael. Для этого к анализу возьмем четырехраундовый алгоритм шифрования Rijndael.

Возьмем 256 входных сообщений, имеющих одинаковые значения во всех байтах кроме одного. Так как операция SubBytes является простой операцией замены одного значения на другое, то после этой операции первого раунда данные все еще будут иметь различия в том единственном байте.

Пусть у нас есть два входа преобразования MixColumn (a, b, c, d) и (a’, b, c, d), тогда соответствующие им выходные значения будут иметь разность (02_x•(a – a’), 01_x•(a – a’), 01_x•(a – a’), 03_x•(a – a’)), где все операции производятся в поле $GF(2^8)$. Следовательно, выходные значения преобразования MixColumn будут иметь отличия во всех четырех байтах. А значит, в нашем случае все четыре байта столбца могут принять любое значение из 256 возможных. Как мы уже выяснили

ранее, после преобразования третьего раунда MixColumn выходные данные при сложении друг с другом по модулю 2 в результате дадут ноль. А значит и перед операцией SubBytes четвертого раунда шифрования это свойство сохранится.

Таким образом, анализ алгоритма шифрования Rijndael может быть проведен с помощью выполнения следующих действий:

1. Необходимо выбрать 256 открытых текстов, которые будут иметь различия только в первом байте.
2. Получить соответствующие им шифртексты, зашифровав их с помощью четырех раундов шифрования Rijndael на секретном ключе.
3. Предположить значение (то есть взять одно из возможных) первого байта подключа четвертого раунда шифрования.
4. Используя 256 известных шифртекстов, получить 256 значений первого байта на входе преобразования SubBytes четвертого раунда.
5. Если сумма полученных 256 значений по модулю два равна нулю, то предположенное значение байта подключа верно.
6. Используя этот же принцип, найти остальные байты подключа четвертого раунда шифрования.
7. Зная значение подключа четвертого раунда, извлечь значение исходного секретного ключа.

Слайдовая атака. С ростом скорости современных компьютеров, скоростные алгоритмы шифрования используют все больше и больше раундов, признавая все существующие криптоаналитические технологии бесполезными. Это главным образом происходит из-за того, что такие популярные методы, как линейный и дифференциальный криптоанализ, являются статистическими атаками, превосходными при статистических непостоянствах. Однако, когда алгоритм шифрования имеет большое количество раундов, каждый добавленный раунд к требует экспоненциального роста усилий атакующего.

Стремление к большому числу раундов можно наглядно увидеть, рассмотрев претендентов на конкурс AES. Несмотря на то, что одним из основных критериев для претендентов была скорость, некоторые представленные кандидаты (при этом не самые медленные) имели действительно большое число раундов: RC6(20), MARS(32), SERPENT(32), CAST(48). Это является следствием того, что после некоторого большого числа раундов даже относительно слабый шифр становится стойким. Например, алгоритм шифрования DES, уже взлом шестнадцати раундов представляет трудную задачу, не говоря о 32 и 48 раундах (двойном и тройном алгоритме DES). Таким образом, для криптоаналитика становится естественным поиск новых методов анализа, не зависящих от числа раундов в алгоритме шифрования.

Здесь мы рассмотрим еще один метод анализа, не зависящий от числа раундов в алгоритме шифрования, который называется “слайдовой атакой” или “скользящей атакой” (Slide Attacks), предложенный в 1999 г. Алексом Бирюковым и Дэвидом Вагнером [16]. Этот метод применим ко всем алгоритмам блочного шифрования

В то время как два других метода криптоанализа, такие как линейный и дифференциальный, концентрируются главным образом на распространенных свойствах техники шифрования, слайдовая атака использует степень самоподобия, что является принципиальным отличием. Под самоподобием понимается использование одной и той же криптографической F-функции, зависящей от одного и того же подключа, в каждом раунде шифрования. В зависимости от структуры алгоритма шифрования слайдовая атака может использовать как слабость процедуры формирования подключей, так и более общие структурные свойства шифра. Самый простой вид этой атаки обычно легко пресечь, избавившись от самоподобия в алгоритме шифрования. Более сложные варианты этой атаки имеют более сложный анализ, и против них гораздо труднее защититься.

Самый простой вариант слайдовой атаки рассчитан на анализ алгоритмов шифрования, состоящих из r раундов, каждый из которых содержит в себе F-функцию, зависящую от одного и того же значения ключа K. Такой тип алгоритмов шифрования называется гомогенным. К гомогенным также относятся алгоритмы, в которых функция формирования подключа периодична, то есть один и тот же подключ извлекается через равное количество раундов. Говоря математическим языком, $F_i = F_j$ для всех $i \equiv j \bmod p$, где p является периодом. В самом простом случае p=1, и в каждом раунде используется один и тот же подключ.

При рассмотрении слайдовых атак, с целью упрощения мы будем рассматривать алгоритмы шифрования, в которых сложение шифруемых данных с подключом происходит непосредственно перед F-функцией. Так, если мы рассматриваем алгоритм шифрования, построенный по схеме Фейстеля, на вход которого поступает n-битное сообщение, то длина подключа будет составлять n/2 битов.

Рассмотрим процесс зашифрования n-битового открытого текста X_0, в результате которого получается шифр-текст X_r. Здесь X_j обозначает промежуточное значение данных после j-го раунда зашифрования, так что

$$X_j = F_j(X_{j-1}, k_j),$$

где j = 1, 2, 3 . . r. В дальнейшем мы иногда будем опускать значение k в обозначении F-функций, и будем писать F(x) или $F_i(x)$ вместо F(x,k) или $F_i(x,k)$. Функция F называется слабой в том случае, если при известных двух равенствах F(x1,k) = y1 и F(x2,k) = y2 ключ k легко определяется. На рис. 1 показано, как может быть применена слайдовая атака к алгоритмам шифрования такого типа.

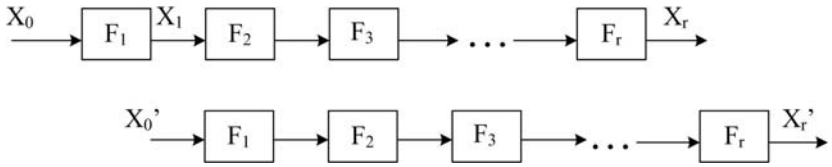

Рис. 1. Схема обычной слайдовой атаки

Идея заключается в том, что можно сопоставить один процесс зашифрования с другим таким образом, что один из процессов будет «отставать» от другого на один раунд.

Пусть X_0 и X_0' обозначают исходные открытые тексты, $X_j = F_j(X_{j-1})$ и $X_j' = F_j(X_{j-1}')$, где j = 1, 2, 3. . . r. Идея заключается в том, что если мы имеем такую пару значений, что $X_1 = X_0'$, то мы также будем иметь соответствующую им пару значений, такую что $X_r = X_{r-1}'$. Предположим, что $X_j = X_{j-1}'$, тогда мы можем сказать, что $X_{j+1} = F(X_j) = F(X_{j-1}') = X_j'$. Пара открытых текстов и соответствующих им шифр-текстов (P,C), (P',C') называется слайдовой парой в том случае, если F(P) = P' и F(C) = C'.

Слайдовая атака проходит следующим образом. Мы получаем $2^{n/2}$ пар открытый – закрытый текст (P_i, C_i), и ищем среди них слайдовые пары. Согласно парадоксу дней рождений, мы ожидаем найти хотя бы одну пару индексов i, i', такую, что $F(P_i) = P_i'$ и $F(C_i) = C_i'$ одновременно выполняются для некоторого ключа. После того, как слайдовая пара найдена, мы можем найти некоторые биты подключа. В том случае, если раундовая функция слабая, мы можем найти весь подключ данного раунда. Для нахождения оставшихся битов секретного ключа необходимо определить следующую слайдовую пару, и с ее помощью провести анализ. Таким образом, достаточно найти всего несколько слайдовых пар для полного определения битов секретного ключа, что и является задачей, стоящей перед криптоаналитиком.

В случае, когда речь идет об алгоритмах шифрования, построенных по схеме Фейстеля, раундовая функция F((l,r), k) = ((l⊕f(r),r), k) модифицирует только половину входного сообщения. Таким образом, условие F(x) = x' можно легко проверить с помощью сравнения левой части сообщения x и правой части сообщения x'. Это условие позволяет нам снизить сложность атаки на основе известных открытых текстов до $2^{n/2}$ известных текстов. У нас есть n-битовое условие нахождения потенциальной слайдовой пары: если (P_i, C_i) образует слайдовую пару вместе с (P_i', C_i'), то тогда $F(P_i) = P_j$' и $F(C_i) = C_j$'. Для нахождения слайдовой пары для алгоритмов шифрования, построенных по схеме Фейстеля, необходимо известные тексты (P_i, C_i) занести в таблицу, после чего для каждого j найти такой текст, чтобы правые половины P_i и C_j' были равны левым половинам P_j' и C_i.

Если не все биты подключа будут найдены с помощью определенной слайдовой пары, то можно будет использовать другие слайдовые пары для определения оставшихся битов.

Для алгоритмов шифрования, построенных по схеме Фейстеля, сложность анализа может быть снижена до $2^{n/4}$ текстов в том случае, если существует возможность использовать выбранные открытые тексты. Для этого необходимо выбрать произвольным образом n/2 битовое значение x. После этого надо подобрать массив из $2^{n/4}$ открытых текстов $P_i = (x, y_i)$, которые будут различаться только случайно выбранной правой частью, и массив из $2^{n/4}$ текстов $P_j' = (y_j', x)$, которые будут различаться только случайно выбранной левой частью. Таким образом, у нас появится $2^{n/2}$ пар открытых текстов, и мы надеемся найти среди них хотя бы одну правильную пару.

В любом из вышерассмотренных случаев поиск слайдовых пар является вычислительно сложной задачей, для решения которой целесообразно использовать параллельные алгоритмы.

Парадокс Дней Рождений и его роль в задачах криптоанализа. Это очень важный вероятностный парадокс, который широко применяется в современной криптографии: от алгоритмов блочного шифрования до систем с открытым ключом [14].

Предпосылкой возникновения парадокса Дней Рождений явился вопрос: как много учеников должно собраться в одном классе, чтобы как минимум двое из них имели день рождения в один и тот же день? С помощью простых вычислений можно выяснить, что если в классе будет находиться 23 ученика, то вероятность того, что у двух из них день рождения в один день, будет больше, чем ½.

Применение парадокса Дней Рождений на практике очень распространено. Предположим, что для атаки на алгоритм шифрования, оперирующий 64-битными блоками данных, противник должен получить две пары открытый – закрытый текст, которые отличаются только младшим значащим битом (типичная задача для дифференциального криптоанализа). Согласно Парадоксу Дней Рождений, только массив, состоящий примерно из 2^{32} открытых текстов, будет содержать требуемые пары с высокой вероятностью. Рассмотрим другой пример для одного раунда 64-битного шифра Фейстеля. Предположим, что в шифре используется произвольная функция F. Злоумышленник может захотеть узнать, как много открытых текстов ему нужно получить для того, чтобы на выходе встретилось два одинаковых шифртекста F-функции. Согласно Парадоксу Дней Рождений, можно определить, что в этом случае требуется только 2^{16} текстов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Грушо А.А., Тимонина Е.Е., Применко Э.А.* Анализ и синтез криптоалгоритмов. Курс лекций. – Йошкар-Ола: Изд-во МФ МОСУ, 2000.
2. *Столлингс В.* Криптография и защита сетей: принципы и практика. – 2-е изд.: Пер. с англ. – М.: Изд. дом «Вильямс», 2001.
3. *Бабенко Л.К. Ищукова Е.А.* Современные алгоритмы блочного шифрования и методы их анализа. – М.: Гелиос АРВ, 2006.
4. *Шнайер Б.* Прикладная криптография: Протоколы, алгоритмы, исходные тексты на языке Си. – М.: ТРИУМФ, 2002. – 648 с.
5. Matsui M., Linear Cryptanalysis Method for DES Cipher, Advances in Cryptology. – EUROCRYPT'93, Springer-Verlag, 1998. – 386 p.
6. *Biham E., Shamir A.,* Differential Cryptanalysis of the Full 16-round DES // Crypto'92, Springer-Velgar, 1998. – P. 487.
7. *Biham E., Shamir A.* Differential Cryptanalysis of DES-like Cryptosystems. Extended Abstract // Crypto'90, Springer-Velgar, 1998. – P. 2.
8. *Панасенко С.* Алгоритмы шифрования. Специальный справочник. – СПб.: БХВ-Петербург, 2009. – 576 с.
9. *Courtois N., Pieprzyk J.* Cryptanalysis of block ciphers with overdefined systems of equations // ASIACRYPT, 2002. – P. 267-287.
10. *Courtois N., Klimov A., Patarin J., Shamir A.* Efficient algorithms for solving overdefined systems of multivariate polynomial equations // EUROCRYPT, 2000. – P. 392-407.
11. *Courtois N., Gregory V. Bard.* Algebraic Cryptanalysis of the Data Encryption Standard // 11-th IMA Conference, 2007. – P. 152-169.
12. *Kleiman E.* The XL and XSL attacks on Baby Rijndael //http://orion.math.iastate.edu/dept/thesisarchive/MS/EKleimanMSSS05.pdf.
13. *Бабенко Л.К., Маро Е.А.*, Алгебраический криптоанализ упрощенного алгоритма шифрования Rijndael // Известия ЮФУ. Технические науки. – 2009. – № 11 (100). – С. 187-199.
14. *Daemen J., Rijmen V.* AES Proposial: Rijndael. http://csrc.nist.gov/encryption/aes.
15. *Зензин О.С., Иванов М.А.* Стандарт криптографической защиты – AES. Конечные поля. – М.: КУДИЦ-ОБРАЗ, 2002.
16. *Birukov A., Wagner D.* Advanced Slide Attacks – http://www.csberkeley.cdu~daw/papers/advslide-ec00.ps.

Статью рекомендовал к опубликованию д.т.н., профессор Я.Е. Ромм.

Бабенко Людмила Климентьевна – Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Южный федеральный университет»; e-mail: blk@fib.tsure.ru; 347928, г. Таганрог, ул. Чехова, 2; тел.: 88634312018; кафедра безопасности информационных технологий; профессор.

Ищукова Евгения Александровна – e-mail: jekky82@mail.ru; тел.: 88634371905; кафедра безопасности информационных технологий; доцент.

Babenko Lyudmila Klimentevna – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education "Southern Federal University"; e-mail: blk@fib.tsure.ru; 2, Chehov street, Taganrog, 347928, Russia; phone: +78634312018; the department of security of information technologies; professor.

Ischukova Evgeniya Aleksandrovna – e-mail: jekky82@mail.ru; phone: +78634371905; the department of security of information technologies; associate professor.